



## Review Article

## Thermal-hydraulic 0D/3D coupling in OpenFOAM: Validation and application in nuclear installations

Santiago F. Corzo <sup>a, b</sup>, Dario M. Godino <sup>a</sup>, Alirio J. Sarache Piña <sup>a</sup>, Norberto M. Nigro <sup>a, b</sup>, Damian E. Ramajo <sup>a, b, \*</sup><sup>a</sup> CIMEC Centro de Investigación de Métodos Computacionales (UNL, CONICET), Santa Fe, Argentina<sup>b</sup> UNL Universidad Nacional del Litoral, Santa Fe, Argentina

## ARTICLE INFO

## Article history:

Received 19 May 2022

Received in revised form

14 November 2022

Accepted 1 January 2023

Available online 5 January 2023

## Keywords:

CFD

3D/0D coupling

Thermal-hydraulics

## ABSTRACT

The nuclear safety assessment involving large transient simulations is forcing the community to develop methods for coupling thermal-hydraulics and neutronic codes and three-dimensional (3D) Computational Fluid Dynamics (CFD) codes. In this paper a set of dynamic boundary conditions are implemented in OpenFOAM<sup>®</sup> in order to apply zero-dimensional (0D) approaches coupling with 3D thermal-hydraulic simulation in a single framework. This boundary conditions are applied to model pipelines, tanks, pumps, and heat exchangers. On a first stage, four tests are performed in order to assess the implementations. The results are compared with experimental data, full 3D CFD, and system code simulations, finding a general good agreement. The semi-implicit implementation nature of these boundary conditions has shown robustness and accuracy for large time steps. Finally, an application case, consisting of a simplified open pool with a cooling external circuit is solved to remark the capability of the tool to simulate thermal hydraulic systems commonly found in nuclear installations.

© 2023 Korean Nuclear Society, Published by Elsevier Korea LLC. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

The analytic or semi-analytic models supported by empirical correlations have been used for decades to design and assess complex installations [1]. Simple approaches such as the steady-state Bernoulli equation have been applied to solve the transient behavior of hydraulic networks with acceptable accuracy [2]. However, the main limitations of such simple methods are found in the solution of unsteady turbulent flow, multiphase flow, and phase change problems, especially with complex three-dimensional (3D) geometries. In this sense, the use of Computational Fluid Dynamics (CFD) simulation in industrial applications is growing in relevance and confiability. But, despite the capacity of the codes and the progress in hardware, the computer power is still not enough to achieve long time simulations of large installations.

Nuclear safety assessment involves the analysis of large installations with very different dimensional scale components, where single and two-phase flows are present. Thus, the simulation

of long-time Nuclear Power Plant (NPP) events, such as Loss of Coolant Accident (LOCA), Station BlackOut (SBO), and Loss of Forced Accident (LOFA), imposes strong restrictions on the Courant number. This forces the community to use system codes based on the domain reduction and empirically based models. In this sense, the more widely used codes such as RELAP, TRACE, ATHLET, and CATHARE, were mainly developed to be applied to II and III generation NPPs, but the application to IV generation NPPs remains challenging. An interesting document comparing the capabilities of these codes was done by Bestion [3].

As mentioned above, the CFD has not got the ability to address large simulations of nuclear accidents due to two main issues: computational cost and the capability to solve small turbulent scales. The use of Direct Numerical Simulation (DNS) allows computing all the turbulent scales, but the computational cost is still huge. On the other hand, the application of Reynolds Average Navier-Stokes (RANS) models reduces such cost significantly, but the solutions are generally highly dependent on the turbulence modelling. Moreover, the interfacial mass, momentum and energy exchange which characterize multiphase flows, have to be properly estimated through mechanistic and correlation models based on experimental data. Therefore, the reliable use of CFD tools is still confined to a small portion of nuclear reactor components. In this

\* Corresponding author. CIMEC Centro de Investigación de Métodos Computacionales (UNL, CONICET), Santa Fe, Argentina.

E-mail addresses: [damianramajocimec@gmail.com](mailto:damianramajocimec@gmail.com), [dramajo@santafe-conicet.gov.ar](mailto:dramajo@santafe-conicet.gov.ar) (D.E. Ramajo).

List of symbols		$D_s$	Characteristic grid size [m]
$\gamma$	Phase fraction []	<i>Acronyms</i>	
$\mathbf{U}$	Velocity [m/s]	CFD	Computational Fluid Dynamics
$p$ :	Pressure [Pa]	GAMG	Geometric Algebraic Multi-Grid
$t$	Time [s]	MULES	Multid. Univ. Limiter with Explicit Sol
$\rho$	Density [kg/m <sup>3</sup> ]	RANS	Reynolds-averaged Navier-Stokes
$\mu$	Dynamic viscosity [Pa s]	SGS	Symmetric Gauss-Seidel
$\tau$	Stress tensor [N/m <sup>2</sup> ]	VOF	Volume of Fluid
$I$	Identity matrix	SBL:	Siphon Breaker Line
$k$	Turbulent kinetic energy [m <sup>2</sup> /s <sup>2</sup> ]	SBH	Siphon Breaker Hole
$\epsilon$	Turbulent dissipation rate [m <sup>2</sup> /s <sup>3</sup> ]	LOCA	Loss Of Coolant Accident
$\sigma$	Surface tension coefficient [N/m]	LOFA	Loss Of Flow Accident
$\mathbf{g}$	Gravitational acceleration [m/s <sup>2</sup> ]	NPP	Nuclear Power Plant
$\mathbf{S}$	Mean rate stress tensor	PIMPLE	PISO + SIMPLE
$\mathbf{x}$	Position vector [m]		

context, the multi-dimensional methodology seems to be the intermediate step between the traditional system codes and full 3D simulation.

To overcome the above mentioned issues, the nuclear engineering community has invested immense efforts to couple system codes with CFD codes [4] [5] [6]. There are few publications addressing this topic through semi-implicit or implicit techniques while most such papers are based on explicit coupling. In the later, the system codes and 3D CFD codes run in a staggered way, communicating the necessary data through temporary files and flags. One of the first attempts was done by Martin et al. [7], who coupled RELAP5 with other codes transferring data through a Parallel Virtual Machine (PVM) protocol. In the same way, Aumiller et al. [8] did so using RELAP5-3D with a CFD code showing good results while finding some numerical instability. More recent developments also using explicit methods, coupled RELAP with ANSYS-CFX [9], ATHLET with OpenFOAM<sup>®</sup> [10], and TRACE with Star-CCM [11]. In all cases the system codes assumed the master role, sending the necessary information such as mass flow, pressure, and temperature, amongst others, to the CFD code. These mass-flow/pressure couplings are relatively simple because the pressure in the coupled domains remains constant until the end of the time step, but they could lead to stability problems under large time steps or fast transient simulations [12]. In order to eliminate numerical instabilities Weaver et al. [13] proposed a generic semi-implicit coupling technique, which uses the master/slave communication roles but update both the velocity and pressure in each communication step. This coupling methodology showed good performance in several flow conditions [14] [15]. Bandini et al. [16] presented a detailed compilation of the publications concerning code-coupling tools to simulate new-generation reactors focusing on the validation of these tools against experimental facilities.

The objective of the present paper is to combine low cost zero-dimensional (0D) modeling based on the implementation of empirical correlations inside dynamic boundary conditions (DBC) within the 3D CFD OpenFOAM-7<sup>®</sup> framework [17] to simulate thermal-hydraulic installations. That is, single components such as pipes, tanks, elbows, pumps, flow restrictions, and heat exchangers are spatially lumped inside dedicated DBCs, thus guaranteeing the strong coupling between 0D and 3D models. Three DBCs were implemented: first, a single condition to connect the DBC to a single boundary patch of a 3D domain. Second, a double condition to connect two boundary patches of the 3D domain. Third, a thermal DBC to model the heat transfer in shell and tube heat exchangers.

The main advantage of the proposed implementation is in the semi-implicit nature of the DBC, which is updated inside each pressure-velocity iteration loop (PIMPLE), improving stability and accuracy. However, some significant limitations due to the 0D DBC approach should be highlighted:

- Only average values are available for the density, pressure, velocity, and temperature. This could represent an obstacle to model some relevant phenomena in nuclear safety such as the natural circulation heat remotion during LOFA or siphoning effects during LOCA. These limitations could be solve by including ad hoc empirical models or implementing one-dimensional DBCs;
- The inability to solve complex hydraulic networks with plenums, manifolds or bifurcations. Currently, this can be overcome by using 3D components to represent these multi-junction components;
- The inability to handle variable cross-section pipes and calculating the changes in pressure and velocity that strongly affect the frictional and form pressure drop;
- d- The inability to account for the heat exchange with the environment and the thermal inertia of the materials, which could have a significant effect on the thermal evolution of the fluids during transient simulations.

This paper is outlined as follows: first, the 0D mathematical background is introduced and the OpenFOAM<sup>®</sup> [17] solver is described. Then, a set of benchmarks are solved and results are compared against RELAP5Mod3 [18], full 3D-CFD and experimental measurements. Finally, a case including thermal and hydraulics DBCs is proposed. The last consists of a 3D water pool coupled with an external cooling circuit represented by DBCs. The pool houses a set of heated bodies, which could represent many different industrial processes.

## 2. Zero-dimensional modeling

In this section the following aspects are discussed: 0D momentum and energy equations, mathematical background and numerical aspects of 3D0D code implementation in OpenFOAM<sup>®</sup>.

The unsteady single-phase Reynolds average momentum equation is:

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = -\nabla p + \nabla \tau + \rho \mathbf{g} \quad (1)$$

where  $\mathbf{U}$ ,  $t$ ,  $p$ ,  $\tau$ , and  $\mathbf{g}$  are the velocity, time, pressure, stress tensor and gravitational acceleration, respectively. The OD formulation of Equation 1 to solve the frictional and form pressure drops in pipelines through the Darcy-Weisbach approach reads as:

$$\frac{U - U_0}{\Delta t} (\rho L S_f) = (p_{inl} - p_{out}) S_f + \rho g (h_{inl} - h_{out}) S_f - S_f \frac{|U|}{U} \left( \frac{1}{2} \rho \lambda \frac{L}{D_h} U^2 + \frac{1}{2} \rho K U^2 \right) + \Delta P_{ext} S_f \quad (2)$$

where  $p_{inl}$ ,  $p_{out}$ ,  $h_{inl}$ , and  $h_{out}$  are the pressures and heights of the inlet and outlet pipe ends.  $S_f$  is the pipe cross-section,  $L$  is the total pipe length,  $\lambda$  and  $K$  are the friction and form coefficients and  $D_h$  is the hydraulic diameter. The term  $\Delta P_{ext}$  in Equation 2 is a generic term, which can be used to introduce pumps or to represent any momentum source such as a buoyancy force. The ratio  $|U|/U$  is included to take into account the sign of the friction and form pressure drop.

The non-compressible approach is normally employed for low Mach number flows. Moreover, for constant cross section pipes the area  $S_f$  can be removed and Equation 2 can be integrated in time. To account for that, the Backward-Euler approach is applied:

$$U = U_0 + \Delta t [(p_{inl} - p_{out}) / (\rho L) + g (h_{inl} - h_{out}) / L] + \Delta t \left[ \left( \frac{1}{2} \lambda \frac{L}{D_h} U^2 + \frac{1}{2} K U^2 \right) / L + \Delta P_{ext} / \rho \right] \quad (3)$$

The total form pressure loss is calculated as the sum of all form drops  $K = \sum_{i=1,n} K_i$ . As noted in Equation 3, once the inlet and outlet pressures  $p_{out}$  and  $p_{in}$  are known, the velocity  $U$  is estimated by numerical iteration.

Fig. 1 sketches the three possible coupling configurations: in Fig. 1-a the DBC is used to represent a fixed pressure ( $p_{in} = p_0$ ). As a consequence, the velocity imposed to the boundary patch of the 3D domain is a function of this fixed pressure  $p_0$  and the pressure in the 3D patch,  $p_{bc2}(U_{bc2} = f(t, p_{bc2}, p_0))$ . In Fig. 1-b the DBC is used to model a variable level tank connected to a pipeline. In this case,  $p_{in}$  is calculated as the sum of a reference pressure  $p_0$  plus the hydrostatic one,  $p_{in}(t) = p_0 + \rho g h(t)$ , where  $h(t) = h_0 - (U_{bc2} S_f \Delta t) / A_r$ . Hence, the DBC computes the liquid level  $h$  to estimate the discharge flow velocity,  $U_{bc2} = f(t, p_{bc2}, p_{in})$ . Finally, Fig. 1-c sketches a more general case where two 3D domains are coupled through a OD pipeline, and the velocity in the OD model depends on the pressure difference between both 3D domains,  $U_{bc1,2} = f(t, p_{bc,1}, p_{bc,2})$ .

The frictional pressure drop coefficient  $\lambda$  in Equation 3 depends on the flow regime; for laminar flow ( $Re < 2200$ ) the Hagen-Poiseuille correlation is appropriated:

$$\lambda_l = 64 / Re \quad (4)$$

For turbulent flow ( $Re \geq 3000$ ) the Colebrook-White equation is suitable [19]. This transcendental equation can be solved using the Zigrang-Sylvester approach [20]:

$$\frac{1}{\sqrt{\lambda_t}} = -2 \log_{10} \left[ 0, 27 \frac{r}{D_h} \right]$$

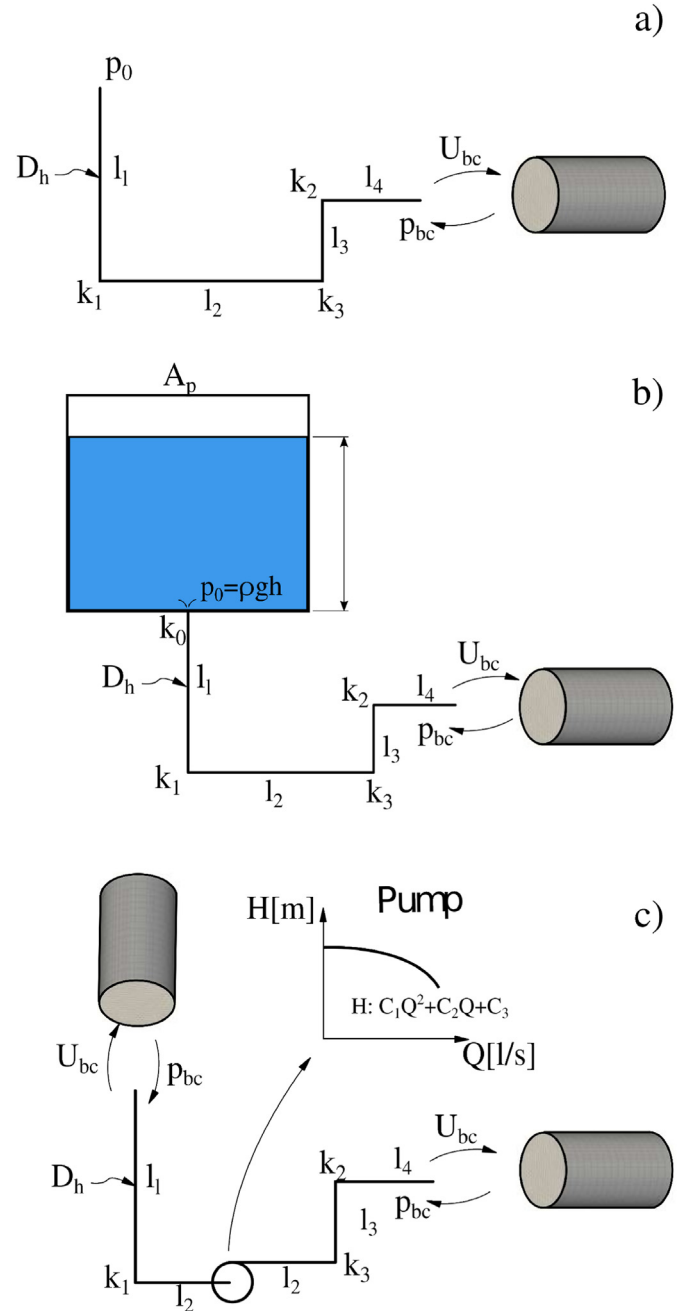


Fig. 1. Sketches of coupling between the 3D domains and DBCs: a) a 3D domain connected to a pipeline subjected to a constant pressure, b) a 3D domain connected to a variable level tank, c) two 3D domains linked through a pipeline.

$$-2 \log_{10} \left[ \frac{2, 15}{Re} \left[ 1, 14 - 2 \log_{10} \left( \frac{r}{D_h} + \frac{21, 25}{Re^{0,9}} \right) \right] \right] \quad (5)$$

where  $r$  is the roughness of the surface. Finally, the transition between laminar and turbulent flow is computed through:

$$\lambda_{lt} = \left( 3, 75 - \frac{8, 250}{Re} \right) \left( \lambda_{t,3000} - \lambda_{l,2200} \right) + \lambda_{l,2200} \quad (6)$$

where  $\lambda_{t,3000}$  is the turbulent friction factor at  $Re = 3000$ , and  $\lambda_{l,2200}$  is the laminar friction factor at  $Re = 2200$ .

The current DBC allows to include pumps working under normal

operation (pump quadrant). Pumps are represented with the characteristic Height(H) - Flow rate(Q) curve described through a polynomial function with four user-input coefficients:  $H = f(Q) = C_1Q^3 + C_2Q^2 + C_3Q + C_4$ . Then, the external source in Equation 3 is calculated as  $\Delta P_{ext} = \rho gH$ .

Regarding thermal coupling, a heat exchanger DBC is also implemented. This performs a 0D thermal balance and imposes the temperature at the 3D boundary patches taking into account the heat transferred between the primary and secondary circuits of the heat exchanger. The current DBC mimics the well-known shell and tube heat exchangers shown in Fig. 2. The DBC reads the flow rate ( $\dot{m}_h$ ) and temperature ( $T_{h,inl}$ ) of the fluid at the 3D inlet patch and calculates and imposes the temperature at the outlet one. In this DBC, the flow rate ( $\dot{m}_c$ ) and temperature ( $T_{c,inl}$ ) of the secondary circuit are user-input parameters.

The numerical procedure implemented to find outlet temperatures  $T_{h,out}$  and  $T_{c,out}$  for the primary and the secondary circuits is based on the widely used Effectiveness-NTU Method [21], which is summarised in Table 1. The efficiency of the heat exchanger,  $\epsilon$ , depends on the geometry as well as on the mass flow rates of both circuits. This efficiency is also a function of the type of heat exchanger, and involves two dimensionless parameters: the capacity ratio  $c$  and the number of transfer units  $NTU$ . In the Effectiveness-NTU method, the unknown variables for each time-step are the heat transfer rate  $\dot{Q}$  and the outlet temperatures of the two circuits,  $T_{h,out}$  and  $T_{c,out}$ . Meanwhile heat transfer area  $A$ , heat flow  $\dot{q}_h = \dot{q}_c = \dot{Q}$ , global heat transfer coefficient  $U_{gl}$  and inlet

temperatures  $T_{h,in}$  and  $T_{c,in}$  are known or input-user parameters.  $\dot{Q}$  is the real heat transfer rate, which is determined from a thermal balance.  $C_c = \dot{m}_c C_{p,c}$  and  $C_h = \dot{m}_h C_{p,h}$ , where  $C_p$  is the heat capacity.  $\dot{Q}_{max}$  is the heat transfer rate corresponding to the maximum difference of temperature among the circuits.  $C_{min}$  is the lowest value between  $C_h$  and  $C_c$ , and  $U$  is the overall heat transfer coefficient. Finally, the efficiency  $\epsilon$  for a heat exchanger with one shell pass and four tube passes is computed by Equation 7.a [21].

### 2.1. Numerical implementation of the dynamic boundary conditions

As above mentioned, three DBC were implemented:

- A single DBC, named *single Pipe Fixed Value*, to impose hydraulic conditions at a single boundary patch, as the shown in Fig. 1-a,b.
- A double DBC, named *coupling Pipe Fixed Value*, to hydraulically link two boundary patches belonging to a 3D domain or two 3D domains, as showed in Fig. 1-c.
- A heat DBC, named *coupling Heat Fixed Value*, to thermally link two patches belonging to a 3D domain or two 3D domains.

A typical input-user data for the *coupling Pipe Fixed Value* DBC is depicted below for the velocity field:

```
BCname
{
  type      couplingPipeFixedValue;
  master   yes;      //ID []
  neiPatchName "NeiName"; //Nei. []
  hStart   1.0; //Master height [m]
  hEnd     3.0; //Neighbour height [m]
  Lline    1.0; //Pipe length [m]
  rug      5e-4; //Pipe roughness [m]
  kEff     0.4; //Eff. loss coeff. []
  Dh       0.0254; //Hydr. diameter [m]
  pmpC1    3.0; //Pump coefficient []
  pmpC2    7.0; //Pump coefficient []
  pmpC3    1.0; //Pump coefficient []
  value    uniform ( 0 0 0 );
}
```

This DBC example corresponds to a pipeline linking the boundaries "BCname" and "NeiName". Both boundaries declare the same input data, but only the master computes the velocity.

The DBC implementation and the PIMPLE pressure-velocity coupling method in OpenFOAM® is briefly described in the algorithm 1. PIMPLE is a combination of PISO (Pressure Implicit with Splitting of Operator) and SIMPLE (Semi-Implicit Method for

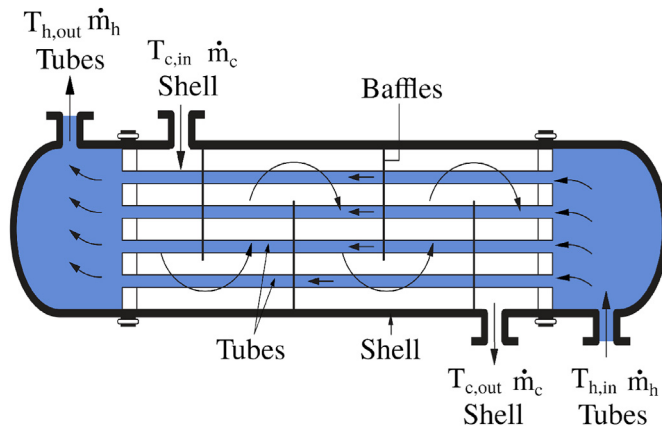


Fig. 2. Sketch of shell and tube heat exchangers represented by the DBC.

Table 1  
Summary of the Effectiveness-NTU method.

Equation	Equation number
$\epsilon = \frac{\dot{Q}}{\dot{Q}_{max}}$	1.a
$\dot{Q} = C_c(T_{c,out} - T_{c,inl}) = C_h(T_{h,inl} - T_{h,out})$	2.a
$\Delta T_{max} = T_{h,inl} - T_{c,inl}$	3.a
$\dot{Q}_{max} = C_{min}(T_{h,inl} - T_{c,inl})$	4.a
$c = \frac{C_{min}}{C_{max}}$	5.a
$NTU = \frac{UA_s}{C_{min}}$	6.a
$\epsilon = 2 \left\{ 1 + c + \sqrt{1 + c^2} \frac{1 + \exp[-NTU\sqrt{1 + c^2}]}{1 - \exp[-NTU\sqrt{1 + c^2}]} \right\}^{-1}$	7.a
$T_{h,out} = T_{h,inl} - \frac{\dot{Q}}{C_h}$	8.a

Pressure-Linked Equations). The algorithm includes three loops; The first one is the time loop progression. The second loop is the "outer" PIMPLE loop where the momentum equation system ( $UEqn_i$ ) is assembled and the velocity  $U_{i-1}^n$  is predicted for the first time. The third loop is for the PISO correction, where the Poisson pressure equation ( $p_j^n$ ) is assembled, solved and updated. After that, the velocity  $U_j^n$  is updated with the corrected pressure. Note that the energy equation is solved out of the PISO loop, once the velocity and pressure are updated.

In OpenFOAM® the Dirichlet boundary conditions are refreshed every time the momentum equation is assembled inside the outer PIMPLE loop in step 4. In this step the OD  $U$  equation (Equation 3) is solved implicitly as:

$$U_i^n = U^{n-1} + \Delta t \left[ (p_{bc,1}^n - p_{bc,2}^n)_{i-1} / (\rho L) + g(h_{inl} - h_{out}) / L \right] + \Delta t \left( \frac{1}{2} \lambda_{i-1} \frac{L}{D_h} U_i^{n,2} + \frac{1}{2} K U_i^{n,2} \right) / L + (\Delta t H_{i-1} g) / L \quad (7)$$

In Equation 7 the superscripts  $n$  and  $n-1$  indicates the current and previous time steps, and subscript  $i-1$  corresponds the previous PIMPLE iteration. In this equation both friction factor  $\lambda_{i-1}$  and the pump head  $H_{i-1}$  are explicitly solved, as was above described.

#### Algorithm 1. PIMPLE algorithm

---

```

Input: read input parameters: Mesh, Initial and boundary conditions, and numerical settings.
// Time-step loop
1 for  $t \rightarrow t_0$  to  $FinalTimeStep$  do
  // Outer PIMPLE LOOP
  2 for  $i \rightarrow 1$  to  $nOuterCorr$  do
    Assemble the momentum eqn.  $UEqn_i$  in terms of  $U_{i-1}^n, p_{i-1}^n$ 
    4 Solve Momentum predictor ( $U_i^n$ ) from  $UEqn$ 
    // PISO loop
    5 for  $j \rightarrow 1$  to  $nCorr$  do
      Assemble the Poisson pressure eqn.  $pEqn$ 
      7 Solve pressure field  $p_j^n$ 
      8 Solve velocity fields  $U_j^n$ 
    9 Assemble the energy eqn.  $TEqn_i$ 
    10 Solve temperature equation ( $T_i^n$ ) from  $TEqn_i$ 

```

---

The velocity  $U^n$  in the second order polynomial Equation 7 is implicitly solved:

$$U^n = \frac{-B \pm (B^2 - 4AC)^{1/2}}{2A} \quad (8)$$

where  $A = \Delta t(0.5k_{eff}/L) + 0.5\lambda/D_h$ ,  $B = 1$ , and  $C = -U^{n-1} - \Delta t \left[ (p_{bc,1}^n - p_{bc,2}^n) / (\rho L) + g(h_{inl} - h_{out}) / L \right] - (\Delta t H g) / L$

As noted, the argument ( $B^2 - 4AC$ ) in Equation 8 has to be positive, thus imposing a restriction on the maximum time step. However, a sub-cycling method is implemented to overcome that limitation.

Regarding the thermal DBC, the energy balance is written as:

$$T_{bc,1}^n = T_{bc,2}^n + \frac{q_i}{\rho C_p U_i^n S_f} \quad (9)$$

where  $T_{bc,2}$  is the temperature read in the outlet boundary patch and  $T_{bc,1}$  is the computed temperature imposed at the inlet boundary patch. The energy equation is assembled at the end of

each PIMPLE loop. Therefore, the inlet temperature is also updated every PIMPLE iteration  $i$ .

The implemented DBCs follow the general form of all OpenFOAM® boundary conditions [17]. That is, a source file (e.g. *couplingPipeFvPatchVectorField.C*) is created with the BC constructor/destructor. For the two hydraulic DBC, a fixed velocity value (Dirichlet) is imposed. Therefore, an *updateCoeffs()* function is required to update the velocity in each PIMPLE iteration. Additionally, a declaration file *couplingPipeFvPatchVectorField.H* is created. This groups all the constructors, declarations of variables and functions of the class. Once the source code is compiled, the DBCs are available for every single-phase and two-phase solvers in OpenFOAM®.

### 3. 3D CFD mathematical background

Simulations were performed with single-phase (*pimpleFoam*) and two-phase (*interFoam*) solvers. For the two-phase problems the Volume of Fluid (VOF) method proposed by Hirt and Nichols [22] was chosen. *interFoam* is suitable for solving segregated problems where the interface among phases is well defined. The VOF method relies on the definition of an indicator function, which identifies whether the computational cell is filled by one fluid, by the other, or by a mixture of them. This is accomplished by the phase fraction  $\gamma$ , which takes values within the range  $0 \leq \gamma \leq 1$ . The extreme values correspond to single-phase flow, e.g.  $\gamma = 0$  for air

and  $\gamma = 1$  for water while intermediate  $\gamma$  values indicate the presence of the interface in this cell. Such interface is typically smeared over a few cells, and is therefore highly sensitive to the grid resolution. The advantage of VOF in free-surface problems is that the method is single-phase in most of the domain. The transport of  $\gamma$  is a pure advection equation, which is solved simultaneously with the continuity and momentum equations.

The VOF equations for unsteady non-compressible two-phase flow are:

$$\nabla \cdot \mathbf{U} = 0 \quad (10)$$

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{U} \gamma) = 0 \quad (11)$$

$$\frac{\partial (\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \sigma \kappa \nabla \gamma \quad (12)$$

where  $\mathbf{U}$  represents the velocity. As noted, there is only one velocity field, which is shared by both phases.  $\gamma$  is the phase volume fraction, and  $\boldsymbol{\tau}$  is the viscous stress tensor ( $\boldsymbol{\tau} = 2\mu \mathbf{S} - 2\mu(\nabla \cdot \mathbf{U})\mathbf{I}/3$ ) with  $\mathbf{S}$  the mean strain rate tensor ( $\mathbf{S} = 0.5[\nabla \mathbf{U} + (\nabla \mathbf{U})^T]$ ). The viscous stress



tensor can be re-written as  $\tau = 2\mu\mathbf{S}$  by substituting the non-compressible condition (Eq. 10). The last term in Equation 12 was proposed by Brackbill [23] to model the surface tension force. In this,  $\sigma$  is the surface tension coefficient and  $\kappa$  is the local curvature of the interface, which is defined as:

$$\kappa = -\nabla \cdot \eta = -\nabla \cdot \left( \frac{\nabla \gamma}{|\nabla \gamma|} \right) \quad (13)$$

where  $\eta$  is the normal vector to the interface.

The physical properties of the fluid far from the interface correspond to one of the fluids while mixture properties are calculated in the cells crossed by the interface, as a weighted average based on  $\gamma$  and using constant properties for each phase:

$$\rho = \gamma\rho_l + \rho_g(1 - \gamma) \quad \mu = \gamma\mu_l + \mu_g(1 - \gamma) \quad (14)$$

In order to simplify the pressure condition at boundaries, OpenFOAM<sup>®</sup> uses a modified pressure to account for the hydrostatic pressure. This modified pressure is defined as:

$$p_d = p - \rho\mathbf{g} \cdot \mathbf{x} \quad (15)$$

where  $\mathbf{x}$  is the position vector.

Finally, the momentum equation can be rearranged as follows:

$$\begin{aligned} \frac{\partial(\rho\mathbf{U})}{\partial t} + \nabla \cdot (\rho\mathbf{U} \otimes \mathbf{U}) - \nabla \cdot (\mu\nabla\mathbf{U}) = \\ -\nabla p_d - \mathbf{g} \cdot \mathbf{x} \nabla \rho + \nabla \cdot (\mu(\nabla\mathbf{U})^T) + \sigma\kappa\nabla\gamma \end{aligned} \quad (16)$$

In turbulent problems, the flow fields, e.g. the velocity  $\mathbf{U}$ , has chaotic behavior. Consequently, it can be decomposed into a time-independent mean function  $\bar{\mathbf{U}}$  plus a time-fluctuating  $\mathbf{U}'$  one. For incompressible flow, this ensemble-averaged leads to the Reynolds-Averaged Navier-Stokes (RANS) equation, written as:

$$\begin{aligned} \frac{\partial(\rho\bar{\mathbf{U}})}{\partial t} + \nabla \cdot (\rho\bar{\mathbf{U}} \otimes \bar{\mathbf{U}}) - \nabla \cdot (\mu\nabla\bar{\mathbf{U}}) \\ = -\nabla p_d - \mathbf{g} \cdot \mathbf{x} \nabla \rho + \nabla \cdot (\mu(\nabla\bar{\mathbf{U}})^T) - \nabla \cdot (\rho \bar{U}' \otimes U') + \sigma\kappa\nabla\gamma \end{aligned} \quad (17)$$

where  $-\rho\bar{U}' \otimes U'$  is the Reynolds stress tensor, which arises from the ensemble-averaged. Several models have been developed to estimate this term. One of the most widely used is the Boussinesq's eddy viscosity model, which assumes that the Reynolds stress tensor has the same structure to the laminar one, but with an effective eddy viscosity that accounts for the laminar and turbulent viscosities,  $\mu_{eff} = \mu + \mu_t$ . Therefore, the final form of the RANS equation is achieved by replacing the laminar viscosity  $\mu$  by the effective viscosity  $\mu_{eff}$  in Equation (17).

The turbulent viscosity  $\mu_t$  can be obtained through the eddy viscosity model by stating a relation between the turbulent intensity (proportional to the velocity fluctuations) and a characteristic length scale  $L$ , as follows:

$$\mu_t = C_\mu \rho \mathbf{U}' L = C_\mu \rho \frac{k^2}{\epsilon} \quad (18)$$

where  $k$  is the turbulent kinetic energy, and  $\epsilon$  is the turbulent dissipation rate. In this paper, the standard two-equation  $k-\epsilon$  model [24] with standard wall law was chosen to close the equation system:

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho k \mathbf{U}) = \nabla \cdot \left( \frac{\mu_t}{\sigma_k} \nabla k \right) + 2\mu_t S : S - \rho \epsilon \quad (19)$$

$$\frac{\partial \rho \epsilon}{\partial t} + \nabla \cdot (\rho \epsilon \mathbf{U}) = \nabla \cdot \left( \frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right) + C_{1\epsilon} 2 \frac{\epsilon}{k} \mu_t S : S - \rho C_{2\epsilon} \frac{\epsilon^2}{k} \quad (20)$$

The model constants were by default  $C_{\mu} = 0.09$ ,  $C_{1\epsilon} = 1.44$ ,  $C_{2\epsilon} = 1.92$ ,  $\sigma_k = 1.0$  and  $\sigma_\epsilon = 1.3$ .

## 4. RESULTS

Five tests were proposed to assess the DBCs. Tests 1, 2 and 3 were carried out to evaluate the accuracy of the DBC in relatively simple pipeline flows while tests 4 and 5 assessed the capability of the DBCs to simulate more complex thermal-hydraulics installations.

The same setting parameters were used for the five cases. Up-wind schemes were used for the divergence terms on the spatial discretization, linear schemes for gradients, and linear "corrected" schemes for the Laplacian terms. The Multi-grid method (GAMG) was chosen for solving the pressure equation system, and the "smooth Solver" was chosen for solving the other linear equation system. A relative tolerance of  $1 \times 10^{-3}$ , and an absolute tolerance of  $1 \times 10^{-9}$  were considered for convergence criteria. Twelve iterations were set for the outer PIMPLE loop, and ten iterations for the PISO loop. PIMPLE residual controls of  $1 \times 10^{-2}$  and  $1 \times 10^{-9}$  for the relative and absolute errors were specified.

### 4.1. Case 1: tee-junction pipe

Case 1 is showed in Fig. 3. It consists of a 3D Tee-junction with three inlet/outlet boundaries. A *single Pipe DBC* is connected to boundary  $P_1$  representing a horizontal straight pipe with a pump, and a fixed pressure of  $1.3 \times 10^5$  Pa at the pipe inlet. The pump curve data corresponds to a Westinghouse pump available in RELAP database. Other *single Pipe DBC* was connected at boundary  $P_2$  to represent an elevated tank open to the atmosphere, with an initial level of 0.2 m and a cross section area of 0.1 m<sup>2</sup>. Finally, a time depending pressure condition was set at boundary  $P_3$ . The main dimensions of the case and the pressure condition imposed at  $P_3$  are displayed in Fig. 3. As shown, abrupt pressure changes were applied in order to assess the capacity of the implemented tool to solve transient flows.

Fig. 4 shows the mass flow rate through the boundaries  $P_1$  (a) and  $P_2$  (b) and the comparison with RELAP (SC). In both boundaries, the positive MFR corresponds to water entering to the Tee-junction. Clearly, the flow in  $P_1$  is governed by the pump. During the first 2 s the pressure at  $P_3$  remains unchanged and the MFR at  $P_1$  is quickly imposed by the pump. An almost constant MFR around 27 kg/s is reached and remains over 20 kg/s the first 10 s. During the first 4 s the tank discharged and the water level is reduced, but after this time the flow inverts and the tank inventory starts to grow. After 8 s the pressure at  $P_3$  increases abruptly reducing the MFR through  $P_1$  as well as increasing the MFR through  $P_2$ . Both MFRs at  $P_1$  and  $P_2$  show a similar behavior although the MFR through  $P_1$  is significantly higher. After 10 s, the MFRs reach steady values up to the end of simulation.

The comparison between the 3DOD model and RELAP (SC) showed very good agreement. The large discrepancies were for  $P_2$  and could be justified by the inaccuracy of the SC for predicting the pressure drop in the Tee-junction, both in forward and backward flow directions.

In order to study the influence of the mesh, three grids of 184,927, 587,883 and 4,540,897 cells were considered for the Tee-junction domain. The results are displayed in Fig. 5 for  $P_1$

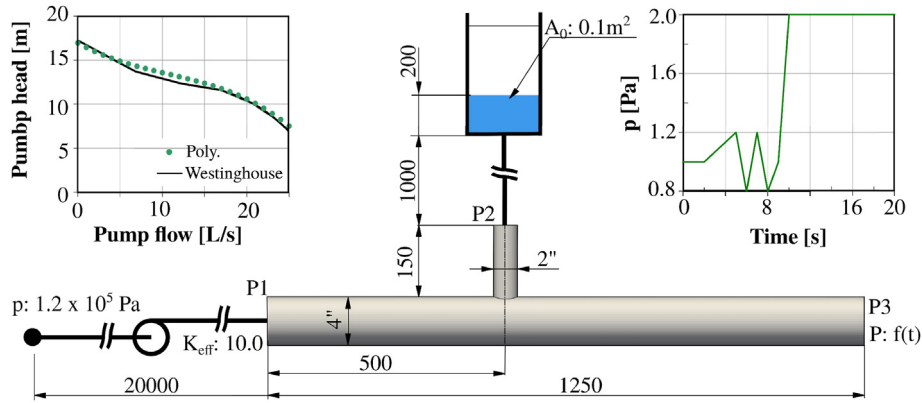


Fig. 3. Case 1: Details of the test domain.

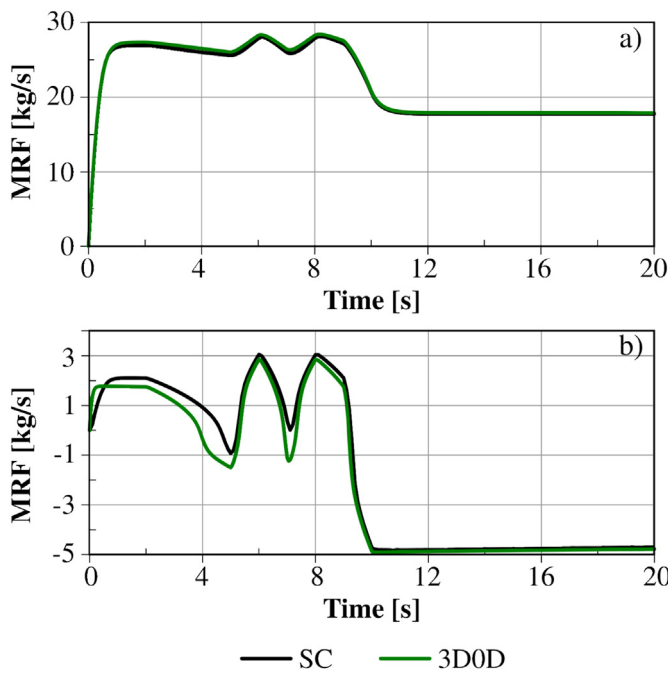


Fig. 4. Case 1: Mass flow rate through P1 (a) and P2 (b) with the 3D0D model and RELAP (SC).

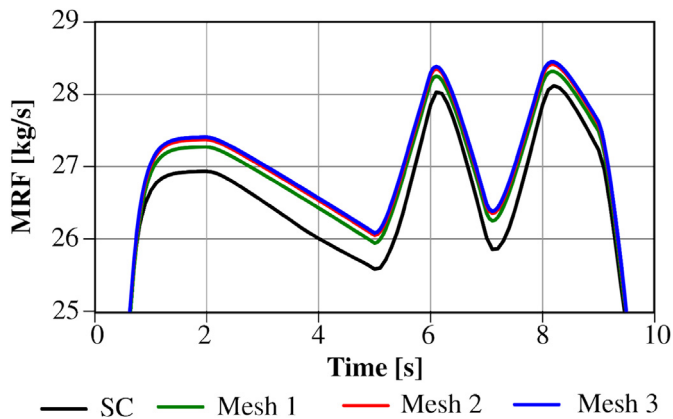


Fig. 5. Case 1: Mass flow rate through P1 with the 3D0D model and three grids and RELAP (SC).

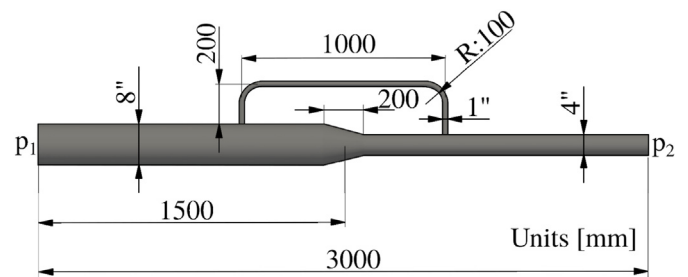


Fig. 6. Case 2: By-pass piping geometry.

boundary, which showed the largest discrepancies. On the one hand, mesh converge was reached and the intermediate grid was quite fine to achieve acceptable solutions. On the other hand, after refining, the 3D0D results turned away from the SC predictions.

The results obtained for Case 1 demonstrated the versatility of the *DBC* to solve many types of hydraulic components. The pressure variations imposed at the  $P_3$  boundary and the dynamic response at the two other boundaries allowed the assessment of the *DBC* under forward and backward transient flows.

#### 4.2. Case 2: by-pass piping

Case 2 is depicted in Fig. 6. This test was proposed to demonstrate the capability of the "coupling Pipe Fixed Value" *DBC* to link two patches belonging to the same 3D domain. The test consists of a 3D main pipe with a smooth reduction and a small by-pass pipe connecting the both parts of the main pipe. At the left boundary ( $P_1$ ) the water flow rate is imposed with a sinusoidal function  $Q(t) = 3.2 \times 10^{-3} m^3/s [1 + \sin(\pi t)]$  ranging from 0 to  $6.4 \times 10^{-3} m^3/s$ , while the right boundary ( $P_2$ ) is opened to the atmosphere. Before imposing the sinusoidal flow rate, the problem was initialized with a constant flow rate of  $3.2 \times 10^{-3} m^3/s$ . In this case, three approaches were considered: full 3D CFD, RELAP system code and 3D0D model. For the 3D0D case, the main pipe was simulated with a 3D domain while the by-pass pipe was modelled through *DBC*. Only a part of by-pass pipe ends (length of one diameter) was included in the 3D domain in order to capture better the entrance and discharge flows. Therefore, the *DBC* declaration included the two elbows, the friction loss along the pipe, and the difference on elevation between the by-pass ends. The full 3D CFD domain was meshed with 4, 321, 833 cells while the 3D0D domain was meshed with 3, 694, 759 cells. In the RELAP model the main pipe consisted of two pipes with 5 cells each, joined with a single junction for representing the

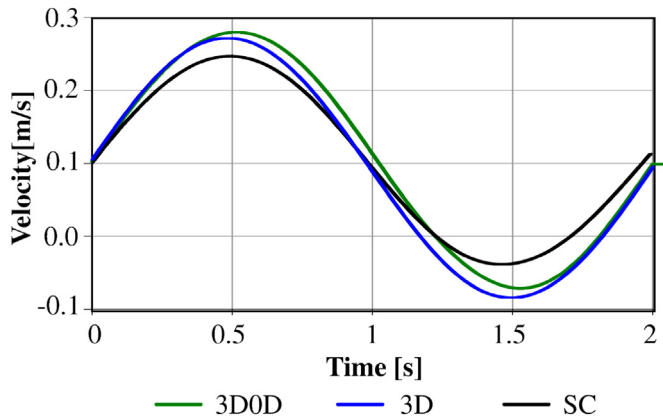


Fig. 7. Case 2: Velocity through the by-pass pipe for the full 3D, the 3D0D and the system code models.

area reduction. The pressure drop across the reduction was accounted through a suitable coefficient at the single junction. Finally, a pipe with 10 cells was included to represent the by-pass pipe and connected to the main pipe through single junctions.

Fig. 7 shows the velocity in the by-pass pipe obtained with this three approaches. As noted, the three curves start from the same point, which is indicating that all models reached the same velocity during the initial steady-state stage. In the first quarter of the cycle, the inlet flow through  $P_1$  duplicates (see the function  $Q(t)$ ), but the velocity through the by-pass pipe increases more than twice. On the other hand, when the main pipe flow becomes zero (at 1.5 s), the flow through the by-pass inverts its direction. The 3D0D and the SC results were quite similar, but the full 3D model underestimated the velocity through the by-pass pipe for around 15% during the extreme inlet flow rates. It should be mentioned that, to achieve an accurate 3D CFD simulation of the internal flow through a pipe is not a simple task, due to the strong effect on the pressure drop and flow rate estimations of the near wall mesh, the turbulent wall law and the turbulent closure model.

#### 4.3. Case 3: Oscillating liquid columns in a U-bend tube

This widely known test was the chose for assessing the "single Pipe Fixed Value" *DBC*. Test geometry, displayed in Fig. 8, consists of a large U-bend made of transparent PVC hosepipe of 12mm of diameter with the column extremes opened to the atmosphere. Ethyl alcohol was used instead of water to reduce the capillarity effects only. The test was initialized by considering a liquid level difference of 0.1 m between both columns. Such level was computed by filming with a high speed camera (120 fps). Although the fluid was coloured to improve visualization, the full liquid level evolution was hard to obtain. Therefore, experimental measurements were centered on estimate the extreme locations of the liquid interface at every cycle. The test was numerically solved using full 3D CFD and 3D0D models. For the 3D0D one, the left half of the geometry was simulated using a 3D CFD model and the right half was represented by a *DBC*. Due to low velocity, turbulence modeling was avoided and the pressure loss was mostly caused by friction because of the large radius of the U-bend. The 3D domain was meshed with structured hexaedral cells with local refinement near the walls. For the full 3D model the mesh size was of 95,400 cells, while for the 3D0D it demanded half of them. The mean and maximum  $y^+$  calculated at the highest flow velocity (during the first oscillation) were 3.25 and 14.73, respectively.

Fig. 9-a shows the liquid level at the right column during the

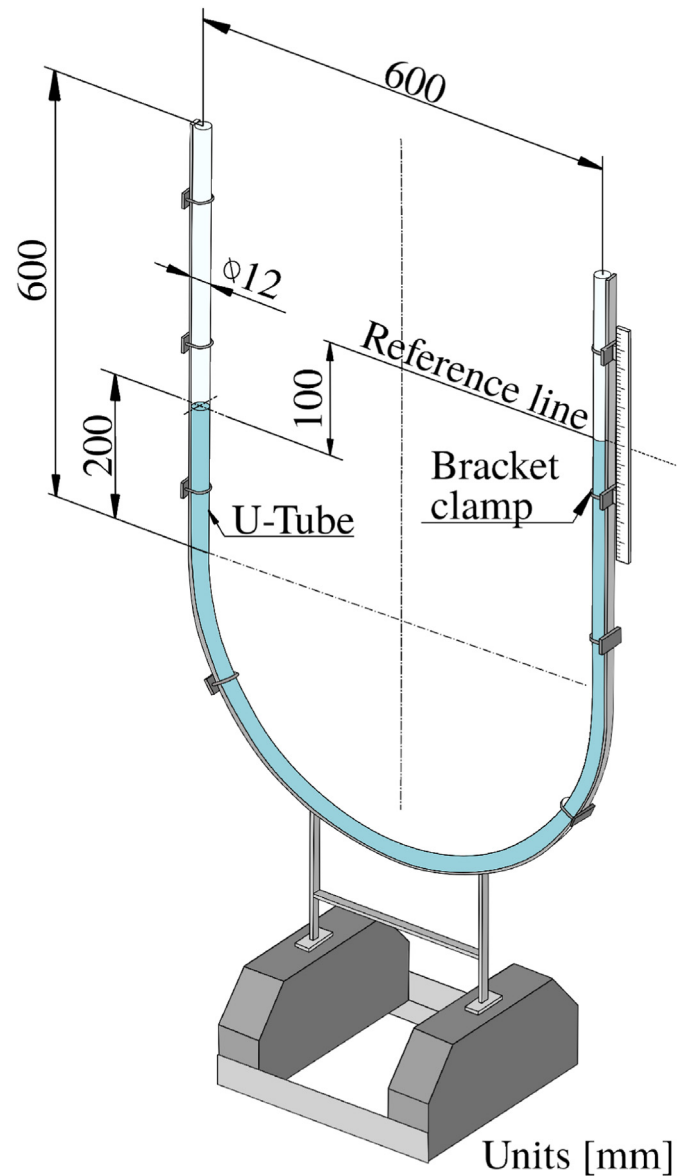


Fig. 8. Case 3: U-bend tube geometry.

first 20 s. This time period was sufficient enough for oscillations to stop. As expected, the transient solution was the typical mass-spring-dampen motion. As noted, the oscillation period of the 3D0D model was slightly lower than the experiment. The full 3D displayed a similar oscillation period, but the maximum liquid levels were underestimated. Fig. 9-b focuses on the first cycle, where numerical and experimental data agreement was very good. Similar to the previous test, the full 3D CFD approach underestimated the velocity. Nevertheless, the agreement with experimental data was more than satisfactory.

Note that, in this case, the differences between the 3D0D and the full 3D CFD models are considerably lower than in the previous cases. The explanation could be found in the fact that turbulence modeling was not needed due to the Reynolds number being less than 2200. The errors in the first-cycle oscillation period were 1.6% and 2.52% while the errors in the maximum displacements were 3.57% and 8.48% for the 3D0D and 3D cases, respectively.

This case shows once again the capability of the *DBC* to



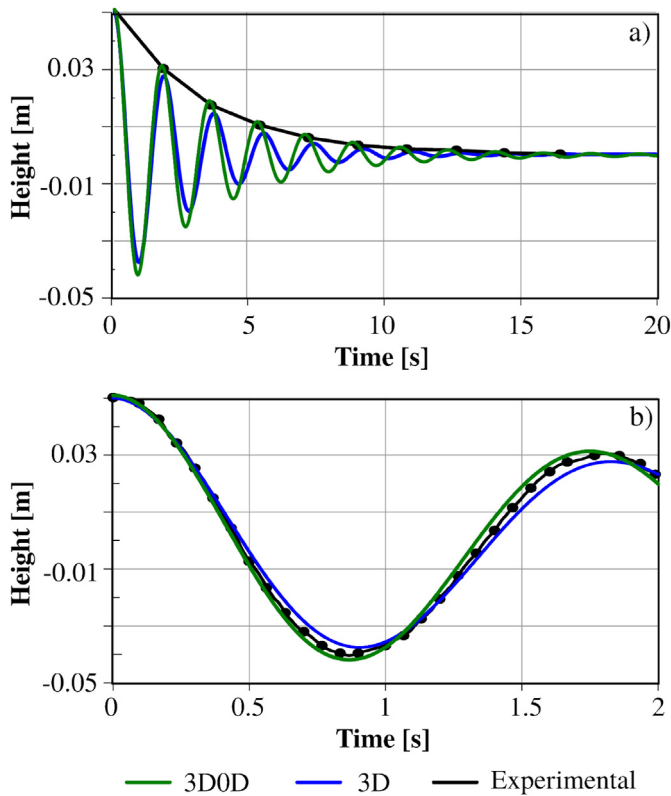


Fig. 9. Case 3: Liquid level results at the right column: a) During the first cycle of oscillation and b) during the complete transient.

represent flow inversion, gravity effect, and energy loss due to wall friction.

#### 4.4. Case 4: pump coast down in pool reactor

This case addresses the simulation of a real pump coast down test in a mock-up research reactor carried out by Wang et al. [25]. The test facility consists of an open-pool reactor with an external coolant circuit. The coolant is pumped from the bottom of the pool and returned through its top. The coolant is impulsed by a centrifugal pump placed 7.26 m downstream of the pool. An inertia tank is placed between the pool and the pump with the aim of

damping the fast reduction on the coolant mass flow rate caused by a pump shutdown. Fig. 10 shows the experimental facility and its relevant geometry parameters as well as the initial liquid levels. The figure also indicates the components represented with 3D CFD and 0D approaches. 3D domains were employed to simulate the pool and the pipeline including the Tee-junction (pipe P1). The rest of the circuit was modeled with DBCs. That is, the pipe P2 and the pump being modelled with a coupling Pipe Fixed Value DBC while the pipe P3 and the inertia tank represented with a single Pipe Fixed Value DBC.

The pressure drop form coefficients for each pipe were calculated from the pressure measurements reported by Wang et al. [25] and a generic pump curve (volumetric flow rate vs pump head) was adopted. The pump shaft inertia was neglected due to the small size of the pump.

The model was run for 100 s until reaching the steady-state condition with a mass flow rate of 8.04 kg/s, which is 4% lower than the experimental one (8.4 kg/s). Similarly, liquid levels in both the pool and the inertia tank were established at 7.02 m and 4.25 m, respectively while for the experiments were of 7 m and 4.2 m. After the initial 100 s, pump shutdown was simulated by establishing the pump coefficients as null in the DBC.

Fig. 11 shows mass flow rates through pipes P1 (pool outlet) and P2 (pool inlet) obtained through the 3D0D model (a) and RELAP (b). Results were compared against experimental data. As shown, agreement was very satisfactory for both approaches and the major discrepancies were less than 0.5 kg/s in the pool outlet. Both models captured the fast flow drop and the change on the flow direction in pipe P2, which is explained by the discharge of the liquid column once the pump was shutdown. Fig. 11-c shows the velocity field in the Tee-junction connecting the main line with the inertia tank during seconds 0.5, 5 and 20. Even after a short 0.5 s period, the velocity through P1 was quite higher than through P2 and most of the flow was moving from the pool to the inertia tank. Flow direction inversion through P2 was predicted at 1.8 s with the 3D0D, but occurred at 2.05 s during experiment. Therefore, after 5 s the flow through P2 moved downwards and the inertia tank was fed from both sides. This continued until the pool and inertia tank levels became nearly similar. Their levels are presented in Fig. 12.

Due to the loss of forced flow, the level in the pool decreased slowly while the inertia tank level increased more quickly due to its lower cross section area. Equilibrium level was around 6.6 m and was reached after around 35 s.

The 3D0D model tended to overestimate the flow rate in the pool outlet and to underestimate it in the pump line. The last is also

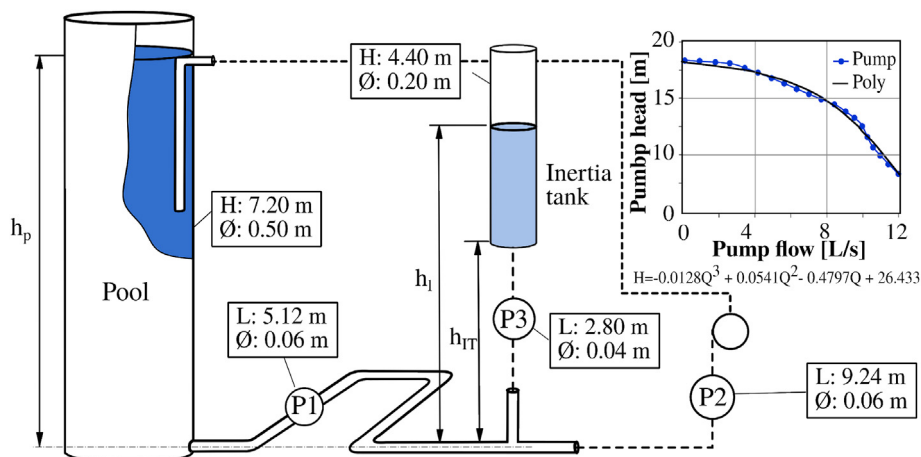


Fig. 10. Case 4: Test domain details.

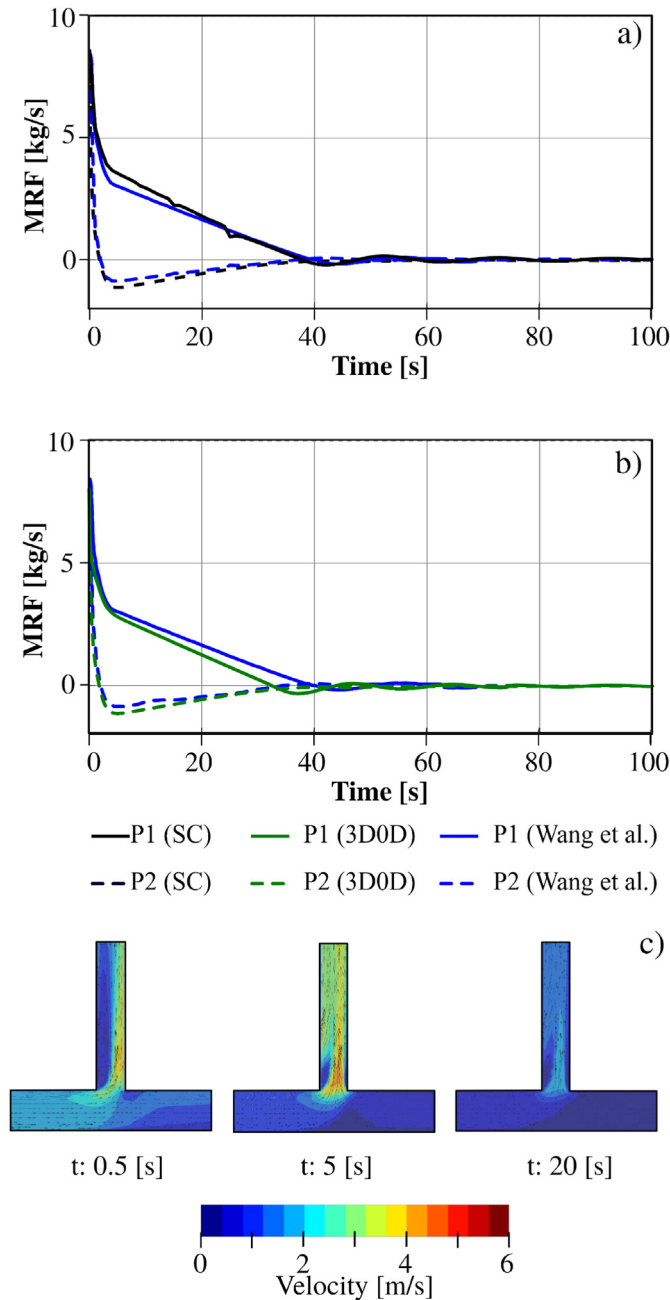


Fig. 11. Case 4: Pump shutdown in pool reactor with inertia: a) 3D0D MFR results, b) SC MFR results, c)3D0D velocity in the tee junction for 0.5s, 5s and 20s.

observed with the RELAP model and could be improved by a better fitting of the pressure loss coefficients. Nevertheless, both numerical approaches were very satisfactory and reflected quite well the coast down test.

#### 4.5. Case 5: cooling processing pool

To conclude the study, the simulation of a generic cooling processing pool was performed. This simple case could represent many complex industrial processes such as grinding balls in a thermal quenching or tempering in metallurgical plants, the decay pool for nuclear spent fuels storage, or cheese freezing in food industry, among others. In these kinds of installations the designing target could be to achieve homogeneous conditions in the whole pool

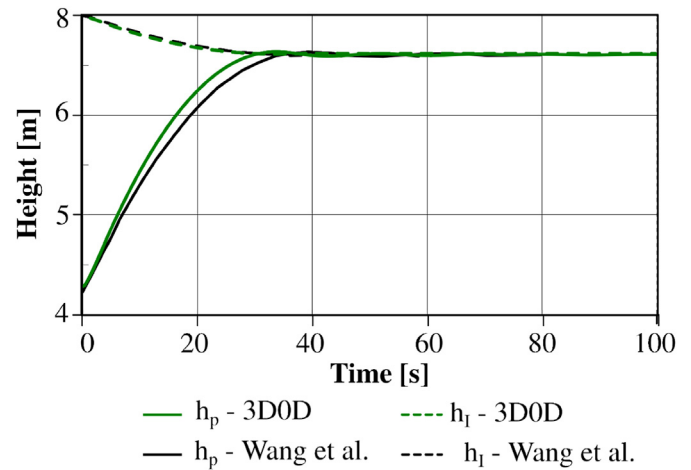


Fig. 12. Case 4: Water column in pool reactor and inertia tank.

(temperature, concentration of any chemical product, etc.), or a specific heating/cooling rate. No matter the application, the pool sizing, the coolant mass flow rate, and the number and location of inlet/outlet throttles should be of relevance for the efficiency of the process. Moreover, the correct sizing of the external devices such as the cooling/heating system, pipeline, pump, storage tanks, or valves could become a hard task if many operation conditions have to be considered. In this context, the use of full-geometry 3D CFD models for designing large and complex installations is restricted due to the number of components as well as the necessity to include control systems and operational procedures. However, 3D0D models allow the representation of external system components, which many times are complex but not crucial, into a DBC. That allow reducing the overall problem focusing to study on the different inlet/outlet configurations in order to optimize the overall installation.

Fig. 13 shows the proposed problem, which consist of a water pool of 2.5 m × 2.5 m × 1.6 m width, depth and height respectively. It was designed for fast cooling of investment casting baskets, filled with spherical balls of 1 inch. The nine baskets were arranged in a squared configuration as displayed in the Figure. The coolant was forced to pass through the baskets extracting the heat from the balls. Then, it flows through the external circuit (DBC) to transfer the heat to a secondary circuit. Fig. 13 shows the external system composed by a heat exchanger, a pump, a regulation valve, and a purifier used for catalyst processes or extraction of impurities. Although the last is not included in the DBC, it is easy for this functionality to be incorporated in the current DBC.

The balls were modeled as porous media through the Darcy-Forchheimer equation. In this case, only the Darcy coefficient was used. The permeability of the baskets was calculated using the Carman-Kozeny equation [26],  $k = (\epsilon^{5.5}d^2)/5.6$ , with the void fraction  $\epsilon = 0.4$ . Regarding heat transfer, a constant volumetric heat source of 80 kW was imposed to every basket and the initial temperature in the pool was 37°C. A more realistic situation should include a heat transfer coefficient depending on the flow conditions, but this is out of the scope of the current paper.

Two types of analysis were carried out with the 3D0D model: first the steady-state operation condition was found for two inlet/outlet configurations. Second, a transient condition of partial loss of the forced flow in the secondary circuit was simulated.

##### 4.5.1. Steady-state simulation

Two inlet/outlet configurations were compared to find the best design in terms of cooling uniformity. Fig. 13 on the right sketches

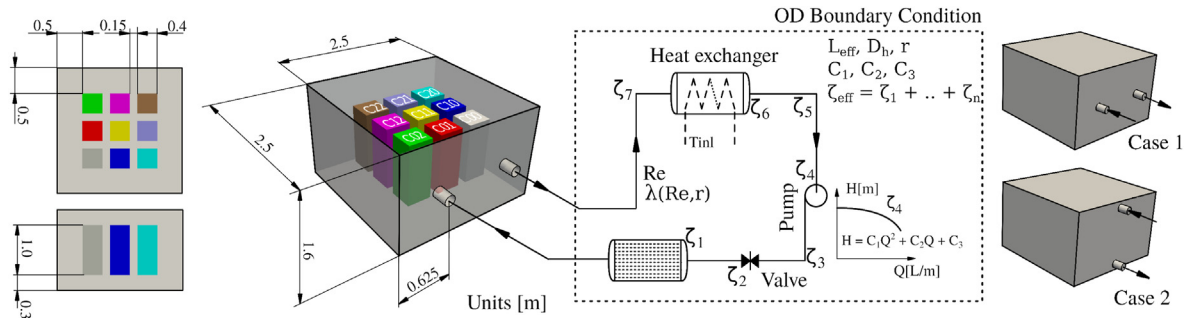


Fig. 13. Case 5: Dimensions and 3DOD configuration.

Table 2  
Operational and model setting parameters.

Primary circuit		
Parameter	Value	Unit
Total length	37.0	[m]
$\zeta_{tot}$	2000.0	[-]
Pump coef.	0.0, -0.01, 0.0, 13.0	$[ms^2/L^2] [ms/L] [m]$
Roughness	$1 \times 10^{-6}$	[m]
Secondary circuit		
Parameter	Value	Unit
Transfer area	47.0	$[m^2]$
Mass flow rate	11.1	$[kg/s]$
Inlet temperature	27	$[^\circ C]$
$U_{gl}$	836.8	[-]

the inlet/outlet designs. In case 1, the inlet and outlet locations were situated on the middle height plane while in configuration 2 they were placed on the vertical plane (inlet at the top and outlet at the bottom).

The external system included a piping of 0.17 m of diameter and a total length of 37 m. The overall pressure loss was represented by one equivalent loss coefficient  $\zeta_{eff} = 2,000$ . Finally, the pump curve

was represented with coefficients  $C_1 = 0.0, C_2 = -0.01, C_3 = 0.0$  and  $C_4 = 13.0$ . The thermal boundary conditions for the inlet/outlet throttles were set through the coupling Heat Fixed Value DBC, using a shell and tubes heat exchanger. Table 2 summarizes external circuit parameters.

Pool domain was discretized with a structured mesh with 557,000 cells with local refinement close to the baskets and the inlet/outlet throttles. As in the previous cases, small pipes of 0.25 m were included in the inlet/outlet throttles. The runs were initiated with a small time step of  $1 \times 10^{-5}$  s to prevent the pump from initially introducing a high flow rate. Next, the time step was automatically calculated to hold the Courant number below 50. In both cases, the steady-state coolant flow rate in the external circuit was reached after only 10 s, but the total run time was more than 10,000 s to achieve time-averaging fields in the pool.

Fig. 14 displays the temperature along a vertical line across all nine baskets. As clearly noted, configuration 1 showed more similar temperatures in all baskets. Only the basket C02 showed significant discrepancies with respect to the others. This was due to the proximity of this basket with the high-velocity cool inlet jet. Although temperature profiles were similar for all baskets, the temperature through the vertical direction varied in more than 30°C. Configuration 2 displayed a different behavior because the temperature profile of baskets C01 and C11 turned away from the rest, but the temperature difference along the vertical direction was only <9°C, which was significantly lower than in Configuration 1. Such thermal distribution was due to the cool inlet jet entering from the top.

Fig. 15 on top shows the temperature in three mean planes-split pool. For configuration 1 a high thermal stratification is clearly observed. On the contrary, for configuration 2 the cool inlet jet descends to the bottom side due to density difference and the location of the outlet throttle. This descendant flow explains the

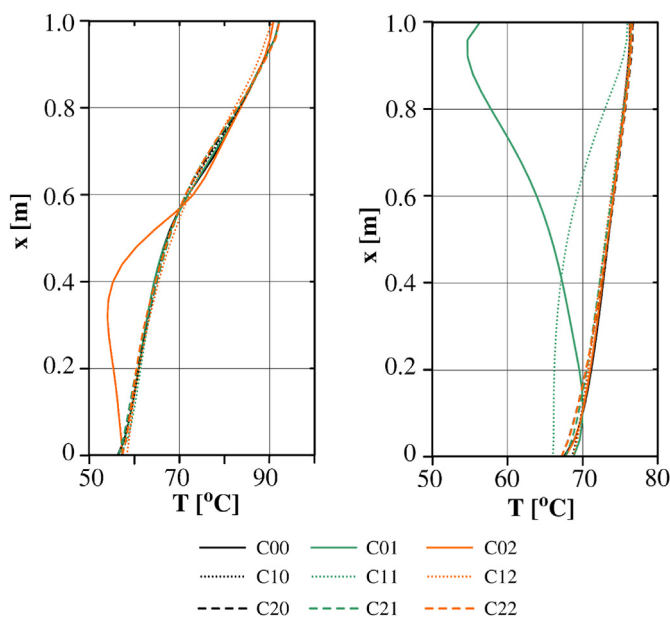


Fig. 14. Temperature along a vertical line crossing the nine baskets: a) Configuration 1, b) Configuration 2.

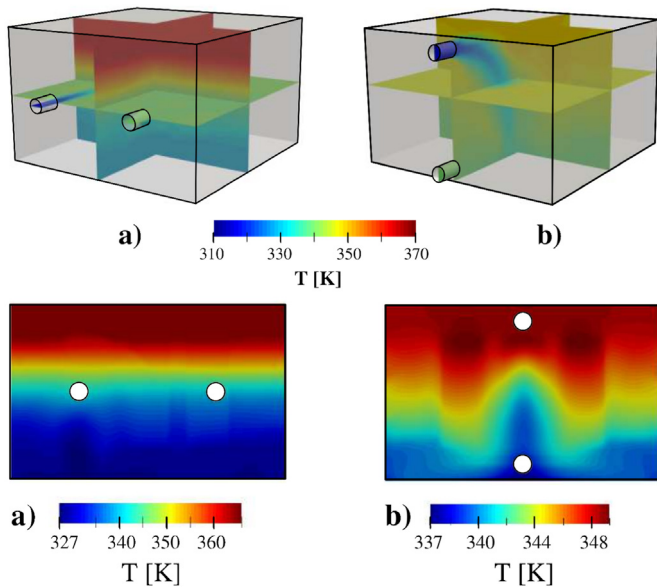


Fig. 15. Temperature distribution in mean planes for a) configuration 1, b) configuration 2.

lower temperature found in baskets C01 and C11 in Fig. 14-b. Fig. 15 at the bottom shows temperature distribution in the mean vertical plane for both configurations. In this case, temperature ranges were defined based on the minimum and maximum values of each configuration. On one hand, configuration 1 displayed homogeneity in the horizontal direction, but significant stratification in the vertical one. On the other hand, configuration 2 showed less homogeneity in the horizontal direction, but a quite lower vertical stratification. Summarizing, most of configuration 1 baskets displayed a large temperature difference along the vertical direction. Therefore, configuration 2 was more appropriated to fulfil the designed target. It seems clear that a re-designing of the pool considering two or three inlets at the top and equal number of outlets at the bottom sides, should improve the results.

As expected, for this two steady-state simulations there were not significant differences on the macroscopic parameters, such as the mean temperature in the pool, mass flow rate in the external circuit or inlet/outlet temperatures. Table 3 summarizes the steady-state results for both configurations. Note that the mean temperature was equal for both cases, which is correct due to the fact that heat power and the external system were mostly the same for both configurations. After steady state is reached, the total heat power transferred to the secondary circuit is equal to the power imposed on the baskets. However, the inlet/outlet temperatures as well as

the final pool temperature, could not be easily estimated without solve the transient evolution of the overall system. The inlet/outlet temperatures are changing until steady conditions are risen, and these variations only can be captured by using the DBCs.

#### 4.5.2. Transient simulation

In this section, a transient simulation considering a partial loss of the secondary water was carried out for Configuration 1, which was the most compromised from the thermal stratification point of view. Secondary water mass flow rate at 100% was 11.1 kg/s (100 % MFR case) but it suddenly drops to 50% (50 % MFR case) due to the loss of one of two secondary pumps.

This transient simulation was run until reaching a new steady-state thermal condition after 7,000 s. Due to the fact that heat power in the baskets was the same, the mean pool temperature increased up to reaching a new equilibrium. Fig. 16-a shows the temperature distribution in all baskets, which increased around 7°C with respect to the initial condition. The flow reduction of the secondary water also increased inlet/outlet temperatures as displayed in Fig. 16-b. The mean temperature of the pool increased only 5.3°C.

The partial loss of forced flow in the secondary circuit generates a thermal imbalance that increases the temperature of the pool. The new equilibrium is reached once the pool inlet/outlet temperatures

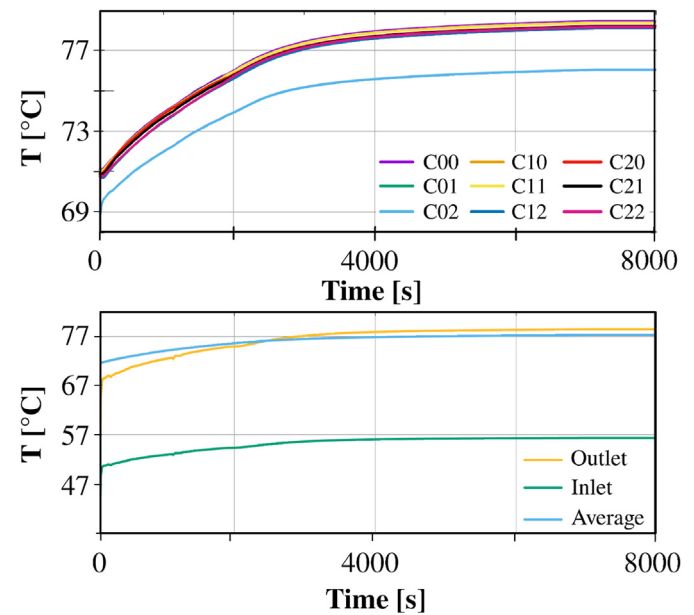


Fig. 16. Case 5: Transient simulation. a) Mean temperature of baskets, b) Mean temperature in the pool and the inlet and outlet.

Table 3

Steady-state results.

Flow in the pool		
Parameter	Configuration 1	Configuration 2
Mean temp. [°C]	71.65	71.18
Inlet temp. [°C]	46.0	45.22
Outlet temp. [°C]	67.7	67
Flow in the external circuit		
Parameter	Configuration 1	Configuration 2
Coolant mass flow rate [kg/s]	7.95	7.96
Heat power [kW]	720	720
Secondary mass flow rate [kg/s]	11.1	11.1
Secondary water outlet temp. [°C]	42.6	42.6



increase, improving heat transfer in the heat exchanger until reaching the power imposed to the baskets. The methodology proposed in this transient simulation demonstrates the capability of the model to simulate operational events that could not be well represented by using fixed boundary conditions. This simple test could be extrapolated to any real operational condition in nuclear installations.

## 5. Conclusions

This paper addresses the implementation and use of dynamic boundary conditions (*DBC*) in OpenFOAM-7.0<sup>®</sup> for coupling 3D and 0D domains (3D0D). The implemented tool allows easy modelling of simple hydraulic components such as tanks, pipes, elbows and flow restrictions, as well as more complex components such as pumps and heat exchangers.

The stability of the method was guaranteed by the semi-implicit coupling given by upgrading of the boundary conditions inside of the pressure-velocity coupling PIMPLE algorithm. This had a greater advantage in contrast to the weak coupling methods based on the explicit linking.

Based on the several tests solve, the following conclusions are reached:

- a The 3D0D coupling strategy is simple, robust and efficient.
- b The 0D implementation of ad hoc code inside the *DBC* allows solving complex pipeline configurations, commonly found in industrial installations.
- c The use of *DBC* allows the introduction of the dynamic response of external systems into the 3D CFD domains without additional computational cost.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to thank Agencia Nacional de Promoción Científica y Tecnológica (PICT 2019-03750) as well as Consejo Nacional de Investigaciones Científicas y Tecnológicas.

## References

- [1] J.P. Tullis, *Hydraulics of Pipelines: Pumps, Valves, Cavitation, Transients*, John Wiley & Sons, 1989.
- [2] B.E. Larock, R.W. Jeppson, G.Z. Watters, *Hydraulics of Pipeline Systems*, CRC

- press, 1999.
- [3] D. Bestion, *System Code Models and Capabilities Section III*, IAEA Collections, 2008, pp. 81–106.
- [4] D. Pialla, D. Tenchine, S. Li, P. Gauthier, A. Vasile, R. Baviere, N. Tauveron, F. Perdu, L. Maas, F. Cocheme, et al., Overview of the system alone and system/cfd coupled calculations of the phenix natural circulation test within the thins project, *Nuclear Engineering and Design* 290 (2015) 78–86.
- [5] M. Aufiero, C. Fiorina, A. Laureau, P. Rubiolo, V. Valtavirta, Serpent-openfoam coupling in transient mode: simulation of a godiva prompt critical burst, *Proceedings of M&C+ SNA+ MC* (2015) 19–23.
- [6] C. Wang, H. Nilsson, J. Yang, O. Petit, 1d–3d coupling for hydraulic system transient simulations, *Computer Physics Communications* 210 (2017) 1–9.
- [7] R.P. Martin, Relap5/mod3 code coupling model, *Nuclear Safety* 36 (2) (1995) 290–298.
- [8] D. Aumiller, E. Tomlinson, R. Bauer, A coupled relap5-3d/cfd methodology with a proof-of-principle calculation, *Nuclear Engineering and Design* 205 (1–2) (2001) 83–90.
- [9] W. Li, X. Wu, D. Zhang, G. Su, W. Tian, S. Qiu, Preliminary study of coupling cfd code fluent and system code relap5, *Annals of Nuclear Energy* 73 (2014) 96–107.
- [10] J. Herb, Coupling Openfoam with Thermo-Hydraulic Simulation Code Athlet, 9th OpenFOAM Workshop, Zagreb (Croatia), 2014.
- [11] T.P. Grunloh, A. Manera, A novel domain overlapping strategy for the multi-scale coupling of cfd with 1d system codes with applications to transient flows, *Annals of Nuclear Energy* 90 (2016) 422–432.
- [12] D. Martelli, N. Forgiione, G. Barone, I. Di Piazza, Coupled simulations of the nacie facility using relap5 and ansys fluent codes, *Annals of Nuclear Energy* 101 (2017) 408–418.
- [13] W. Weaver, E. Tomlinson, D. Aumiller, An executive program for use with relap5-3d, in: BT-3394, 2001 RELAP5 Users Seminar, Sun Valley, Idaho, 2001.
- [14] D. Aumiller, E. Tomlinson, W. Weaver, An integrated relap5-3d and multi-phase cfd code system utilizing a semi-implicit coupling technique, *Nuclear engineering and design* 216 (1) (2002) 77–87.
- [15] D. Aumiller, F. Buschman, E. Tomlinson, D. Gill, Development of an integrated code system using r5exec and relap5-3d, *Nuclear Technology* 193 (1) (2016) 183–199.
- [16] G. Bandini, M. Polidori, A. Gerschenfeld, D. Pialla, S. Li, W. Ma, P. Kudinov, M. Jeltsov, K. Kööp, K. Huber, et al., Assessment of systems codes and their coupling with cfd codes in thermal–hydraulic applications to innovative reactors, *Nuclear Engineering and Design* 281 (2015) 22–38.
- [17] C.J. Greenshields, *Openfoam User Guide Version 7*, the openfoam foundation, 2019.
- [18] C. Fletcher, R. Schultz, *Relap5/mod3 Code Manual Volume V: User's Guidelines*, vol. 83415, Idaho National Engineering Laboratory, Lockheed Idaho Technologies Company, Idaho Falls, Idaho, 1995.
- [19] W. Ambrosini, N. Forgiione, J. Ferreri, M. Bucci, The effect of wall friction in single-phase natural circulation stability at the transition between laminar and turbulent flow, *Annals of Nuclear Energy* 31 (16) (2004) 1833–1865.
- [20] G. Lerchl, H. Austregesilo, *The Athlet Code Documentation Package*, vol. 1, *User's AZs Manual*, GRS-P, 1995.
- [21] W.M. Kays, A.L. London, *Compact Heat Exchangers*, 1998.
- [22] C. Hirt, B. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, *Journal of computational physics* 39 (2) (1981) 201–225.
- [23] J. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *Journal of computational physics* 100 (2) (1992) 335–354.
- [24] B. Launder, D. Spalding, The numerical computation of turbulent flows, *Computer Methods in Applied Mechanics and Engineering* 3 (2) (1974) 269–289.
- [25] G. Wang, B. Wang, J. Wen, R. Tian, Z. Niu, X. Liu, Experimental study on the hydraulic characteristics of inertia tank after the failure of pump power, *Annals of Nuclear Energy* 151 (2021), 107885.
- [26] M. Kaviany, *Principles of Heat Transfer in Porous Media*, Springer Science & Business Media, 2012.