Annals of Nuclear Energy 84 (2015) 244–257

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Coupled fine-mesh neutronics and thermal-hydraulics – Modeling and implementation for PWR fuel assemblies

Klas Jareteg [a,*], Paolo Vinai [a], Srdjan Sasic [b], Christophe Demazière [a]

[a] Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
[b] Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

ARTICLE INFO

Article history:
Received 31 March 2014
Accepted 20 January 2015
Available online 18 March 2015

Keywords:
Fine-mesh solver
Neutronics
Thermal-hydraulics
Parallelization
Coupled deterministic nuclear reactor modeling

ABSTRACT

In this paper we present a fine-mesh solver aimed at resolving in a coupled manner and at the pin cell level the neutronic and thermal-hydraulic fields. Presently, the tool considers Pressurized Water Reactor (PWR) conditions. The methods and implementation strategy are such that the coupled neutronic and thermal-hydraulic problem is formulated in a fully three-dimensional (3D) and fine mesh manner, and for steady-state situations. The solver is built on finite volume discretization schemes, matrix solvers and capabilities for parallel computing that are available in the open source C++ library foam-extend-3.0. The angular neutron flux is determined with a multigroup discrete ordinates method (S_N), solved by a sweeping algorithm. The thermal-hydraulics is based on Computational Fluid Dynamics (CFD) models for the moderator/coolant mass, momentum, and energy equations, together with the fuel pin energy equation. The multiphysics coupling is solved by making use of an iterative algorithm, and convergence is ensured for both the separate equations and the coupled scheme. Since all the equations are implemented in the same software, all fields can be directly accessed in such a manner that external transfer and external mapping are avoided. The parallelization relies on a domain decomposition which is shared between the neutronics and the thermal-hydraulics. The latter allows to exchange the coupled data locally on each CPU, thus minimizing the data transfer. The code is tested on a quarter of a 15×15 PWR fuel lattice. The results show that convergence is successfully reached, and correct physical behaviors of all fields can be achieved with a reasonable computational effort.

1. Introduction

To simulate the behavior of a nuclear reactor core, multiple fields of physics need to be considered. The distribution of the neutrons determine the amount of energy released by fission in the fuel. In a Light Water Reactor (LWR), the released energy is conducted through the solid fuel pins to the conjugate liquid (or vapor) water, and it is removed by the forced water flow. The water is not only acting as coolant, but also as moderator for the neutrons. In turn, the density of the water couples to the neutron distribution. Besides, the fuel temperature, which depends on the power and the coolant conditions, gives another feedback to the macroscopic neutron cross-sections via the Doppler effect. Other phenomena such as thermal expansion of the solid fuel and other core structures, fluid–structure interaction and material properties also impact the behavior of the core.

Furthermore, the reactor core is a multiscale system, with the scales ranging from the core width and height to the atomic ones governing nuclear reactions. For the neutronic solvers, the multiscale problem has typically been solved using a multistage procedure: the macroscopic cross-sections are generated in advance with a high order lattice solver and are employed for full core calculations performed with a low order coarse mesh solver. For thermal-hydraulics an equivalent multistage scheme can be seen in the use of subchannel codes and lower dimensional system codes, although such codes are not sequentially applied.

The multiphysics and multiscale can be tackled using a wide variety of schemes and methodologies. A splitting approach is often utilized, which consists of separate methodologies and codes for each field of physics and scale. The dependencies could then be regained using an a posteriori coupling, applying iterative schemes (for an overview see e.g. Ivanov and Avramova, 2007). Due to the split schemes, the couplings are usually only retrieved at the coarsest level, whereas the multiphysics coupling will take place at multiple scales. Therefore, these schemes might be inconsistent. This issue has recently gained a renewed interest, where the use of

* Corresponding author. Tel.: +46 317723077.
E-mail address: klas.jareteg@chalmers.se (K. Jareteg).

http://dx.doi.org/10.1016/j.anucene.2015.01.037
0306-4549/

more computational power has allowed for direct coupling at finer scales, and thus leading to more integrated approaches to the multiphysics problem (see e.g Gaston et al., 2009). Such methodologies typically rely on tightly coupled solvers, using implicit, non-linear techniques to handle the couplings. In other works, fine-mesh multiphysics couplings have been achieved using multiple codes, with a varying level of sophistication for the couplings (see e.g. Kochunas et al., 2012; Hamilton et al., 2013).

The large increase in computer resources has also allowed new applications of numerical simulations in the reactor core. Examples include integrated approaches for direct simulation of fuel material properties (Newman et al., 2009) and numerical simulations of resolved grid-to-rod fretting in a fuel assembly (Bakosi et al., 2013). High-fidelity simulations of a nuclear reactor that include all the mentioned aspects are challenging from a modeling perspective and are beyond the computational capacity available today. Instead, different approaches are still required for the different parts of the multiscale problem.

However, such insight does not imply that smaller scales (as compared to the core global scale) could be discarded. On the contrary, from a higher resolution coupled simulation, the models and correlations used for the coarse scale couplings can be evaluated and improved. In the view of the earlier statement, there are incentives to work with the coupled neutronic and thermal-hydraulic problem already on the fine-mesh level. In fact, a more detailed prediction of the interplay between the two fields can provide a better understanding of, e.g., the local temperature and power distributions in the fuel pins, as well as the conjugate heat transfer between the fuel and the coolant. Such and other fuel assembly parameters and aspects are important both from economical and resource utilization perspectives, and also from safety considerations.

The goal of this article is to describe a fine-mesh simulation framework that has been developed at Chalmers University of Technology, and that aims at reproducing the coupling between neutronics and thermal-hydraulics within PWR fuel assemblies. In particular, the solver can handle steady-state problems with respect to fine meshes, and it includes models for single-phase fluid dynamics, heat transfer in the solid and fluid regions, and neutron transport. We aim to give both specific example and also a more general overview of the key points for such a framework. The full coupled problem is solved using a single code approach, allowing fine-mesh direct multiphysics coupling.

The paper is structured as follows. The implications of the fine-mesh approach are described in Section 2, highlighting some of the key concepts and main issues to be tackled. The specific models used for the neutronics, the fluid dynamics and heat transfer are given in Sections 3 and 4, respectively. In Section 5, we describe the treatment in our methodology of the coupling and data transfer between the different fields. In Section 6, strategies for the parallelization of the coupled system are outlined and discussed. The described models and implementations are then tested on a simplified 15×15 PWR system and reported in Section 7. Finally, a summary and a number of important conclusions are given in Section 8.

2. Fine-mesh considerations

As mentioned above, the target of this work is to couple the neutronic and thermal-hydraulic fields on a sub-pin level. Following the choice of resolution, limitations and approximations will be imposed for other scales. With the chosen level of detail, full core calculations are extremely computationally expensive. Instead, the fine-mesh approach is applied to the fuel assembly calculations, where the computational burden is still relatively large, but can be handled with the available computer resources.

The application of a fine-mesh approach gives a possibility of a full three-dimensional representation of the fuel assembly. This fact is beneficial for two main reasons. First, the geometrical complexity of the fuel assembly can be taken in account with a better resolution. Second, physical phenomena can be better captured, avoiding one dimensional (1D) or two dimensional (2D) approximations that may be questionable.

To preserve the geometry on the sub-pin cell scales in an efficient manner, an unstructured computational mesh is required. This allows for a correct representation of the fuel pins and other structural parts and avoids homogenization of the simulated system. The choice of resolution of the mesh influences the discretization method applied to models and equations. In this work the discretization is based on the Finite Volume Method (FVM). FVM allows unstructured meshes to be used, and ensures local conservation on each cell (Ferziger and Peric, 2002). It is also a well proven method for neutronics and fluid problems, and is a standard practice for Computational Fluid Dynamics (CFD) codes.

To handle the fine-mesh requirements, the use of unstructured meshes, and the coupling of multiple fields of physics, a versatile computational code is necessary. Such a tool could consist of multiple specialized softwares, combined using an external coupling (see e.g. Cardoni and Rizwan-uddin, 2011; Kochunas et al., 2012). An advantage of this scheme is the possibility to use a set of validated tools, limiting the amount of new code to be implemented and tested. However, the external transfer approach introduces not only an expensive overhead, but it also limits the resolution of the coupling and imposes constrains on the coupling algorithms. An alternative to the split software scheme is to include all fields of physics and the coupling in the same code (see e.g. Gaston et al., 2009), which is the approach followed in this work. This allows for a coupling directly on the finest scale, avoiding the computational and methodological penalties associated with external or a posteriori coupling schemes. Further benefits will be discussed in Section 6, where the integrated approach is described, and also used to handle the parallelization in an efficient manner.

To benefit from an existing code, we use the open source C++ library foam-extend-3.0 as a base for the coupled code. foam-extend-3.0 is a fork of OpenFOAM®, earlier named OpenFOAM®-dev and OpenFOAM®-ext (Wikki, 2014). The software gives access to a high performance library with a flexible code with many different applications. In particular, full availability of the source code allows for extending the code at any level, including implemented equations, physical models as well as for the linear solvers and the discretization schemes.

As a consequence of the chosen resolution and using a 3D representation of the system, the computational requirements for this kind of effort can not be overlooked. The fine-mesh approach must be solved using high performance computations (HPC), including fast programming languages, efficient algorithms and use of parallelization to efficiently tackle the problem. Furthermore, the problem is parallelized using the Message Passing Interface (MPI) (MPI Forum, 2009) as implemented in foam-extend-3.0 (further described in Section 6).

3. Neutronics for the sub-pin cell calculations

For the type of system and resolution under study, the selected neutronics solver methodology needs to handle the strong material heterogeneities and the angular dependence. In the case of the sub-pin neutronic calculations, the diffusion approximation is not an accurate choice. Instead a full transport methodology is required. Here, we use the discrete ordinates method, formulated for a general unstructured mesh and using the finite volume method to discretize the equations. We solve the steady state neutronic eigenvalue problem using the power iteration method.

3.1. S_N method

The discrete ordinates method is based on solving the neutron transport equation on a set of directions (or ordinates). The equation to be solved for each separate direction is given by (Larsen and Morel, 2010):

$$\Omega_m \cdot \nabla \Psi_{m,g} + \Sigma_{T,g} \Psi_{m,g} = S_{m,g} + \frac{1}{k} F_{m,g} \tag{1}$$

where the anisotropic scattering source term is given by:

$$S_{m,g} = \sum_{l=0}^{L} (2l+1) \sum_{m'=1}^{M} P_l(\Omega_m \cdot \Omega_{m'}) w_{m'} \sum_{g'=1}^{G} \Sigma_{s,l,g' \to g} \Psi_{m',g'} \tag{2}$$

which is formulated in terms of the Legendre polynomials (P_l) and the ordinate weights (w'_m). The fission source term in terms of the angular flux reads:

$$F_{m,g} = \chi_g \sum_{m'=1}^{M} w_{m'} \sum_{g'=1}^{G} \nu_{g'} \Sigma_{f,g'} \Psi_{m',g'} \tag{3}$$

The scalar flux can then be retrieved by performing the weighted sum over all the different directions such that:

$$\Phi_g = 4\pi \sum_{m}^{M} w_m \Psi_{m,g} \tag{4}$$

The fission source with respect to the scalar flux will read:

$$F_{m,g} = \frac{\chi_g}{4\pi} \sum_{g'=1}^{G} \nu_{g'} \Sigma_{f,g'} \Phi_{g'} \tag{5}$$

Standard notations are employed for all quantities.

We assume the most general case with a fully anisotropic scattering. Thus, in each inner iteration while solving Eq. (1) (i.e. for each energy group (g) and each ordinate m), the full dependence of all other groups and directions will enter through the scattering term $S_{m,g}$. To reduce the computational cost of evaluating the scattering source, an expansion of the angular flux on real spherical harmonics is used, such that:

$$S_{m,g} = \sum_{g'=1}^{G} \sum_{l=0}^{L} (2l+1) \sum_{r=-l}^{l} R_{lr} \phi_{g,l,r} S_{s,l,g' \to g} \tag{6}$$

where $\phi_{g,l,r}$ are the expansion coefficients and R_{lr} are the real spherical harmonics as given in Hébert (2010).

The choice of directions (Ω_m) and weights (w_m) for S_N can potentially have a large influence on the accuracy of the solution, as discussed by e.g. Abu-Shumays (2001). The directions and weights in this work are based on the level symmetric quadrature set as given in Hébert (2010).

3.2. Solution and parallelization of S_N

Due to the nature of Eq. (1), the discretized equation can be solved with a sweeping algorithm that corresponds to the solution of a lower diagonal matrix using a Gauss–Seidel method, and based on the ordinate direction Ω_m. For an unstructured (as well as a structured) mesh such sweeping order can be found starting from an inlet boundary and iteratively transversing the cells of the mesh (Plimpton et al., 2005). Cyclic dependencies are avoided if only convex cells are used, which is the case in this work. A brief outline of the applied sweep order methodology is given in Fig. 1.

When the problem is parallelized by splitting the space in different domains (as further discussed in Section 6.3), the sweeping order also needs to be parallelized. In order to conserve the single sweep over the mesh, each separate domain must wait for neighboring upstream domains to finish their sweeps. To still utilize the time waiting, the different domains should optimally start with different directions. Such algorithm is not implemented in the current work. Instead all domains are concurrently sweeping for the same direction. This will introduce a penalty for the parallelized version of the discrete ordinates solver. However, the used algorithm is easier both to implement and to maintain in terms of generalization and implementation.

Much effort has been spent on the development of acceleration techniques for the discrete ordinates method (see e.g. Larsen and Morel, 2010). Nevertheless, no acceleration is used in this work, and the problem is instead solved by iteratively updating the directions, group by group.

An outline of the applied solution methodology is given in Fig. 1. During the initialization, the decomposed mesh is read by each CPU, whereafter the sweeping order is determined. Finally, the cross-section sets are determined (see Section 3.3). Each neutronic iteration starts with an update of the cross-sections based on the current temperature distribution in the system. Thereafter an outer iteration is used to update the eigenvalue problem, using an inner iteration to update the angular flux. After convergence of the outer iteration, the power profile is updated. Optionally, we use a diffusion solver during the first iteration. This allows an approximate fission source and criticality value to be calculated with a smaller computational cost, which can be used to increase the acceleration and convergence of the S_N solver.

3.3. Cross-sections and macroscopic data

To close Eq. (1), cross-sections and macroscopic data are needed. Such data are here generated using a Monte Carlo approach. We use the software Serpent (Leppänen, 2012), to create a set of two-dimensional condensed and homogenized cross-sections. The details of this approach can be found in Jareteg et al. (2014b).

To match the resolution of the system, sub-pin cell dependent cross-sections are needed. The data are thus generated for regions split azimuthally as well as radially and tabulated according to position. Furthermore each specific radial and azimuthal region is run for a set of temperatures, as to give a temperature-dependent macroscopic data in all regions of the fuel.

The radial and azimuthal discretization used in this work can be seen for a quarter of a 15×15 fuel pin array in Fig. 2. The regions displayed in Fig. 2 are automatically mapped to an unstructured mesh of the deterministic calculations (as exemplified in Fig. 3). In this manner the code is kept very general, and more physics and different geometrical structures can be easily added later.

The scheme followed is given in Fig. 3. The first part, covering the generation of the cross-sections, takes place before the coupled deterministic calculations, and it provides geometrical sets and cross-section tables for each region as displayed in Fig. 2. The produced sets are then used to calculate a mapping based on the geometrical information generated by the Python script and the cell centers in the unstructured mesh. Once a mapping is determined for each cell in each CPU, the correct table can be read. Each time the cross-sections are updated (as seen in Fig. 1), the mapping information is re-used, and the cross-sections updated locally for each cell.

4. Thermal-hydraulic model for the fine-mesh calculations

In this work the thermal-hydraulic problem is solved using mass, momentum and energy equations, all formulated from first principles. The equations and the models applied are aimed at the sub-pin cell resolution, and are consistent with the neutronic calculations. This means that the moderator gradients of velocity, density and temperature between the fuel pins and the temperature gradient within the fuel pins must all be resolved. The

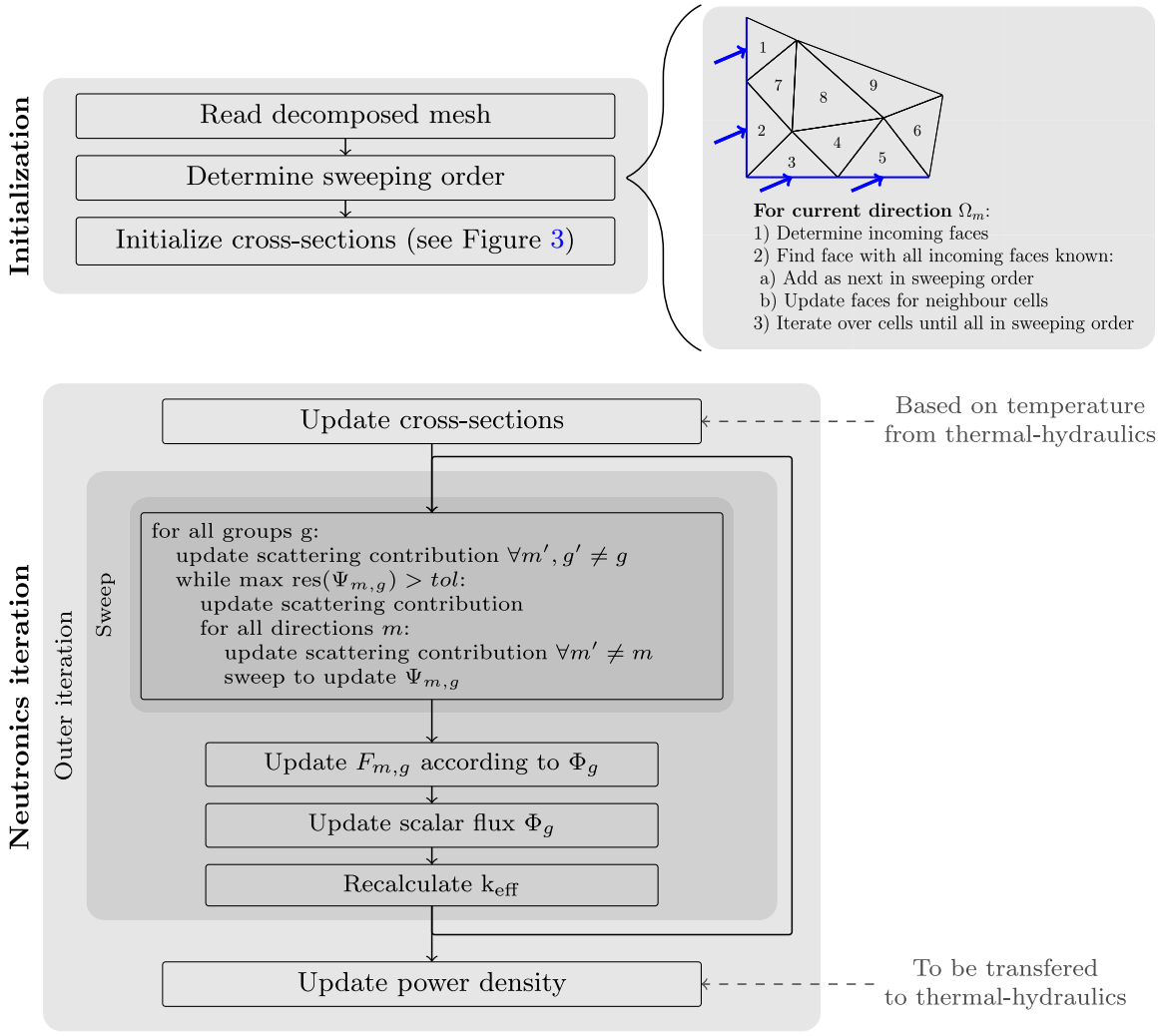

Fig. 1. Applied algorithm for the discrete ordinates method.

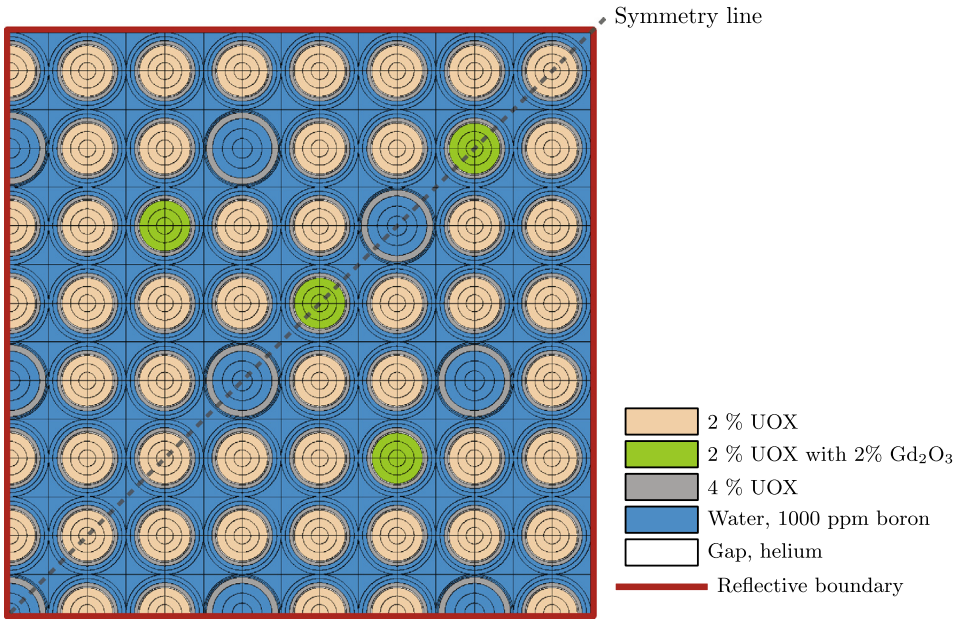

Fig. 2. Fuel pin discretization in horizontal plane, using 4 azimuthal and 8 radial regions per pin cell, in total 1775 regions.

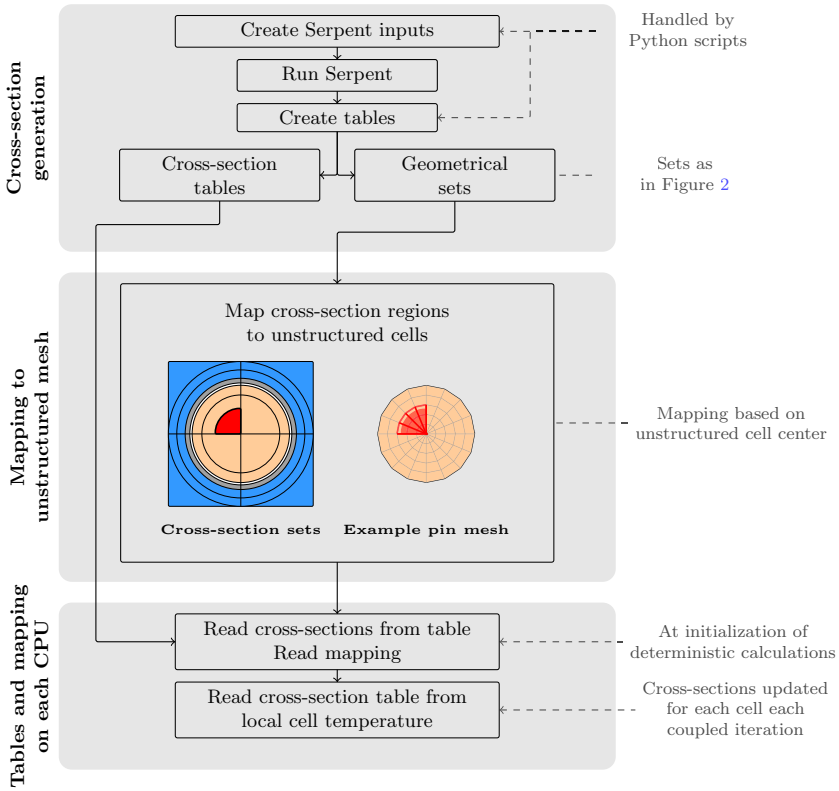

Fig. 3. Mapping scheme for the cross-sections.

approach followed here is thus more representative for CFD than for typical lower dimensional codes, which are to a larger extent based on empirical correlations.

Using a CFD methodology introduces new type of equations and solvers, both with the aim to reduce the reliance on measured or approximated quantities and to reach a higher resolution. As an example, in this work a two equation turbulence model is solved on the same mesh resolution as the momentum equation. This, in combination with the wall boundary conditions, allows for the pressure drop over the channel to be directly calculated, avoiding the use of pressure drop correlations.

The work is limited to pure single phase liquid systems, i.e. at this stage the existence of multiple phases encountered in Boiling Water Reactors (BWRs) and in the subcooled boiling region in PWRs are not taken into account. Such gas–liquid heated problem is however the focus of the next developmental stage.

Even though a single phase methodology is applied, the heat transfer from the fuel pins will still lead to a change in density in the moderator and thus also influence the momentum and mass conservation equations. This is accounted for by solving the full conjugate heat transfer problem, including the moderator, the cladding encapsulation, the gaps and the fuel pins, in one fully coupled system.

4.1. Pressure–velocity solver

The momentum equation is formulated for the steady state problem and by time-averaging the Reynolds decomposed velocity. This results in a filtered momentum equation such that:

$$\nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = \nabla \cdot \bar{\bar{\tau}} - \nabla \cdot \overline{\rho \mathbf{u}' \otimes \mathbf{u}'} - \nabla P + \rho \mathbf{g} \tag{7}$$

with the mean velocity given by $\mathbf{U}$, the fluctuating contributions given by $\mathbf{u}'$ and with the density ρ, the pressure P and the gravity $\mathbf{g}$. To close the momentum equation, the stress tensor (τ_{ij}) is modeled as (Panton, 2005):

$$\tau_{ij} = \mu\left(U_{i,j} + U_{j,i} - \frac{2}{3}U_{k,k}\delta_{ij}\right) \tag{8}$$

where μ is the viscosity. The velocity fluctuation term is related to the stress tensor and a kinetic viscosity according to:

$$-\overline{\rho u_i' u_j'} = \mu_t\left(U_{i,j} + U_{j,i} - \frac{2}{3}U_{k,k}\delta_{ij}\right) - \frac{2}{3}\rho k \delta_{ij} \tag{9}$$

where k is the turbulent kinetic energy which is computed based on a two-equation turbulence model, here formulated as (see e.g. Ferziger and Peric, 2002):

$$\nabla \cdot (\rho k \mathbf{U}) = \nabla \cdot \left(\mu + \frac{\mu_t}{\sigma_k}\nabla k\right) + \mu_t(\nabla \otimes \mathbf{U}) : (\nabla \otimes \mathbf{U} + (\nabla \otimes \mathbf{U})^T) - \rho\epsilon \tag{10}$$

where σ_k is a model constant and with the turbulent dissipation ϵ calculated by the second equation:

$$\nabla \cdot (\rho \epsilon \mathbf{U}) = \nabla \cdot \left(\mu + \frac{\mu_t}{\sigma_\epsilon}\nabla \epsilon\right) + C_{1\epsilon}\frac{\epsilon}{k}\mu_t(\nabla \otimes \mathbf{U}) : (\nabla \otimes \mathbf{U} + (\nabla \otimes \mathbf{U})^T) - C_{2\epsilon}\rho\frac{\epsilon^2}{k} \tag{11}$$

where $\sigma_\epsilon, C_{1\epsilon}$ and $C_{2\epsilon}$ are model constants. After solving the turbulent kinetic energy and the turbulent dissipation, Eqs. (10) and (11), the turbulent kinetic viscosity in Eq. (9) is computed as:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{12}$$

Finally, the time-averaged, steady state mass equation reads:

$$\nabla \cdot (\rho \mathbf{U}) = 0 \tag{13}$$

The momentum and mass equations are used to solve for the coupled pressure and velocity distribution in the moderator. Such a formulation is typical for incompressible flow CFD calculations (see e.g. Ferziger and Peric, 2002). To resolve the steady state coupling, the SIMPLE algorithm is applied (Patankar and Spalding, 1972). This algorithm relies on first solving a momentum predictor step based on Eq. (7), resulting in an estimated velocity field. The momentum equation is then reformulated and inserted in the mass conservation equation, now solved for the pressure field, based on the predicted velocity field. Finally, the momentum equation is updated from the newly calculated pressure field.

4.2. Conjugate heat transfer

Since the moderator acts also as the coolant, the thermal coupling to the fuel pins must be taken into account. The flow pattern has an impact on the coolant ability to extract the heat at the wall of the cladding and the heating of the moderator causes a change in the density leading to a change in the flow.

The regional dependence of the conjugate heat transfer can be solved either based on an iterative approach or a monolithic, implicit approach. Using the iterative approach, each region or a group of regions are computed separately. The dependence is then handled by imposing alternating Dirichlet and Neumann boundary conditions. For the system here simulated, consisting of a quarter of a 15×15 fuel assembly, with the gap and cladding explicitly modeled, in total 184 different material regions are encountered. The high number of regions makes the system unattractive to solve in an iterative manner. If, instead, formulating and computing the full system in an implicit manner, the couplings between the regions are treated implicitly in the equation system. Such an approach, applied in this work, avoids the iterations between the regions.

We express energy conservation in terms of temperature. This allows to directly couple the equations on the faces of the cells at the region boundaries. The moderator temperature conservation equation is derived from the energy conservation equation considering steady state conditions only and applying a Reynolds decomposition to filter the rapid fluctuations. The procedure followed is further described in (Jareteg et al., 2014a), and the final equation will read:

$$(\rho c_p(T))\mathbf{U}\cdot\nabla T = \beta(T)\mathbf{U}\cdot\nabla P + \nabla\cdot(K_{\text{eff}}(T)\nabla T) \tag{14}$$

where c_p is the specific heat capacity, β is the thermal expansion coefficient and K_{eff} is the effective thermal conductivity that includes the heat transfer enhancement from turbulence.

The temperature equation for the solid regions is derived by applying Fourier's law to the heat transfer equation, such that:

$$-\nabla\cdot(K(T)\nabla T) = q''' \tag{15}$$

where K is the thermal conductivity of the material and q''' is the power density, that is only non-zero in the fuel and is given by the energy released per fission and the computed neutron flux. The heat transfer in the gap is solved using Eq. (15), neglecting the radiation heat transfer.

4.3. Solver methodology

An outline of the combined pressure–velocity and conjugate heat transfer algorithm is given in Fig. 4. In the first stage the momentum predictor and the turbulence equations are solved for the moderator. This is followed by the implicit solution of the conjugate heat transfer problem between all solid material regions and the moderator. Given the new temperature field, all thermophysical properties are updated. Finally, the effect of buoyancy is accounted for using the Boussinesq approximation (Ferziger and Peric, 2002), the pressure equation is solved and the momentum is calculated according to the new pressure. The scheme presented in Fig. 4 corresponds to one sub-iteration in the thermal-hydraulics.

In the applied methodology no explicit thermal expansion in the fuel is treated. The influence of such an approximation will be considered in later work.

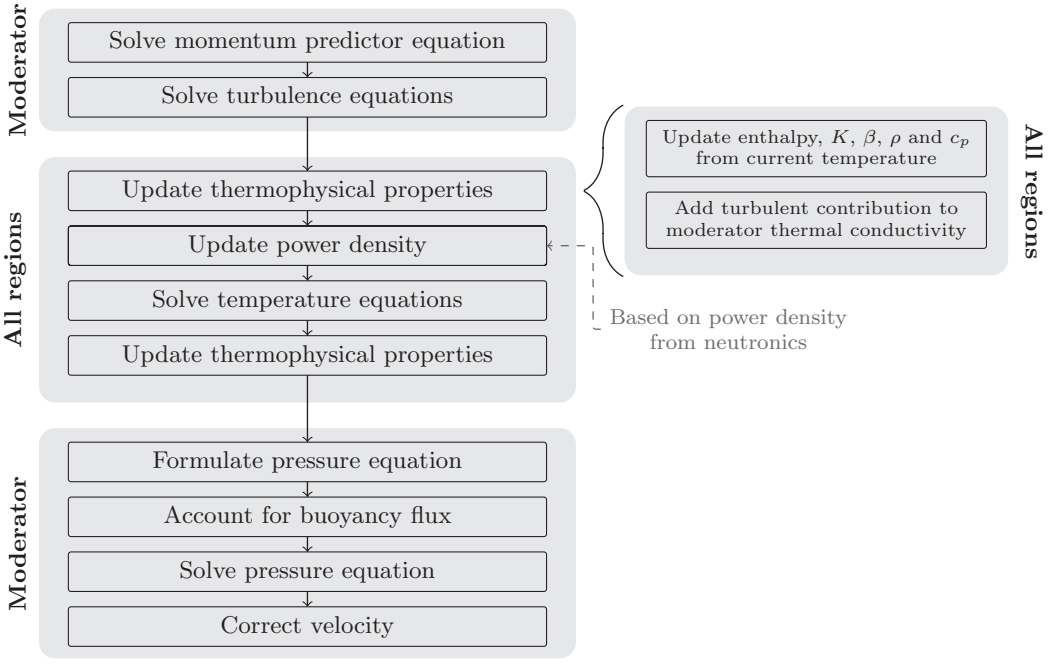

Fig. 4. Thermal-hydraulics solver methodology.

5. Multiphysics treatment and coupled aspects

Based on the separate models for the neutronics and thermal-hydraulics presented in the previous Sections 3 and 4, the aspects related to coupling of those phenomena will be discussed hereafter. Focus will be given on the discretization of the separate fields, on the scheme used for a consistent and conservative spatial mapping and on the iterative algorithm for resolving the multiphysics dependencies.

5.1. Mesh resolutions

In order to calculate accurate solutions for all separate fields of the problem, we should apply different computational meshes. Considering, for example, the different characteristic length scales present in the neutron angular flux and the near wall turbulent flow in the moderator, using the same mesh for all fields is not an efficient solution procedure. Instead of a common resolution, the mesh for each separate field should be optimized for the type of physics encountered. In the example mentioned, the near wall flow will (based on our models) require a finer mesh close to the fuel pins than for the neutron flux, so that a comparable precision can be obtained for the simulated system.

The application of different meshes also allows the computational effort to be optimized. Again, it would be a waste of resources to perform the neutronic calculations directly on the fluid flow mesh. Each part of the problem will thus be discretized with a mesh in such a way that a consistent level of detail is reached for the coupled problem. In this work the mesh for the moderator is chosen to have twice the number of cells in the azimuthal direction as compared to that for the solid heat transfer and the neutronics, as displayed in Fig. 5. In general, the validity of mesh resolution should be judged both from the grid dependence of the separate fields as well as from the grid dependence for the coupled problem.

The foam-extend-3.0 internal mesh format applied in this work primarily relies on mesh faces with two connecting cells. However, when solving the conjugate heat transfer problem, a single temperature equation is desired even in the case that two moderator faces meet a single cladding face (see Fig. 5). Since the discretization of the solid regions is more coarse than the moderator discretization, a matrix level mapping of the faces is required. This is handled with the general grid interface routines available in foam-extend-3.0 (Jasak, 2009).

Whereas the specific implementation and choice of resolution and mesh will depend on the chosen framework, the importance of such elements should be stressed. Again, a larger flexibility as well as a better consistency, are reached using a single framework for all calculations, as exemplified here by the seamless use of different mesh discretizations in the solid regions and the moderator region.

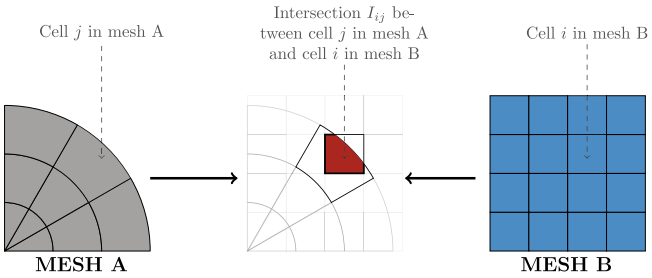

Fig. 6. Example of mapping of two overlapping meshes.

5.2. Consistent spatial mapping

As a consequence of using multiple meshes, a consistent and conservative mapping scheme is needed for overlapping meshes. As regards the mapping of the power density from the neutronic to the thermal-hydraulic fuel pin mesh, the correct energy released by fission is needed in the heat transfer problem given by Eq. (15). In the same manner, a correct thermophysical state must be mapped from the thermal-hydraulic to the neutronic mesh, in order to accurately update the cross-sections.

The mapping scheme here implemented is based on finding volumetric overlaps between cells in the different meshes. We apply a collocated finite volume method, where the cell values are calculated for all quantities. Since these cell values are considered constant over the cell, an exact and conservative mapping can be achieved if the volumetric overlaps between the cells are known. An example is seen in Fig. 6. Given a cell j in mesh A and a cell i in mesh B, the overlap is calculated using a polygon intersection algorithm in the horizontal plane and a direct overlap in the axial direction. The polygonal intersection algorithm relies on the Sutherland–Hodgman algorithm (Sutherland and Hodgman, 1974), as earlier implemented in foam-extend-3.0 (Page et al., 2010). The calculation of the overlap in the axial direction relies on all cells having parallel faces in the axial direction. This is assured by using only hexahedron and prism elements.

Given the calculated volumetric intersections I_{ij} for all cells j intersecting with cell i, an extensive property c can be transferred from mesh A to mesh B as follows:

$$c_i = \sum_j \frac{c_j I_{ij} V_j}{V_i} \tag{16}$$

where V_i and V_j are the cell volumes of cell i and cell j, respectively.

The implemented scheme is fully automatic and the calculations are performed at the initialization of the code, completely removing any hard coding or manual calculation of the mappings between the neutronic and the thermal-hydraulic meshes. This also means that the meshes can be changed without any further intervention in the code or in any input files. In order for the map-

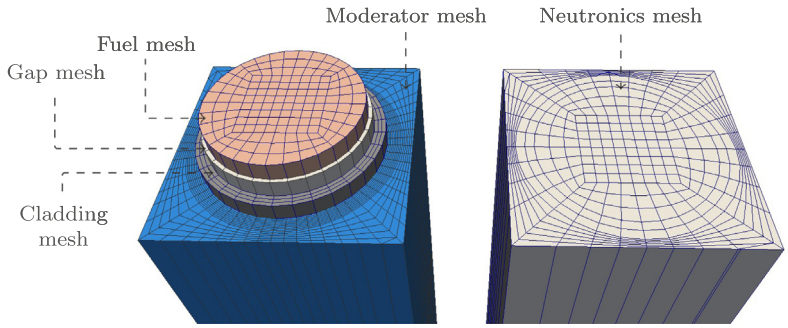

Fig. 5. Horizontal mesh discretization exploded for the thermal-hydraulics (left) and the neutronics (right).

ping to occur correctly, the material boundaries should, however, be preserved between all the meshes. If this is not the case, the transferred quantity will not be interpreted correctly in the receiving cells.

In the chosen mapping scheme based on intersection calculations and not on point interpolation, only extensive quantities should be transferred. Since temperature is not an extensive quantity, the enthalpy is used in the transfer from the thermal-hydraulic cells to the neutronic cells.

The mapping utility also opens for other fields of physics to be coupled to the problem, all potentially using separate meshes. This can further be exploited to introduce different scales of the same physics. A coarse neutron solver can be applied on its own mesh, then automatically coupled to the fine-mesh solver.

5.3. Coupling scheme

The multiphysics coupling can be resolved using either a fully implicit technique, an explicit approach or a combination of both. Whereas the fully implicit techniques are based on combining the fields in a non-linear problem, the fully explicit techniques will rely on iteration between the different fields. Potentially a non-linear scheme will increase the convergence rate and possibly also reach convergence for systems not possible to solve in a segregated manner (Gaston et al., 2009). However, the implicit approach will also increase the memory usage, and will require implementation of good preconditioners for each separate field of physics (see e.g. Zou et al., 2013).

We apply an explicit approach in this work, where we solve each field in a separate manner. However, the coupled dependencies are fully resolved, iterating between the different physics. In the steady state problem solved in this paper, it will be shown that reaching a global convergence is not a major problem. Further, the number of iterations between the thermal-hydraulics and the neutronics are few, and thus the incentive for the non-linear approaches are limited. In the presented case, acceleration of the neutronics and thermal-hydraulics problems separately would be of higher priority as future developments.

As we address each part separately, the independence between the modules is increased, which also simplifies the development of each separate module and gives a larger freedom to the solution procedure of each field. Nevertheless, in future development, approaching transient simulations and adding more physical phenomena, an outermost non-linear solver might be beneficial or even necessary.

The iterative procedure followed in this work can be seen in Fig. 7. After initialization of all fields the neutronics is solved first. Given the new power profile the thermal-hydraulics is then solved, resulting in a new temperature and enthalpy field.

During the first full iteration, only the diffusion equation neutronics is solved (as described in Section 3.2). In the second iteration, the thermal-hydraulics is processed, without the temperature equation, but only pressure and velocity. Finally, in the third iteration all fields are calculated. Consequently, the fourth iteration is the first iteration with the cross-sections updated according to a new temperature profile. Such a procedure allows the problem to converge in a more stable manner, not starting the full coupled problem only with the initial guesses.

Sub-iterations are used both in the neutronics and thermal-hydraulics. In earlier work such sub-iterations were avoided (Jareteg et al., 2013). It was, however, found that an overall faster convergence was reached if all fields were allowed to converge within each outermost coupled iteration (Jareteg et al., 2013). Yet, a maximum number of sub-iterations are deployed for both neutronics and thermal-hydraulics, since full convergence is still not necessary until the full coupled problem has converged.

6. Efficiency and scalability; implementation and parallelization

Due to a large number of degrees of freedom in the fine-mesh calculations, a severe computational cost needs to be tackled already for a single fuel assembly. Consequently, the code implementation and the use of high performing computer languages are both of significant importance. This is true for the separate models as well as for the coupling scheme. Furthermore, we must make it possible for the full problem to be parallelized in a scalable manner.

6.1. Model and equation implementation

The separate parts of the multiphysics problem are all implemented using foam-extend-3.0. The library foam-extend-3.0 is based entirely on C++ programming language and contains a large set of solvers for different type of problems in fluid mechanics as well as heat transfer. For the implementation of the neutronics, the existing internal structure is used for the equation discretization, for the equation solvers, and for all general structures such as matrices and meshes.

When selecting a framework, or writing a completely new, the performance of the code is not the only point of concern. In fact, the code should be easy to extend and it should be written in such a way that it is simple to maintain. Also from this aspect foam-extend-3.0 is a seemingly viable choice. The code uses modern

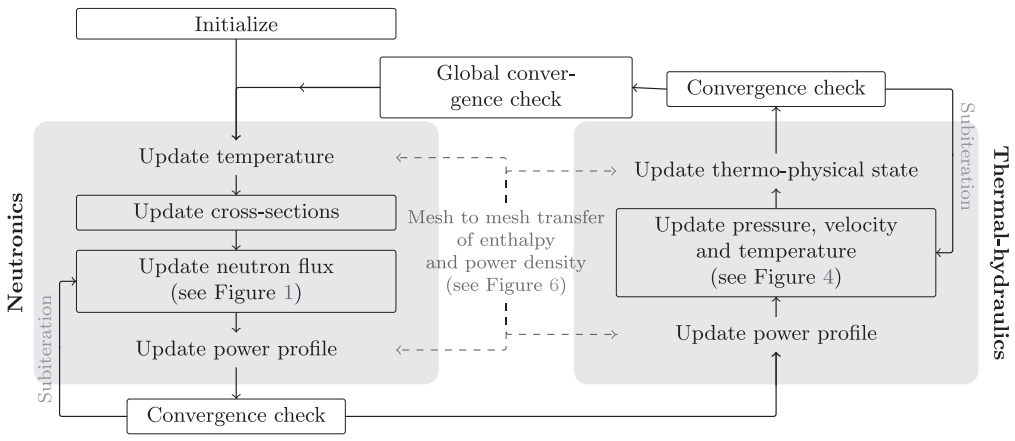

Fig. 7. Iterative scheme applied for the coupling of the thermal-hydraulics and the neutronics.

C++ functionalities such as templates and polymorphism to allow for easy and convenient extensions and modifications to the code.

The code foam-extend-3.0 contains a high level equation format, which makes possible a fast implementation of additional equations. This is used for the heat transfer equations, (14) and (15), whereas the pressure and velocity solver is based directly on the existing solvers. For the cross-section formalism, the mapping algorithm, and the sweeps for the discrete ordinates method, more extensive coding work was needed.

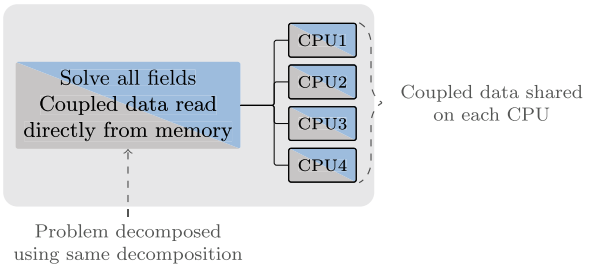

Fig. 9. Example of single code coupling scheme avoiding external transfer.

6.2. Coupling implementation

For the multiphysics couplings, different potential approaches were described in Section 5.3. The implementation of the coupling method could be based on different schemes. Irrespective of whether an implicit, non-linear coupling or an iterative segregated approach is used, two setups could be followed. Either all equations are solved in the same code entity (a single code approach) or multiple different softwares (multiple codes approach) address different subsets of the equations.

Using the multiple codes implementation, an external transfer of information is necessary to exchange the coupled states. In the present case, the power density would be passed from the neutronics to the thermal-hydraulics, while the thermophysical state would go in the reverse direction. In many cases such data transfer can severely reduce the efficiency of the code and take a major part of the program execution time (see e.g. Yan et al., 2011).

Further, the possibilities to obtain efficient parallel couplings between the neutronics and thermal-hydraulics are limited. In many approaches two different softwares are used, each performing its own parallelization. The communication is in such a case often handled by a central application, running on a single CPU, and therefore all the information must be gathered to and shared via this CPU. An example of the multiple codes scheme can be seen in Fig. 8. Typically, a coupling application is placed between the different codes, handling the transfer of data (see e.g. Kochunas et al., 2012). This type of coupling is especially limiting for transient calculations where a high number of transfers between the codes is necessary.

In the single code approach, that is chosen in the present work, all parts of the problem are solved in the same code. This spares not only the external transfer but raises possibilities for a more efficient parallelization. Avoiding any external transfer of data via IO also permits a higher resolution of the coupling since the overhead is basically removed. An example of such a scheme is given in Fig. 9. The transfer of information can now be done by direct memory access. The iterative algorithm can also be switched between the different fields without the overhead cost of the multiple codes approach, which is also typically needed for the time stepping in a transient calculation.

6.3. Parallelization by shared domain decomposition

To parallelize the problem, different strategies could be followed. The domain could be split in different spatial regions, corresponding to a domain decomposition, or the problem could be split in its different fields of physics each solved on a separate set of CPUs, corresponding to a functional decomposition (Calvin and Nowak, 2010). Equivalent to multiple codes parallelized on separate CPUs (as in Fig. 8 and discussed in Section 6.2), the coupled data information transfer in a functional decomposition becomes an obstacle. If the parallelization is instead based on a common domain decomposition for all fields, as displayed in Fig. 9, the exchange of information will take place locally for each CPU, minimizing the amount of data that needs to be updated between CPUs. Such an argument is of particular importance since the fine-mesh direct coupling results in a large quantity of data to be shared.

Applying the domain decomposition method, only the data at the domain interfaces need to be exchanged during the solution of the equations. While the foam-extend-3.0 structure for parallelization is employed, the information transfer is handled according to a zero-halo layer approach, i.e. there is no overlapping between the decomposed domains. In practice, the boundary data exchange occurs only at the point of the sparse matrix solution, or prior to a sweep in the neutronics (corresponding to prior to "`sweep to update` $\Psi_{m,g}$" in Fig. 1).

To get the same decomposition for the neutronic and thermal-hydraulic meshes, the geometry is in this case decomposed along planes parallel to x, y and z-axis of the system. In general, the decomposition should be done such that the amount of work to

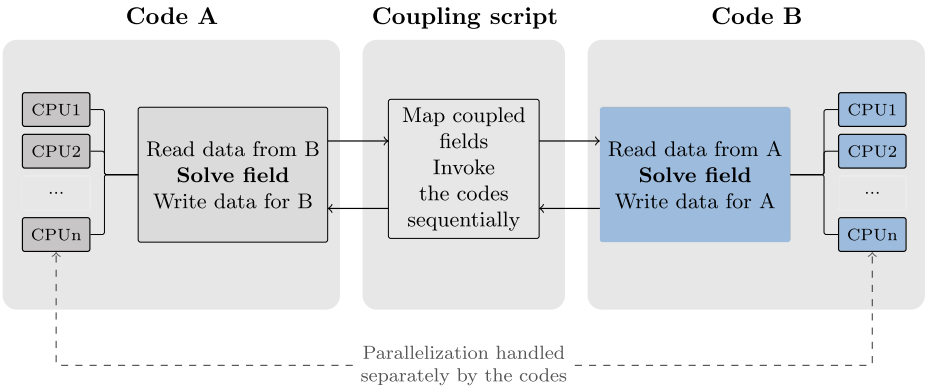

Fig. 8. Example of data transfer in the multiple codes approach.

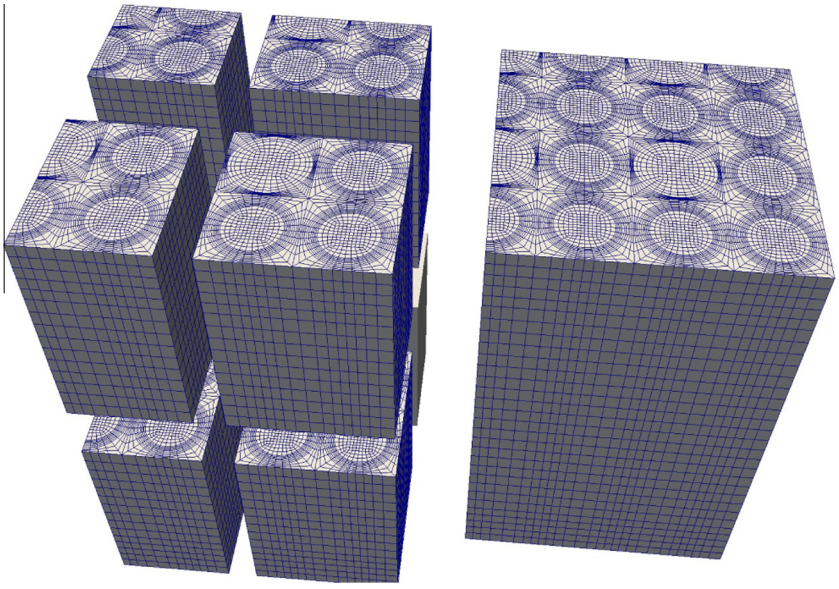

Fig. 10. Example of decomposition for a quarter of a 7×7 pins lattice.

Table 1
Geometry specification for the simulated assembly, with control rod guide tube values in brackets.

Fuel pin radius	0.41 cm
Cladding inner radius	0.43 cm (0.48 cm)
Cladding outer radius	0.49 cm (0.58 cm)
Pitch	1.25 cm
Fuel height	100 cm
Bottom reflector	20 cm
Top reflector	20 cm

Table 2
Mesh specification for the simulated assembly.

Region	Number of cells
Moderator	6,088,000
Fuel (per pin)	8000
Cladding (per pin)	4800
Gap (per pin)	1600
Neutronics	798,000

Table 3
Selected boundary conditions and inlet conditions.

Quantity	Initial condition	Inlet	Outlet
Temperature	540 K	540 K	Zero gradient
Pressure	15 MPa	Zero gradient	15 MPa
Velocity	2 m/s $\hat{z}$	2 m/s $\hat{z}$	Zero gradient
Neutron angular flux	1.0	Outgoing direction: zero gradient Incoming direction: fixed value zero	

Table 4
Neutronic parameters.

N (Eq. (1))	8
G (Eq. (1))	8
L (Eq. (6))	2
Total power	400 kW

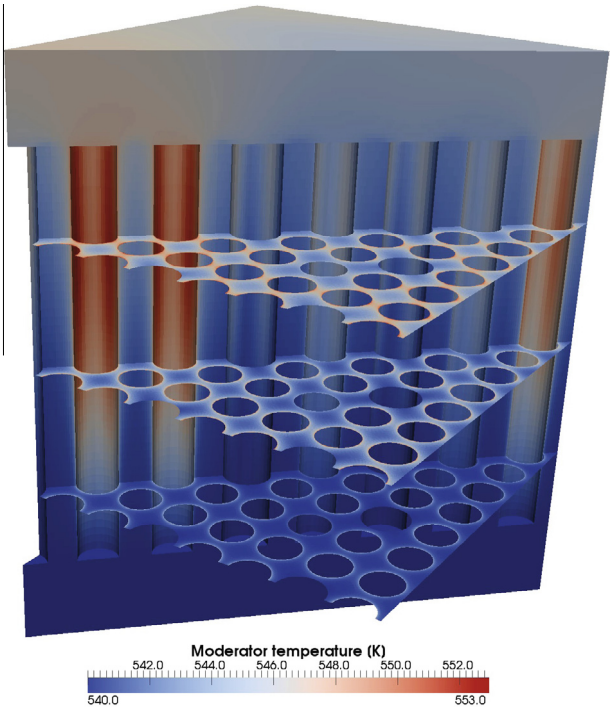

Fig. 11. Moderator temperature at three horizontal planes, with the axial dependence at a diagonal cut in the background.

be done by each CPU is equivalent. Since the loading will potentially be different for different equations, there must be a compromise. The shared decomposition restricts all used meshes to have faces along the chosen lines for splitting, so that every cell in all meshes exists entirely on a single CPU. In the case of the regular pattern of a fuel bundle this condition is not a major restriction. An example of the decomposition can be seen in Fig. 10 for a 7×7 fuel pins system. The example system is split between the fuel pins and at mid-elevation.

7. Application and results

To illustrate the capabilities of the methods and implementations described above, we apply the developed code to a quarter of a 15×15 fuel lattice, with reduced height with respect to a fuel assembly. The steady-state problem is solved for conditions typical of a PWR.

7.1. Geometry and mesh

The geometry of the simulated system is specified in Table 1, and the material composition in the horizontal plane corresponds to the presented case in Fig. 2. No spacers are included in the simulated system. Reflectors are modeled at the bottom and the top of the assembly, here consisting of moderator only without any other structural parts, as the inlet orifice and the top nozzle.

The applied mesh is summarized in Table 2. All meshes consists of prism and hexahedron elements only. The mesh is generated using an in-house developed software, which utilizes the repeating lattice structure of the assembly to generate a discretization with the same mesh characteristics for all pin cells. The generated mesh is written in a foam-extend-3.0 mesh format. The developed application also generates and prepares all boundary and initial conditions for all fields calculated.

7.2. Boundary and initial conditions

The axial boundary conditions are given as inlet and outlet boundary conditions, specified for all fields at the lower faces of the bottom reflector mesh and at the upper faces of the top reflector mesh. The conditions are specified as fixed value (Dirichlet) or zero gradient (homogeneous Neumann) conditions, with a special treatment for the case of the angular neutron flux. In the horizontal direction, reflective boundary conditions are used for all fields.

For the angular neutron flux ($\Psi_{m,g}$), a special implementation of the reflective boundary conditions is needed. Using the symmetry of the level symmetric weights and directions for S_N, a reflected direction can always be found for planes with normals parallel to the Cartesian coordinate axes. Thus for an incoming direction, the corresponding (i.e. reflected) outgoing direction is determined and the outgoing value of the flux is directly applied as an inlet condition for the incoming direction.

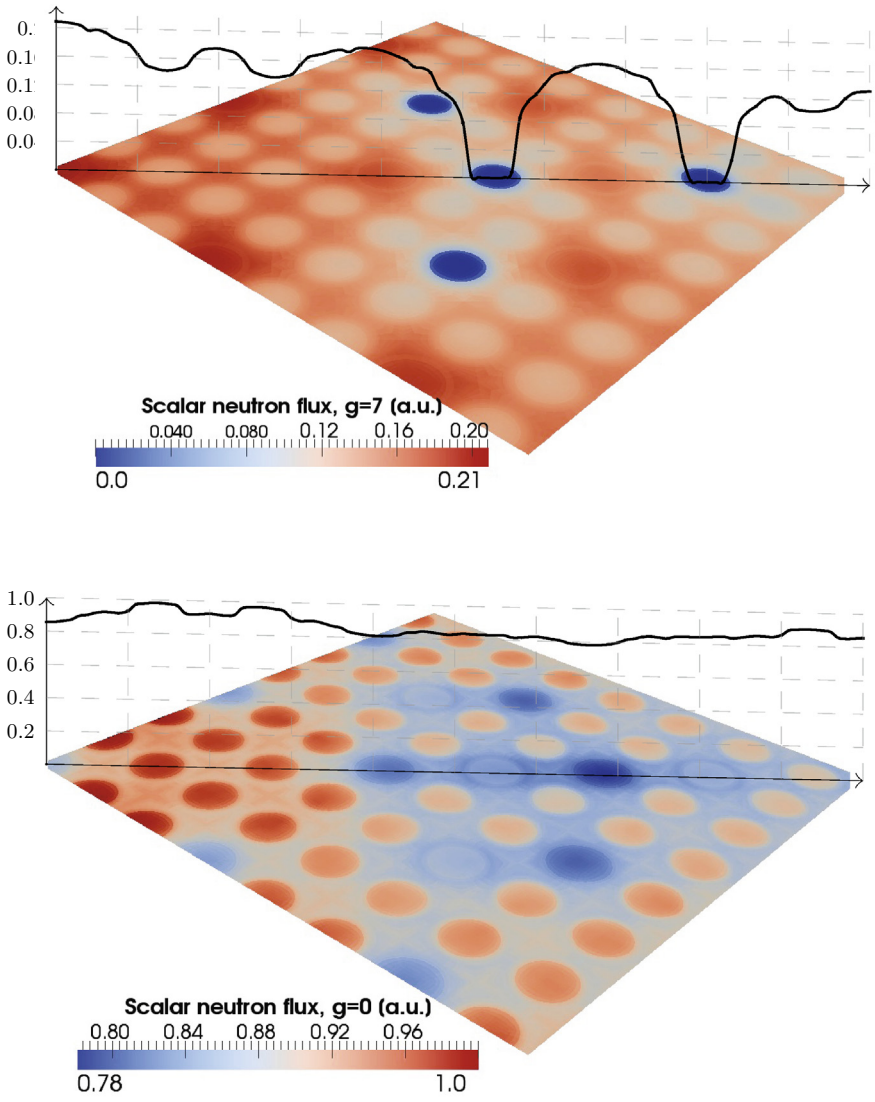

Fig. 12. Scalar flux at mid-elevation for the fast group ($g = 0$, bottom) and the thermal group ($g = 7$, top).

Constant initial conditions are used in all fields, but, as described in Section 5.3, during the first iterations not all fields are solved. Selected boundary and initial conditions are given in Table 3.

7.3. Domain decomposition

The system is decomposed into 64 parts, using 3 cutting planes in each of the Cartesian directions and in such a way that each cutting plane is along faces in all of the meshes, as described in Section 6.3. No automatic load balancing is applied. However, due to the computational burden of the discrete ordinates method, the neutronics mesh is the most important to balance, with the aim of an even computational effort on all CPUs.

The choice of decomposition is not unique and to optimize the performance not only an even distribution of cells should be the target. In our case with an axially coarser and horizontally finer mesh, also the cutting plane direction will have a major significance on the performance. Since only face values will need to be transferred, as few faces as possible should be transferred. Consequently, from a data transfer point of view, in our case it is better to cut in planes with normals $\hat{x}$ and $\hat{y}$.

7.4. Neutronic parameters

The settings for the neutron transport solver are given in Table 4. Eight discrete energy groups are used, based on the group structure from CASMO-4 (Studsvik Scandpower, 2009). We apply S_8, which corresponds to 80 discrete directions for each energy group, for a total of 640 neutron flux fields. A larger set of tests for the discrete ordinates, applying the same code was performed and reported elsewhere (Jareteg et al., 2014b). The total power is the power integrated over all the fuel pins, and is employed to normalize the scalar and angular neutron fluxes.

7.5. Results

The presented case was run on 64 Nehalem CPUs (Intel® Xeon® E5520, 2.27 GHz), divided on 8 computational nodes, with a total wall-clock running time of 48,000 s (≈14 h).

The moderator temperature distribution is shown in Fig. 11. As can be seen, both horizontal and axial heterogeneous distributions are achieved. The maximum axial temperature occurs at the top of the lattice, just below the top reflector, and the maximal horizontal temperature at the points where the distance between the pins is smallest.

Fig. 12 gives the scalar flux at mid-elevation for the fast and the thermal group. The flux profile along the symmetry line at mid-elevation is also given. The lowest energy group flux has strong minima in the pins which partially consist of burnable absorber, and maxima in the empty fuel rod channels. In contrast, the highest energy flux has minima in the water channels.

Considering both the slices and the line plots, some ray effect can be found. Such artifacts are typical for the discrete ordinates methods, and arise from the inability to reconstruct the angular flux with a set of few ordinates (Lewis and Miller, 1984). The easiest remedy against this is to increase the order of the method, which will however also increase the computational time.

The convergence of the coupled system, with residuals for each separate equation, is given in Fig. 13. The diagram displays the first eight outer iterations (each corresponding to a full loop in Fig. 7), with the convergence for the subiterations.

As explained in Section 5.3, during the first iteration only the diffusion neutronics is solved, whereas in the second iteration

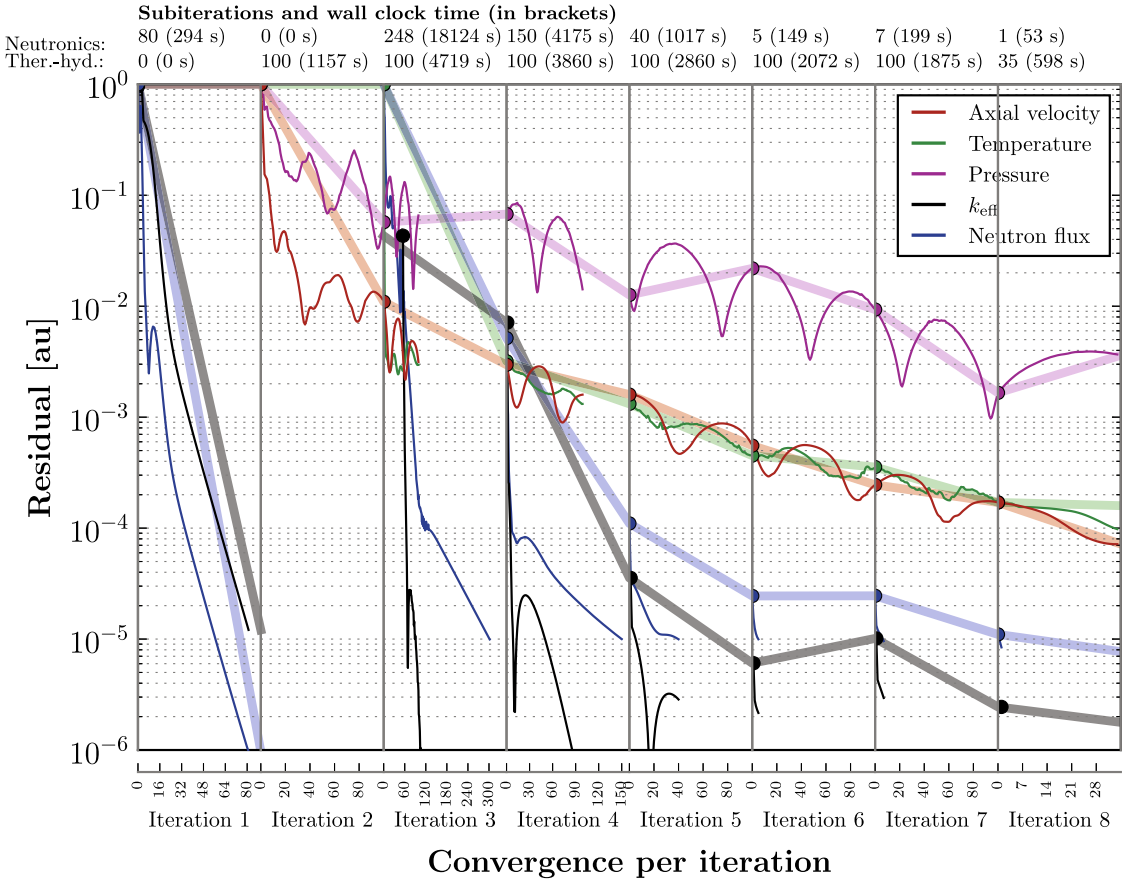

Fig. 13. Convergence results for the coupled system, with outer iteration convergence as opaque broader lines and the corresponding inner iterations as thinner lines.

the pressure and velocity are calculated. In the third iteration the S_N solver, as well as the temperature equation, are added and during the fourth iteration the first update of the cross-sections occurs. As seen from the figure, the interdependence is close to convergence after the seventh iteration. From iteration nine a single S_N sweep occurs for each outer iteration, and the total change in k_{eff} is only 10 pcm from iteration 8 to iteration 50. The thermal-hydraulics solver takes in total 50 iterations to converge. However, much less than 100 subiterations for most of the iterations are required. The slower convergence of the thermal-hydraulics solver is a property of the applied CFD-algorithm, and is not connected to the coupled solver.

From Fig. 13, it is shown that the largest effort is spent for the first neutronic iteration including S_N (Iteration 3). After this iteration the thermal-hydraulics requires the majority of the time. Summing over all iterations, the thermal-hydraulic and neutronic calculations take 45.1% and 54.1% of the total time, respectively. For such comparison it should also be noted that the thermal-hydraulic mesh is in the presented case finer than the neutronic mesh (see Table 2) and also that the computational time for the neutronics depends on the chosen number of energy groups and directions.

For the thermal-hydraulics the subiteration convergence is not monotonic. Instead, a periodic behavior was observed, where the axial velocity and the pressure residuals decrease and increase out of phase. Again, such a behavior is a property of the algorithms and matrix solvers applied and not any artifact of the coupling.

8. Summary and conclusion

We present a fine-mesh solver for the coupled neutronic and thermal-hydraulic problem that can be applied to PWR sub-pin level calculations. The work focuses on the methods and the implementation strategy of such a framework.

The use of HPC and fully parallelized solvers are both pointed out as key issues. The fine-mesh approach results in a large number of computational cells, relying on the use of computational clusters to solve the problem in a feasible time. The implementation of the decomposition necessary for the parallelization is described, along with a scheme minimizing the amount of data transfer by keeping all fields of physics in the same spatial region on the same CPU.

The methodology is implemented as a standalone application based on the open source C++ library foam-extend-3.0. It includes a neutronic solver based on the discrete ordinates method, and a thermal-hydraulic solver based on the mass, momentum and energy equations, complemented by a turbulence model. Both fields of physics are solved using the same simulation framework. This feature is essential to directly formulate the coupling on the fine-mesh level. In addition, a fully conservative mesh mapping scheme is included, and it aims at exchanging coupled data between different meshes using the finite volume methodology.

The fine-mesh solver is tested for the case of a quarter of a 15×15 fuel pin lattice, and a converged coupled solution is reached. Physically correct dependencies are obtained for the simulated variables and the results confirm that the present coupled scheme works. It also shows that the algorithms and implementations are efficient enough to produce converged results on a fine-mesh within 14 h using 64 CPUs. The convergence behavior of the coupled solver points out that there is much room for improvement of the separated models for neutronics and thermal-hydraulics. However, the convergence of the multiphysics problem demonstrates that the presented iterative scheme is working well to resolve the coupled dependencies.

Many interesting and challenging areas need to be investigated, including a future extension to two-phase flow simulations, so that BWR cases and departure from nucleate boiling in PWRs can also be considered. Such an extension requires not only to formulate and implement accurate two-fluid models, but it also poses a new type of coupled problem, including new conjugate heat transfer regimes, and an anticipated stronger feedback to the neutronics. Furthermore, other fields of physics could be fit in the same framework, including explicit thermal expansion, fluid–structure interaction and others.

In the presented simulation, reflective boundary conditions are used for all fields in the horizontal direction, disregarding the global dependence of all fields. Future work could tackle this approximation by either running a larger case using the presented framework (thus realized by investing more computational effort) or by coupling multiple scales of resolution, only adding a coarse layer to the simulation. Whereas the first approach is easier but more computationally expensive, the second method is more challenging but a it has the potential for a better use of the resources at hand.

Acknowledgments

The Swedish Center for Nuclear Technology (SKC) is acknowledged for financially supporting this PhD project. This work is carried out within the DREAM (Deterministic Reactor Modelling) framework at Chalmers University of Technology. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

References

Abu-Shumays, I.K., 2001. Angular quadrature for improved transport computations. Transport Theory Stat. Phys. 30 (2-3), 169–204, URL <http://dx.doi.org/10.1081/TT-100105367>.

Bakosi, J., Christon, M.A., Lowrie, R.B., Pritchett-Sheats, L.A., Nourgaliev, R.R., 2013. Large-eddy simulations of turbulent flow for grid-to-rod fretting in nuclear reactors. Nucl. Eng. Des. 262, 544–561, ISSN 0029-5493.

Calvin, C., Nowak, D., 2010. High Performance Computing in Nuclear Engineering. Springer Science+Business Media, Url: <http://dx.doi.org/10.1007/978-0-387-98149-9_12>.

Cardoni, J., Rizwan-uddin, 2011. Nuclear Reactor Multi-physics Simulations with Coupled MCNP5 and STAR-CCM+. M&C 2011, Rio de Janeiro, Brazil.

Ferziger, J., Peric, M., 2002. Computational Methods for Fluid Dynamics. Springer.

Gaston, D., Newman, C., Hansen, G., Lebrun-Grandié, D., 2009. Moose: a parallel computational framework for coupled systems of nonlinear equations. Nucl. Eng. Des. 239 (10), 1768–1778, URL <http://www.sciencedirect.com/science/article/pii/S0029549309002635>.

Hamilton, S., Clarno, K., Berrill, M., Evans, T., Davidson, G., Lefebvre, R., Sampath, R., Hansel, J., Ragusa, J., Josey, C., 2013. Multiphysics Simulations for LWR Analysis. M&C 2013, Sun Valley, Idaho.

Hébert, A., 2010. Multigroup Neutron Transport and Diffusion Computations. Springer-Verlag, URL <http://dx.doi.org/10.1007/978-0-387-98149-9_8>.

Ivanov, K., Avramova, M., 2007. Challenges in coupled thermal hydraulics and neutronics simulations for LWR safety analysis. Ann. Nucl. Energy 34 (6), 501–513, URL <http://dx.doi.org/10.1016/j.anucene.2007.02.016>.

Jareteg, K., Vinai, P., Demazière, C., 2013. Investigation of the Possibility to Use a Fine-mesh Solver for Resolving Coupled Neutronics and Thermal-hydraulics. M&C 2013, Sun Valley, Idaho.

Jareteg, K., Vinai, P., Demazire, C., 2014a. Fine-mesh deterministic modeling of PWR fuel assemblies: proof-of-principle of coupled neutronic/thermal hydraulic calculations. Ann. Nucl. Energy 68 (0), 247–256, URL <http://www.sciencedirect.com/science/article/pii/S0306454914000036>.

Jareteg, K., Vinai, P., Sasic, S., Demazière, C., 2014b. Influence of an S_N Solver in a Fine-mesh Neutronics/Thermal-hydraulics Framework. Submitted to PHYSOR 2014, September 28–October 3, Japan.

Jasak, H., 2009. General Grid Interface – Theoretical Basis and Implementation. NUMAP-FOAM Summer School, Zagreb, 2–15 September, 2009.

Kochunas, B., Stimpson, S., Collins, B., Downar, T., Brewster, R., Baglietto, E., Yan, J., 2012. Coupled Full Core Neutron Transport/CFD Simulations of Pressurized Water Reactors. PHYSOR 2012, Knoxville, Tennessee, USA, April 15–20.

Larsen, E.W., Morel, J.E., 2010. Advances in discrete-ordinates methodology. In: Nuclear Computational Science. Springer, pp. 1–84, URL <http://link.springer.com/chapter/10.1007/978-90-481-3411-3_1>.

Leppänen, J., 2012. Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code, VTT Technical Research Centre of Finland.

Lewis, E., Miller, W.J., 1984. Computational Methods of Neutron Transport. John Wiley & Sons, Inc., USA.
MPI Forum, 2009. MPI: A Message-Passing Interface Standard. <http://www.mpi-forum.org/>.
Newman, C., Hansen, G., Gaston, D., 2009. Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO_2 nuclear fuel rods. J. Nucl. Mater. 392 (1), 6–15.
Page, M., Beaudoin, M., Giroux, A.M., 2010. Steady-state capabilities for hydro turbines with OpenFOAM. IOP Conf. Ser.: Earth Environ. Sci. 12, 012076, URL <http://dx.doi.org/10.1088/1755-1315/12/1/012076>.
Panton, R., 2005. Incompressible Flow, third ed. Wiley.
Patankar, S., Spalding, D., 1972. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. Int. J. Heat Mass Transfer 15, 1787–1806.
Plimpton, S., Hendrickson, B., Burns, S., McLendon, W., Rauchwerger, L., 2005. Parallel Sn sweeps on unstructured grids: algorithms for prioritization, grid partitioning and cycle detection. Nucl. Sci. Eng. 150, 267–283.
Studsvik Scandpower, 2009. CASMO-4.
Sutherland, I.E., Hodgman, G.W., 1974. Reentrant polygon clipping. Commun. ACM 17 (1), 32–42, URL <http://doi.acm.org/10.1145/360767.360802>.
Wikki, 2014. Foam-extend-3.0. (last visited March 9, 2014). URL: <http://wikki.gridcore.se/foam-extend>.
Yan, J., Kochunas, B., Hursin, M., Downar, T., Karoutas, Z., Baglietto, E., 2011. Coupled Computational Fluid Dynamics and MOC Neutronic Simulations of Westinghouse PWR Fuel Assemblies with Grid Spacers. NURETH-14, Toronto, Ontario, Canada, September 25-30.
Zou, L., Peterson, J., Zhao, H., Zhang, H., Andrs, D., Martineau, R., 2013. Solving Implicit Multi-mesh Flow and Conjugate Heat Transfer Problems with RELAP-7. M&C 2013, Sun Valley, Idaho.