# A Region Monolithic Conjugate Heat Transfer Solver

Vuko Vukčević[a], Arne Eggers[a], Nevena Perović–Rott[a]

[a] *SimScale GmbH, Ridlerstraße 31B, 80339 München, Germany*

**Abstract**

The paper presents an implicit, region coupled conjugate heat transfer (CHT) solver based on arbitrary polyhedral finite volume method (FVM). The solver can handle multiple fluid and solid regions that are solved monolithically, leading to fast convergence and strong parallel scalability. ghost fluid method (GFM) is used to enforce the continuity of temperature and heat flux across the interfaces in an implicit manner. Verification and validation of the solver is presented for three cases with increasing complexity, the last case being an electronics cooling application compared to experimental measurements.

*Keywords:*

Conjugate Heat Transfer, Finite Volume Method, Ghost Fluid Method, Region-Monolithic Solution Algorithm, Verification and Validation

## 1. Introduction

Conjugate heat transfer (CHT) is an often encountered problem in different branches of engineering: from classical heat exchangers (HE) to growing electronics cooling (EC) applications due to the proliferation of cloud infrastructure. A common CHT problem often includes a single or multiple fluids and multiple solids, where the conduction of heat through the solids and convection and conduction through the fluids is of interest. Some of the CHT problems are tackled with empirical (0-D) models, which are easy to use but hard to generalize, especially when one considers the complexity and uniqueness of geometries involved.

Compared to empirical models based on analytical expressions, high–fidelity computational fluid dynamics (CFD) solutions provide extremely accurate results taking into account possible intricacies of the geometry and the complexity of the real–life operating conditions with few approximations. However, such high–fidelity solutions often come with the following demanding requirements:

*Email addresses:* vvukcevic@simscale.com (Vuko Vukčević), aeggers@simscale.com (Arne Eggers), nrott@simscale.com (Nevena Perović–Rott)

1. CAD model generation, where the model needs to be of sufficient quality for finite volume (FV) mesh generation,
2. FV mesh generation, which is often tedious to set-up when dealing with multiple CAD bodies and potentially complex simulation set-up;
3. Performing CFD simulation, which requires,
    - Significant know–how and amount of work to set up a successful simulation. Certain electronic assemblies have hundreds of components with thousands of interfaces, making it difficult and error–prone to set up manually;
    - High computational resources needed to perform such analyses on complex geometries,
    - An advanced solution algorithm that takes into account the strong coupling between multiple regions in contact in a CHT problem.

Our goal at SimScale is to alleviate such requirements and make high–fidelity tools widely available, ultimately leading to better designs. SimScale offers automated meshing that takes into account physics set up, highly automated simulation set up and on–demand computational resources in the cloud. Although such an approach solves many of the problems listed above, a robust solution algorithm is crucial, which is the focus of this work.

Before discussing different solution algorithms, we shall briefly outline a common CHT problem. A common CHT problem usually has a number of fluid and solid regions, where the heat transfer between the solids and the surrounding fluids is of interest. The dominant physical phenomena in solid regions is the heat transfer, where the thermal expansion is assumed to be negligible. The fluid flow is assumed to be buoyant and is governed by Navier–Stokes equations, including heat transfer due to convection and conduction. Since the heat transfer is solved in all regions, regardless of whether they are fluids or solids, these regions are strongly–coupled, *i.e.*, the temperature field in a given solid will have a direct influence on the temperature, pressure and velocity fields in the adjacent fluid, and vice–versa.

There are two approaches to such a multi–region coupled problem:

1. Region–partitioned approach, where the regions are solved sequentially and independently of each other. This implies that while we are solving heat transfer for the $n$-th region, the temperature field in all other regions is kept fixed with values from the previous time step. One such implementation is found in OpenFOAM's ® [1] CHT solver: `chtMultiRegionSimpleFoam`. The partitioned algorithm is described in Alg. 1 and it leads to weak coupling between regions that is generally slow to converge.

2. Region–monolithic approach, where the heat transfer in all regions (fluids and solids alike) is solved simultaneously. Such an approach is numerically more complex as it involves implicit treatment of boundary conditions at the interfaces between regions, thus providing significantly faster convergence. The monolithic algorithm is described in Alg. 2.

The key benefit of the region–monolithic approach is that the heat transfer equations are implicitly coupled across all regions, leading to significantly faster convergence. Another benefit of the monolithic approach that is not as obvious is the improved parallel scaling. Finite volume CFD codes are usually parallelised using distributed memory, where each processor gets its own part of the solution domain to solve, communicating to its neighbours via message passing interface (MPI). OpenFOAM's ® CHT solver is an example of partitioned approach where each region is decomposed into a predefined number of domains. This inevitably leads to load imbalance because a densely discretised region (*e.g.*, air around different electronic components) may end up having hundreds of thousands of cells per processor, while a sparsely discretised region (*e.g.*, a small chip) may end up having only a few cells per processor.

The solution proposed in this work is based on a region–monolithic approach due to the drawbacks of the partitioned approach. The algorithm is implemented within second–order accurate, arbitrary polyhedral FV framework [2, 3]. Boundary conditions at the interface between different regions are handled with the ghost fluid method (GFM) [4, 5, 6, 7, 8], allowing implicit coupling between the regions. As discussed above, such an approach also yields decomposition–agnostic CFD software with parallel scaling comparable to standard single–region CFD analyses.

**Algorithm 1** Region–partitioned CHT algorithm.

```
while simulation not converged do
  for each fluid region do
    solve velocity and pressure equation
    solve heat transfer equation
  end for
  for each solid region do
    solve heat transfer equation
  end for
end while
```

**Algorithm 2** Region–monolithic CHT algorithm.

```
while simulation not converged do
  formulate and solve a single velocity and pressure equation for all regions
  formulate and solve a single heat transfer equation for all regions
end while
```

The paper is organized as follows. Sec. 2 presents the mathematical model of a CHT problem with emphasis on interface boundary conditions and the possibility of combining the two sets of governing equations for solids and fluids into a single equation set. Sec. 3 presents the numerical framework with details on GFM treatment of boundary conditions for all variables. Sec. 4 presents verification and validation of the proposed model, followed by its application on a realistic engineering problem. The paper is summarised in Sec. 5.

## 2. Mathematical model

This section presents the mathematical model for a number of steady, incompressible, and viscous fluids surrounded by multiple solids, where the heat transfer between solids and fluids is of interest. The governing equations for fluid and solid regions are presented separately, followed by the boundary conditions at the interfaces between different regions. The governing equations are finally formulated in a way that they yield a single set of equations valid for all regions.

### *2.1. Governing equations for a fluid region*

Fully nonlinear, incompressible, turbulent and steady–state flow is assumed, which can be described by the continuity and the Navier–Stokes equations, respectively:

$$\nabla\cdot(\rho\mathbf{u}) = 0\,, \tag{1}$$

$$\nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \nabla\cdot(\rho\nu_e\nabla\mathbf{u}) = -\nabla p_d - \mathbf{g}\cdot\mathbf{x}\nabla\rho^* + \mathbf{S}_\mathbf{u}\,, \tag{2}$$

where $\mathbf{u}$ is the velocity field, $\nu_e$ is the effective fluid viscosity defined as the sum of the laminar viscosity and turbulent viscosity using eddy-viscosity turbulence models[1], $\mathbf{g}$ is the gravitational acceleration, $\mathbf{x}$ is the position vector, $\mathbf{S}_\mathbf{u}$ are the collective source and sink terms and $p_d$ is the dynamic pressure defined as:

$$p_d = p - \rho^*\mathbf{g}\cdot\mathbf{x}\,, \tag{3}$$

where $p$ is the pressure field.

Note that $\rho$ and $\rho^*$ are distinguished in order to be able to handle both closures by the Boussinesq approximation and the Equation of State (EOS). In the case of the Boussinesq approximation:

$$\begin{aligned}\rho &= 1\,,\\ \rho^* &= 1 - \beta(T - T_{ref})\,,\end{aligned} \tag{4}$$

[1] In this work, we use the Menter's $k$–$\omega$ $SST$ model with standard wall functions [9]

where $\beta$ is the thermal expansion coefficient at a reference temperature $T_{ref}$. The Boussinesq approximation is strictly valid only for small temperature differences, where the reader is referred to [10] for additional details.

In case of a closure with an EOS, $\rho$ and $\rho^*$ become the same:

$$\rho = \rho^* = \frac{p}{RT}\,, \tag{5}$$

where a perfect gas equation of state is assumed and $R$ denotes the gas constant.

Energy equation is written with the temperature as the primary variable and reads:

$$\nabla\cdot(\rho c_p \mathbf{u} T) - \nabla\cdot(\kappa\nabla T) = S_T\,, \tag{6}$$

where $c_p$ is specific heat capacity of the fluid at constant pressure, $S_T$ are power sources and sinks, while $\kappa$ is thermal conductivity defined as:

$$\kappa = \frac{\rho c_p \nu}{P_r} + \frac{\rho c_p \nu_t}{P_{r,t}}\,, \tag{7}$$

where $\nu$ and $P_r$ are the laminar viscosity and the laminar Prandtl number, respectively, while $\nu_t$ and $P_{r,t}$ are the turbulent viscosity and the turbulent Prandtl number.

Eqns. (1)– (6) constitute a closed set of equations governing the fluid flow. Note that in the following text it is assumed that $\rho = \rho^*$ for simplicity, although the derivation of the final mathematical model is not affected by such choice.

### *2.2. Governing equations for a solid region*

Heat conduction through an isotropic solid material is governed by:

$$\nabla\cdot(\kappa\nabla T) = S_T\,, \tag{8}$$

Given appropriate boundary conditions, the solution of Eqn. (8) provides the temperature field in the solid. Velocity, pressure and density of the solid can be also thought of as uniform fields defined in the solid since thermal expansion

and stresses are not of interest in this work:

$$\begin{aligned} \rho &= \rho_s \,, \\ p_d &= p_a \,, \\ \mathbf{u} &= \mathbf{u}_s = \mathbf{0} \,, \end{aligned} \tag{9}$$

where $\rho_s$ is the constant density of the solid and $p_a$ is a constant ambient pressure without the hydrostatic component.

### *2.3. Interface boundary conditions*

A solid–fluid interface, denoted by $\Gamma$, is a surface dividing a fluid region $\Omega_f$ and a solid region $\Omega_s$ [2]. Velocity at the interface is equal to zero because the interface is considered to be a no–slip wall:

$$\mathbf{u}_\Gamma = \mathbf{0} \,, \tag{10}$$

In order to derive a boundary condition for dynamic pressure at the interface, it is assumed that the gradient of density in the normal direction to the interface is small[3]. Combining the velocity boundary condition at the interface, Eqn. (10) and the momentum conservation equation, Eqn. (2), it follows that the dynamic pressure gradient normal to the interface is equal to zero:

$$\nabla_\Gamma(p_d) = 0 \,, \tag{11}$$

where $\nabla_\Gamma$ denotes interface normal gradient evaluated at the interface $\Gamma$.

Thermal conductivity of a solid is considered constant in this work, while the thermal conductivity of a fluid is defined by Eqn. (7). Therefore, thermal conductivity has a discontinuity at the interface, which can be written as:

$$[\kappa] = \kappa_s - \kappa_f \,, \tag{12}$$

where $[\cdot]$ denotes a jump–condition operator following [11, 7], defined as a difference between the conductivity infinitesimally close to the interface at the solid side ($\kappa_s$) and the conductivity infinitesimally close to the interface at the fluid side ($\kappa_f$).

Taking into account the discontinuity of thermal conductivity given by Eqn. (12), there are two additional interface conditions for the temperature. The first interface condition states that the temperature field is continuous across the

[2] Note that the analysis also holds for a solid–solid interface

[3] This assumption leads to first–order accuracy near the interface, which shall be addressed in future work.

interface, *i.e.* values infinitesimally close to the interface on both sides are the same:

$$[T] = 0\,. \tag{13}$$

The second interface condition states that the heat flux at the interface has to be the same infinitesimally close on both sides:

$$[\kappa \nabla T] = \mathbf{0}\,. \tag{14}$$

Eqns. (10)– (14) represent all the necessary interface boundary conditions, whose numerical treatment shall be considered in depth in Sec. 3. It is also important to note that the equivalent equations are valid if one considers two solid regions in contact.

### *2.4. Combined governing equations*

If one takes into account interface conditions laid out in the previous section, Sec. 2.3, then it is possible to define all fields: $\mathbf{u}$, $p_d$, $T$ and $\rho$ within the whole volume of interest:

$$\Omega = \Omega_{f1} \cup \Omega_{f2} \cup \ldots \cup \Omega_{fn} \cup \Omega_{s1} \cup \Omega_{s2} \cup \ldots \cup \Omega_{sm}\,, \tag{15}$$

where $\Omega_{fi}$ is the $i$–th fluid region and $\Omega_{sj}$ is the $j$–th solid region. In order to combine different governing equations for the fluid and the solid regions, a fluid indicator field is introduced:

$$\alpha = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_f\,, \\ 0 & \text{if } \mathbf{x} \in \Omega_s\,. \end{cases} \tag{16}$$

$\alpha$ is therefore a piece–wise constant field with the discontinuity at the interface $\Gamma$, which can be used to define the combined governing equations:

- Combined continuity equation:

$$\alpha\Big(\nabla\cdot(\rho\mathbf{u})\Big) + (1-\alpha)\Big(\beta_p(p_d - p_a)\Big) = 0\,, \tag{17}$$

  where the dummy pressure equation in the solid given by Eqn. (9) has been multiplied by $\beta_p = 1\,\mathrm{m}^{-2}\,\mathrm{s}$. This is needed in order to have consistent dimensions of the two terms, having in mind that the continuity equation shall be used to derive the pressure equation in Sec. 3.

- Combined momentum conservation equation:

$$\alpha\Big(\nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \nabla\cdot(\rho\nu_e\nabla\mathbf{u}) + \nabla p_d - \mathbf{g}\cdot\mathbf{x}\nabla\rho + \mathbf{S}_\mathbf{u}\Big) + (1-\alpha)\Big(\beta_\mathbf{u}(\mathbf{u}-\mathbf{u}_s)\Big) = \mathbf{0}\,, \tag{18}$$

  where the dummy momentum equation in the solid given by Eqn. (10) has been multiplied by $\beta_\mathbf{u} = 1\,\mathrm{kg}^{-1}\,\mathrm{m}^3\,\mathrm{s}^{-1}$.

- Combined heat transfer equation:

$$\alpha\Big(\nabla\cdot(\rho c_p\mathbf{u}T)\Big) - \nabla\cdot(\kappa\nabla T) = S_T\,, \tag{19}$$

  where the convection term is only present in the fluid region, while the diffusion (conduction) term is present in both fluid and solid regions, albeit with discontinuity in thermal conductivity $\kappa$ given by Eqn. (12). Embedding temperature interface conditions given by Eqn. (13) and Eqn. (14) shall be discussed in detail in Sec. 3.

- Combined EOS:

$$\rho = \alpha\Big(\frac{p}{RT}\Big) + (1-\alpha)\rho_s\,, \tag{20}$$

  where a perfect gas EOS has been assumed for the fluid. Note that the derivation presented in this work is valid for an arbitrary EOS (assuming no phase change) or even the Boussinesq approximation.

Eqns. (17)–(20) define a complete set of governing equations valid for the whole computational domain $\Omega$ comprising fluid and solid regions. However, it is important to note that at the interface between each pair of regions $\Gamma$, the interface conditions outlined in Sec. 2.4 must be embedded into the discretisation, which is the topic of the next section.

## 3. Numerical framework

Combined governing equations given by Eqn. (17), (18) and (19) are discretised in space using a second–order accurate, collocated FV method for arbitrary polyhedral grids [2]. A convex polyhedral control volume (CV, or simply, a cell) has a number of face–neighbours, where each face of the CV is represented by a surface area vector $\mathbf{S}_f$ and distance vector $\mathbf{d}_f$, both positive towards the neighbouring cells.

Discretised governing equations are presented in Sec. 3.1, followed by the treatment of interface conditions in Sec. 3.1.1, Sec. 3.1.2 and Sec. 3.1.3. A solution algorithm for solving the discretised equations with embedded boundary conditions is presented in Sec. 3.2.

### 3.1. Discretised governing equations

The discretised equations using the general FV method with arbitrary discretisation schemes shall be presented here without detailed analysis. Terms enclosed in curly braces $\{\cdot\}$ are implicitly discretised, resulting in a linear system of equations, while other terms represent explicit contributions that end up on the r.h.s. of equation. For details on standard, compact–stencil FV discretisation, reader is referred to [12, 13, 2, 14].

The discretised momentum equation reads:

$$\left\{\alpha\nabla\cdot(\rho\mathbf{u}\mathbf{u})\right\} - \left\{\alpha\nabla\cdot\left(\rho\nu_e\nabla\mathbf{u}\right)\right\} + \alpha\nabla p_d - \alpha\mathbf{g}\cdot\mathbf{x}\nabla\rho + \left\{\mathbf{S}_\mathbf{u}\right\} + (1-\alpha)\left\{\beta_\mathbf{u}(\mathbf{u}-\mathbf{u}_s)\right\} = \mathbf{0}\,, \tag{21}$$

where the convection and diffusion terms are implicitly discretised, along with the sink term and the last term representing the solid region equation. Note that only the convection and diffusion terms are affected by boundary conditions due to second–order FV discretisation.

The continuity equation is used to formulate the pressure equation:

$$\left\{\alpha\nabla\cdot\left(\frac{1}{a_\mathbf{u}}\nabla p_d\right)\right\} - \alpha\nabla\cdot\left(\frac{\mathbf{H}(\mathbf{u})}{a_\mathbf{u}}\right) + \alpha\nabla\cdot\left(\frac{\mathbf{g}\cdot\mathbf{x}}{a_\mathbf{u}}\nabla\rho\right) + \left\{(1-\alpha)\left(\beta_p(p_d-p_a)\right)\right\} = 0\,, \tag{22}$$

where $a_\mathbf{u}$ is the diagonal of the discretised momentum equation and $H_\mathbf{u}$ the operator collecting off–diagonal contributions and source/sink terms except the pressure and density gradient. For details on pressure–velocity coupling, reader is referred to Jasak [2].

The combined heat transfer equation in its discrete form reads:

$$\left\{\alpha\left(\nabla\cdot(\rho c_p\mathbf{u}T)\right)\right\} - \left\{\nabla\cdot(\kappa\nabla T)\right\} = \{S_T\}\,, \tag{23}$$

where all terms are implicitly discretised. Note that both convection and diffusion terms rely on boundary conditions, while source and sink terms are local. In the following analysis of interface boundary conditions, indicator field $\alpha$ shall be omitted for simplicity since it does not affect the end result and it can be deduced from the context whether certain terms are valid only in the fluid region or in both fluid and solid regions.

#### 3.1.1. Velocity interface condition

Following Sec. 2.3, the velocity at the interface is assumed to be zero in this work, representing a stationary, no–slip wall. The boundary condition given in Eqn. (10) has to be taken into account when discretising the convection and diffusion terms in the momentum equation, which exist only in the fluid region. The second–order FV discretisation

using the Gauss theorem yields the following face contribution for the convection term [2]:

$$F_{\Gamma f}\mathbf{u}_{\Gamma f}\,, \tag{24}$$

where $F$ is the flux through the face, while the $_{\Gamma f}$ subscript denotes a mesh face at the solid–fluid interface. Note that for stationary walls, both $F_{\Gamma f}$ and $\mathbf{u}_{\Gamma f}$ are zero, resulting in no contribution to the diagonal or the r.h.s. of the resulting linear system.

Similarly, the discretisation of the diffusion term yields the following face contribution:

$$\rho_{\Gamma f}\nu_{e,\Gamma f}|\mathbf{S}_{\Gamma f}|\frac{\mathbf{u}_{\Gamma f}-\mathbf{u}_P}{|\Delta_{\Gamma f}|}\,, \tag{25}$$

where $\mathbf{S}_{\Gamma f}$ is the surface area vector of the mesh face, $\mathbf{u}_P$ is the velocity field in the cell centre and $|\Delta_{\Gamma f}|$ is the distance vector from the cell centre to the interface face centre. For stationary wall, since the $\mathbf{u}_{\Gamma f}=\mathbf{0}$, the r.h.s. contribution to the linear system is zero, while the diagonal contribution is $\rho_{\Gamma f}\nu_{e,\Gamma f}|\mathbf{S}_{\Gamma f}|/|\Delta_{\Gamma f}|$.

### *3.1.2. Pressure interface condition*

The pressure equation Eqn. (22) has a diffusion term in the fluid region which has to take into account the pressure interface condition given by Eqn. (11). The discrete interface face contribution reads:

$$\frac{\mathbf{S}_{\Gamma f}}{a_{\mathbf{u},\Gamma f}}\cdot\nabla_{\Gamma f}p_d \tag{26}$$

where $\nabla_{\Gamma f}p_d=0$ according to Eqn. (11). Therefore, the pressure interface condition does not yield neither a diagonal contribution nor the contribution to the r.h.s. of the resulting linear system.

### *3.1.3. Temperature interface condition*

The velocity and pressure interface conditions consider only the fluid domain and are therefore simpler. In contrast, the temperature interface condition needs to take into account the discontinuity of conductivity at the interface, Eqn. (12) and the resulting temperature jump conditions given by Eqn. (13) and Eqn. (14). However, this complexity considers only the diffusion (conduction) term of the temperature equation which is non–zero for both the fluid and solid regions. The convection term that is non–zero only in the fluid region yields the following interface face contribution in its discrete form:

$$F_{\Gamma f}c_pT_{\Gamma f}\,, \tag{27}$$

where $T_{\Gamma f}$ is the temperature at the interface face. Note that since $F_{\Gamma f} = 0$, as implied by the velocity interface condition discussed in Eqn. (24), the convective term at the interface face is zero.

In order to correctly take into account the discretisation of the diffusion term, the GFM method is discussed first. The GFM method [4, 5, 6, 7, 8] is useful for handling field discontinuities at arbitrary interface. In its core, the GFM method is similar to a family of *embedded* method [15, 16, 17, 18, 19]. The only difference between the GFM and the embedded methods seems to be that the embedded methods can be easily generalized to higher–order, while the GFM are inherently either first–order or second–order. Since the framework presented here relies on the second–order FV method, the GFM method as presented in detail in [8] is followed to derive the discretisation that takes into account discontinuity of conductivity and temperature gradient at the interface.

Fig. 1 presents the GFM treatment for a field $\phi$ whose value and gradient may be discontinuous at the interface. The figure represents a 1–D case for simplicity, without loss of generality for 3–D cases. Cells $P$ and $N$ share an interface face $\Gamma_f$, where the field $\phi$ exhibits a discontinuity (*i.e.*, a jump) in both its value and its gradient. Two additional unknowns are therefore introduced: $\phi_\Gamma^P$ and $\phi_\Gamma^N$, which represent values infinitesimally close to the interface in cell $P$ and cell $N$,respectively. If one knows the interface conditions for $\phi$ and its gradient, the two additional unknowns are supplemented by two additional equations. The resulting system of equations can then be solved in order to express interface values ($\phi_\Gamma^P$ and $\phi_\Gamma^N$) in terms of neighbouring cell centre values: $\phi_P$ and $\phi_N$. Finally, when performing the discretisation for cell $P$, the extrapolated value $\phi_N^P$ is used for the neighbouring cell and vice–versa. Such procedure results in second-order accurate, implicit *embedding* of interface conditions into the discretisation scheme. Such procedure shall be shown here for the temperature field.

Following Eqn. (13), the temperature field at the interface has the same value infinitesimally close to the interface on both sides. Rewritten in terms of general GFM discretisation presented in Fig. 1, the equation reads:

$$T_\Gamma^P = T_\Gamma^N \,. \tag{28}$$

The temperature gradient boundary condition given by Eqn. (14) provides three additional boundary conditions since it is a vector equation. In order to simplify the analysis and remove redundant components tangential to the interface, the Eqn. (14) is projected onto a unit vector $\hat{\mathbf{d}}_f$:

$$[\kappa \nabla T] \cdot \hat{\mathbf{d}}_f = 0 \,, \tag{29}$$

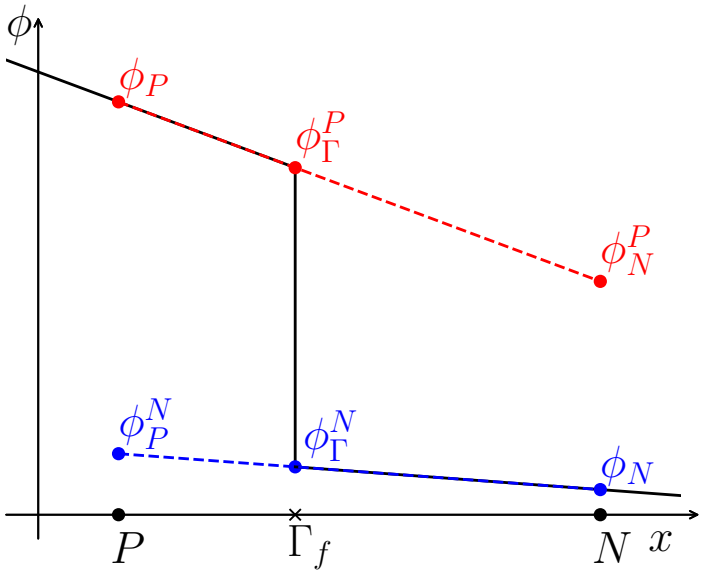



Figure 1: GFM treatment of a discontinuous field $\phi$.

where $\hat{\mathbf{d}}_f = \mathbf{d}_f/|\mathbf{d}_f|$ is the unit vector from cell centre $P$ to cell centre $N$. Eqn. (29) can be discretised as:

$$\kappa_\Gamma^N \frac{T_N - T_\Gamma^N}{|\mathbf{d}_{\Gamma N}|} - \kappa_\Gamma^P \frac{T_\Gamma^P - T_P}{|\mathbf{d}_{P\Gamma}|} = 0\,, \tag{30}$$

where $\mathbf{d}_{\Gamma N}$ is the distance vector from the interface to cell centre $N$ and $\mathbf{d}_{P\Gamma}$ is the distance vector from the cell centre $P$ to the interface. The combined subscript and superscripts $^N_\Gamma$ and $^P_\Gamma$ represent values infinitesimally close to the interface on the neighbour side and one the owner side, respectively. Eqn. (30) and Eqn. (28) may be considered as two additional equations with two additional unknowns: $T_\Gamma^P$ and $T_\Gamma^N$. In order to embed the interface conditions into collocated (cell–centred) FV discretisation, the goal is to obtain values $T_\Gamma^P$ and $T_\Gamma^N$ expressed in terms of cell centred valued $T_P$ and $T_N$. The inversion of a similar system is straightforward and presented in detail in [8] and will not be repeated here. Expressed in terms of cell centre values, the temperature at the interface can be found as:

$$T_\Gamma = T_\Gamma^P = T_\Gamma^N = \lambda \frac{\kappa_{\Gamma N}}{\overline{\kappa}_f} T_N + (1-\lambda)\,\frac{\kappa_{\Gamma P}}{\overline{\kappa}_f} T_P\,, \tag{31}$$

where $\lambda = |\mathbf{d}_{P\Gamma}|/|\mathbf{d}_f|$ is the dimensionless distance from the cell centre $P$ to the interface $\Gamma$ and:

$$\overline{\kappa}_f = \lambda \kappa_{\Gamma N} + (1-\lambda)\,\kappa_{\Gamma P}\,, \tag{32}$$

is the GFM–interpolated conductivity.

In addition to the interface temperature $T_\Gamma$, the FV discretisation schemes occasionally need neighbouring values. When performing discretisation for cell $P$, temperature at the cell $N$ needs to be extrapolated from the interface temperature in order to take into account the discretised interface conditions. Referring to Fig. 1, when performing discretisation for cell $P$, the value $T_N^P$ should be considered instead of $T_N$. The neighbouring value $T_N^P$ can be calculated with second–order accurate extrapolation from the interface:

$$T_N^P = T_\Gamma + \frac{1-\lambda}{\lambda}\left(T_\Gamma - T_P\right) , \tag{33}$$

where the second term represents the extrapolation of the gradient $(T_\Gamma - T_P)/\lambda$ from the interface $\Gamma$ to the neighbouring cell centre at a dimensionless distance of $1-\lambda$. Inserting Eqn. (31) into Eqn. (33) yields the extrapolated value:

$$T_N^P = \frac{\kappa_{\Gamma N}}{\overline{\kappa}_f} T_N + \left(1 - \frac{\kappa_{\Gamma N}}{\overline{\kappa}_f}\right) T_P \, . \tag{34}$$

It's important to stress that the extrapolated neighbour value $T_N^P$ is expressed with values at two cell centres sharing an interface face: $T_P$ and $T_N$, therefore respecting the compact–stencil framework of the collocated FV method. A similar approach can be followed to derive the extrapolated value from cell $N$ to cell $P$: $T_P^N$.

With Eqn. (34) in place, the attention is turned to the discretisation of the diffusion term (see the second term in Eqn. (23)) for an interface face of a cell $P$:

$$\kappa_{\Gamma P} |\mathbf{S}_{\Gamma f}| \frac{T_N^P - T_P}{|\Delta_{\Gamma f}|} + \kappa_{\Gamma P} \mathbf{k}_{\Gamma f} \cdot (\nabla T)_{\Gamma f} \, , \tag{35}$$

where $\mathbf{k}_{\Gamma f}$ is the non–orthogonal correction vector in over–relaxed form (see Jasak [2]) and $(\nabla T)_{\Gamma f}$ is the face–interpolated temperature gradient. The attention is first put on the orthogonal component of the gradient represented with the first term in Eqn. (35), where $T_N^P$ is used instead of $T_N$ in order to take into account the interface conditions [4] Inserting Eqn. (34) in Eqn. (35), yields:

$$\frac{|\mathbf{S}_{\Gamma f}|}{|\Delta_{\Gamma f}|} \frac{\kappa_{\Gamma P} \kappa_{\Gamma N}}{\overline{\kappa}_f} \left(T_N - T_P\right) . \tag{36}$$

Eqn. (36) leads to fully implicit discretisation of the orthogonal part of the diffusion term, where the off–diagonal matrix contribution is the same as the negative diagonal contribution as in the standard FV method. As opposed to the orthogonal part, the non–orthogonal contribution (the second term in Eqn. (35) is treated explicitly and it requires the computation of the temperature gradient multiplied with thermal conductivity $\kappa$. In this work, a least–squares

[4] Note that for all other non–interface faces, a standard discretisation involving $T_N$ is used.

discretisation is used for gradients, following Jasak and Weller [20]. The least–squares discretisation expresses the gradient in terms of the least–squares vectors:

$$\kappa_P \nabla T_P = \kappa_P \sum_f \mathbf{l}_f \left(T_N - T_P\right) , \tag{37}$$

where $\sum_f$ represents sum over all the faces of cell $P$ and $\mathbf{l}_f$ is the least–squares vector as defined in [20]. For non–interface faces, standard treatment is sufficient whereas for a contribution due to an interface face, $T_N$ is replaced by $T_N^P$:

$$\kappa_P \mathbf{l}_f \left(T_N^P - T_P\right) = \mathbf{l}_f \frac{\kappa_P \kappa_N}{\overline{\kappa}_f} \left(T_N - T_P\right) \tag{38}$$

where Eqn. (34) is used to express $T_N^P$ in terms of $T_N$ and $T_P$. Note that the governing equations always consider the gradient of temperature multiplied with thermal conductivity (see Eqn. (23)), which is continuous across the interface as given in Eqn. (14). Having this in mind, the temperature gradient can be readily interpolated to the faces for the discretisation of the non–orthogonal part given in Eqn. (35).

### *3.2. Solution algorithm*

Sec. 3.1.1, Sec. 3.1.2 and Sec. 3.1.3 discussed discretisation details for interface boundary conditions, with the special emphasis on the temperature interface conditions that are embedded within the discretisation for solid–fluid interface faces. Standard FV discretisation is used for ordinary faces, *i.e.*, faces whose both cells belong to either a solid region or a fluid region. Unless otherwise stated for certain test cases, Gauss discretisation using a second–order, upwind–biased scheme is used for the convection terms, while the diffusion terms are discretised using the standard compact–stencil FV discretisation with the over–relaxed approach for explicit, non–orthogonal correction [21].

A segregated solution algorithm is used to solve the discretised equations, Eqn. (21), Eqn. (22) and Eqn. (23). The pressure–velocity coupling is resolved using the SIMPLE algorithm [22], with Rhie-Chow pressure filtering [23]. Symmetric pressure equation is solved with the preconditioned Conjugate Gradient (CG) method, while the momentum equation is solved with the stabilised, preconditioned Bi–Conjugate Gradient (BiCG) method. The temperature equation is also solved with the BiCG method with Crout's version of the ILU algorithm with arbitrary level of fill–in [24] as a preconditioner. Since the conductivities of the fluid and the solid usually differ by four orders of magnitude, the multi–region temperature equation is ill–conditioned. Therefore, a fill–in level of one that considers neighbours of neighbours is used. [5]

[5] Due to the large difference of conductivities at the solid–fluid interfaces, standard multi–grid methods did not prove to be superior to the CG family of methods. The extension of multi–grid methods to such matrices shall be considered for future work.

Solution algorithm for the presented region–monolithic framework is presented in Alg. 3. The simulation starts by initializing the fluid indicator field $\alpha$ defined by Eqn. (16), followed by the calculation of GFM interpolated conductivity field $\overline{\kappa}$ for interface faces, Eqn. (32). The time loop (*i.e.*, nonlinear iteration loop) is entered, and each iteration starts with the solution of the momentum equation, Eqn. (21) with the pressure and the density gradients from the previous iteration. The solution of momentum equation yields cell–centred velocity field, which is then interpolated to the faces using the Rhie–Chow interpolation. The pressure equation is then assembled and solved, Eqn. (22), yielding conservative fluxes. With the updated pressure available, density is calculated from the EOS for non–Boussinesq simulations, Eqn. (5). The updated density is also used to calculate the fluid conductivity given by Eqn. (7). With the density and conductivity updated, the heat transfer equation is assembled and solved, taking into account the temperature interface conditions presented in Sec. 3.1.3. The new temperature field is then used to update the fluid density from the EOS, Eqn. (4) or Eqn. (5), depending on the model used. Optionally, the turbulence modelling equations are solved at the end of a nonlinear iteration and the effective viscosity is updated. The procedure is repeated until the desired level of convergence is reached. [6]

**Algorithm 3** Detailed solution algorithm for the region–monolithic CHT framework.

initialize fluid indicator field $\alpha$, Eqn. (16),
calculate $\overline{\kappa}$ for interface faces, Eqn. (32)
**while** simulation not converged **do**
  solve the momentum equation, Eqn. (21)
  solve the pressure equation, Eqn. (22)
  calculate fluid density from the Equation of State, Eqn. (5) (only for non–Boussinesq simulations),
  calculate fluid conductivity, Eqn. (7),
  solve the heat transfer equation, Eqn. (23)
  calculate fluid density from the Equation of State, Eqn. (4) or Eqn. (5) (depending on the model)
  solve turbulence equations and update effective viscosity
**end while**

## 4. Verification and validation

This section presents the verification and validation of the presented numerical framework. The cases present here are shown in increasing complexity: from simple cases that verify and validate the numerical framework to complex industrial cases that present its capabilities.

[6] For applications of interest in this work, convergence is usually best judged by the average temperature on a given solid–fluid interfaces. Another alternative would be to determine the convergence from residuals of momentum, pressure and temperature equations, although this is not possible for realistic cases involving complex geometries, usually with sharp edges, because the steady state solution might not exist, resulting in oscillating residuals.

### 4.1. 1–D heat conduction through two solids

A simple 1–D heat conduction case is considered first to validate the GFM framework of handling interface discontinuities between two regions with different conductivities. A one metre long rod is considered, consisting of two solids with different conductivities. The left part of the rod has the conductivity of $\kappa_L = 100$ W/(mK), while the right part of the rod has the conductivity of $\kappa_R = 1000$ W/(mK) (see Fig. 2). The temperature at the left boundary is set to $T_L = 300$ K, while the temperature at the right boundary is set to $T_R = 400$ K.

The analytical solution for Eqn. (23) yields a piece–wise linear temperature profile in the rod, with the discontinuity of the gradient at the interface. It can be shown that the analytical solution is expressed as:

$$\begin{aligned} T_L(x) &= \frac{1}{\kappa_L}\left(\frac{\kappa_R(T_R - T_L)}{\Delta}x + T_L\kappa_L\right), \\ T_R(x) &= \frac{1}{\kappa_R}\left(\frac{\kappa_R(T_R - T_L)}{\Delta}x + T_R\kappa_R - \frac{\kappa_R(T_R - T_L)l}{\Delta}\right), \\ \Delta &= l + a\left(\frac{\kappa_R}{\kappa_L} - 1\right), \end{aligned} \tag{39}$$

where $x$ is the longitudinal coordinate, $l = 1$ m is the total length of the rod and $a = 0.5$ m is the interface location.

The numerical solution is obtained for a uniform mesh with 20 cells as shown in Fig. 2. Note that since the numerical framework presented in this works is second–order and the analytical solution is a piece–wise linear temperature profile, the numerical solution must be identical to the analytical solution up to the round–off error. The results presented in Fig. 3 confirm this assumption, validating the GFM discretisation of the temperature laplacian term. Fig. 4 presents the temperature field.



Figure 2: 1–D heat conduction case with two solid materials sharing an interface in the middle.

### 4.2. 2–D flow over a flat heated plate

The second test–case considers a laminar, steady–state flow over a flat plate [25] of unit length, whose bottom side is heated to a constant temperature of $T_s = 310$ K, Fig. 5. Inlet velocity is set to 0.1 m/s, while the inlet temperature is 300 K. A slip wall is used upstream of the plate, while the no–slip wall is assumed at the plate and downstream from the plate. The solid is 1 m long and 0.25 m high. All boundaries are considered adiabatic apart from the bottom wall of the solid and the inlet.

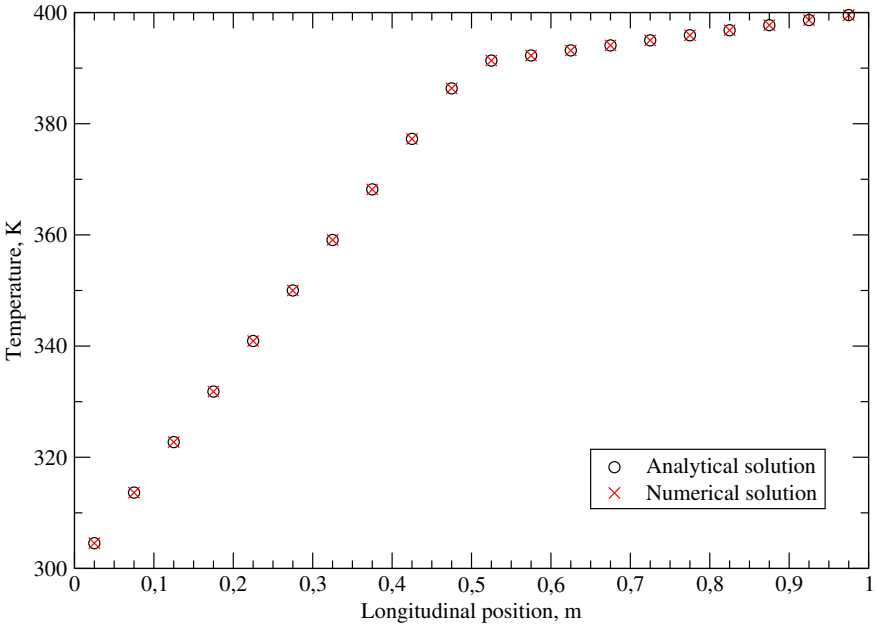



Figure 3: Comparison of analytical and numerical temperature profiles for the 1–D rod case.

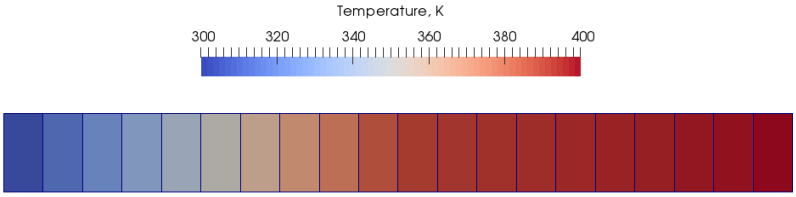



Figure 4: Visualisation of the numerical temperature field.

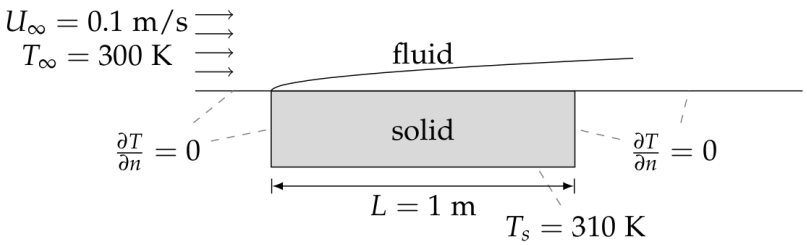
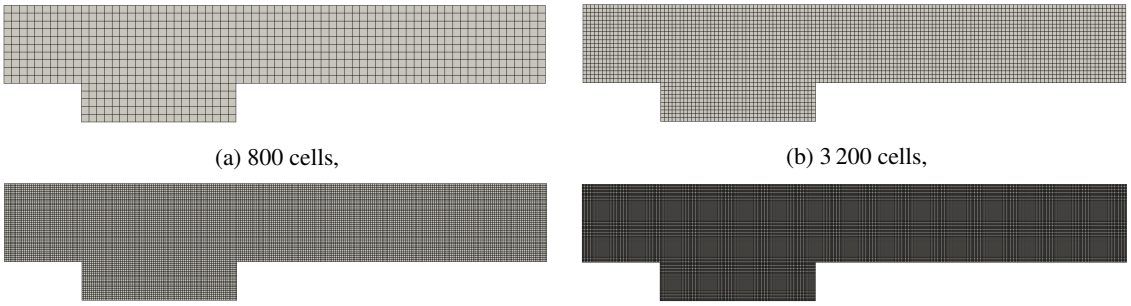



Figure 5: 2–D case: flow over a flat heated plate.

In this study, several different simulations shall be performed based on different combinations of the Reynolds number $R_e$, Prandtl number $P_r$ and the conductivity ratio $k = \kappa_s/\kappa_f$, where $\kappa_s$ and $\kappa_f$ are solid and fluid conductivities, respectively.



(a) 800 cells, (b) 3 200 cells,

(c) 12 800 cells, (d) 51 200 cells,

Figure 6: Four meshes used in the mesh refinement study.

#### *4.2.1. Verification*

In order to verify the numerical framework, a case with the parameters presented in Tab. 1 is considered. Simulations are performed on four, block–structured, uniformly refined meshes. The base mesh has 800 cells, while the successively finer meshes are obtained by applying a refinement factor of 2. The four resulting meshes have: i) 800, ii) 3 200, iii) 12 800 and iv) 51 200 cells; and are presented in Fig. 6. Fig. 7 presents the dimensionless temperature at the interface for the four meshes, defined as:

$$\theta = \frac{T - T_{\text{inf}}}{T_s - T_{\text{inf}}}\,, \tag{40}$$

where $T_s = 310$ K is the temperature at the bottom of the solid and $T_{\text{inf}} = 300$ K is the temperature at the inlet. The results presented in Fig. 7 indicate that the convergence with mesh refinement is achieved. Furthermore, the solution obtained with the coarsest mesh (Mesh 1) is fairly close to the asymptotic solution, rendering coarse–mesh simulations useful from engineering design point of view. Note that the coarse mesh has only 20 cells across the solid–fluid interface.

In order to formalise the mesh refinement study, a procedure by Eça and Hoekstra [26] is employed. Numerical uncertainty generally consists of uncertainty arising from discretisation (spatial and temporal) and iterative uncertainty. In order to keep the iterative uncertainty low, the simulations have been performed until the residuals for all

Table 1: Parameters of the verification case.

| | | |
|---|---|---|
| Prandtl number | $P_r$ | 0.01 |
| Reynolds number | $R_e$ | 500 |
| Conductivity ratio | k | 1 |
| Fluid conductivity | $\kappa_f$ | 100 W/(mK) |
| Fluid specific heat capacity | $c_f$ | 5000 J/K |
| Fluid density | $\rho$ | 1 kg/m$^3$ |
| Solid conductivity | $\kappa_s$ | 100 W/(mK) |
| Solid specific heat capacity | $c_s$ | 100 J/K |

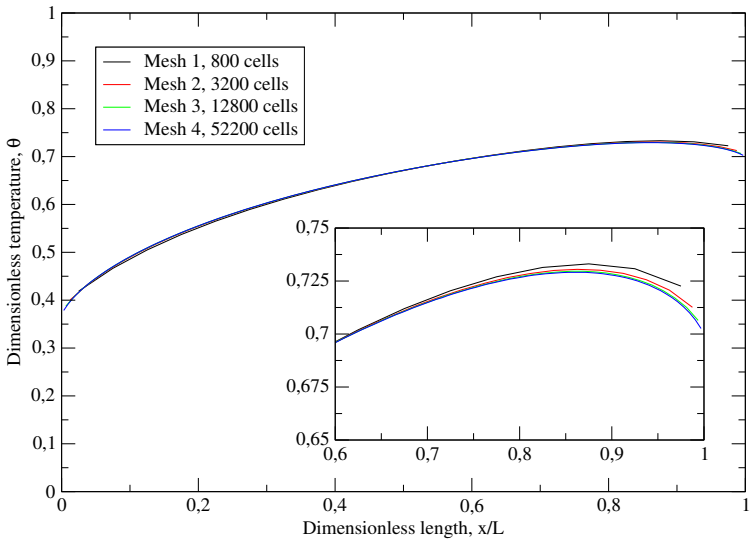



Figure 7: Temperature profiles for the four meshes at the solid–fluid interface.

Table 2: Average dimensionless temperature at the interface.

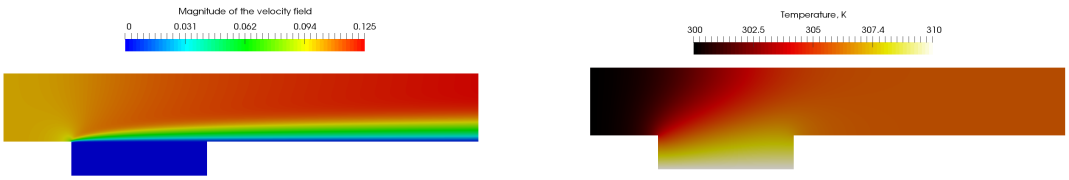
| Number of cells | $\theta$ |
|---|---|
| 800 | 0.63917 |
| 3 200 | 0.63894 |
| 1 2800 | 0.63867 |
| 51 200 | 0.63853 |

Table 3: Numerical uncertainty assessment.

| Mesh | $h_i$ | $U_\theta$ | $p$ |
|---|---|---|---|
| Mesh 1 | 1 | 0.0% | [1,2] |
| Mesh 2 | 2 | 0.1% | [1,2] |
| Mesh 3 | 4 | 0.1% | [1,2] |
| Mesh 4 | 8 | 0.2% | [1,2] |

equations dropped by six orders of magnitude. The numerical uncertainty is calculated for the average interface temperature presented in Tab. 2. The asymptotic solution is $\theta = 0.63834$, which is 0.03% difference compared to the solution on the finest mesh. The results of the numerical uncertainty analysis are presented in Tab. 3 for all meshes, with a uniform refinement factor of 2. The third column presents the assessed numerical uncertainty for average interface temperature. Results for all meshes have numerical uncertainty lower than 0.2%, where the uncertainty is lower for coarser meshes. The achieved spatial order of accuracy is between the first and the second order, as indicated by the last column in the table.

The velocity and the temperature fields on the finest mesh are presented in Fig. 8.




(a) Velocity field, (b) Temperature field,

Figure 8: The solution fields on the finest mesh.

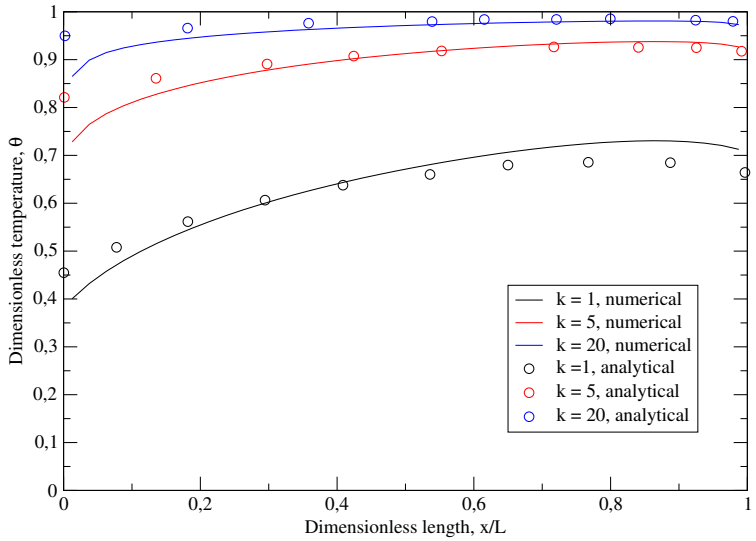



Figure 9: Temperature profiles for different conductivity ratios compared to analytical results.

### *4.2.2. Validation*

The attention is now turned to validation. The same case and set up as the one presented in Sec. 4.2.1 is considered, while the only varying parameter is the conductivity ratio $k$. Three simulations are performed with three conductivity ratios: $k = 1, 5$ and 20. The interface temperature for different conductivities is compared with the analytical solution by Vynnycky *et al.* [25], and is presented in Fig. 9. Good agreement with analytical solution is obtained for all conductivity ratios. The discrepancy between the numerical and the analytical results might be explained by the necessary approximations in the analytical model concerning the boundary layer (see [25], especially considering that the Reynolds number for these cases is kept constant at 500.

### *4.3. 3–D heat sink for electronics cooling*

The third test case considers a plate fin heat sink for electronics cooling applications. The case has been experimentally investigated by [27]. The sketch of a heat sink is presented in Fig. 10, while the parameters used for this case are presented in Tab. 4. The heat source is positioned below the heat sink as shown in Fig. 11, and it is assumed to be a

Table 4: Heat sink parameters used in this study.

| | | |
|---|---|---|
| Number of fins | $n$ | 14 |
| Fin thickness | $t$, mm | 1 |
| Fin height | $H$, mm | 21.8 |
| Fin spacing | $p$, mm | 2.1 |
| Base plate thickness | $t$, mm | 1 |
| Heat sink width | $W$, mm | 41.4 |
| Heat sink length | $L$, mm | 57.2 |

cube with the side length of $a = 12.45$ mm, resulting in the contact area of $A_s = 1.555$ cm$^2$.

The cooling fluid is air and the heat sink is an extruded aluminium alloy with conductivity $\kappa_a = 209$ W/(m K). The transistor specifications given by the manufacturer are the contact area $A_s$ and the thermal resistance between a heat sink and the transistor of $R_{jc} = 0.5K/W$. Since no material properties of the heat source are provided, the conductivity is calculated as:

$$\kappa_{hs} = \frac{0.5a}{R_{jc}A_s} = 80\text{W/(mK)}\,. \tag{41}$$

Computational domain is presented in Fig. 11. Apart from the length of the channel which spans 22× heat sink lengths, the width and the height of the rectangular channel are 7.47 cm and 13.07 cm, respectively.

Inlet velocity is $U = 13.9$ m/s, while the inlet temperature is set to $T_a = 299.6$ C. All walls of the channel are considered no–slip and adiabatic. The gauge pressure at the outlet is set to $p_o = 0$ Pa. The transistor is a heat source with $P = 60.24$ W.

Three simulations are performed on three unstructured meshes:

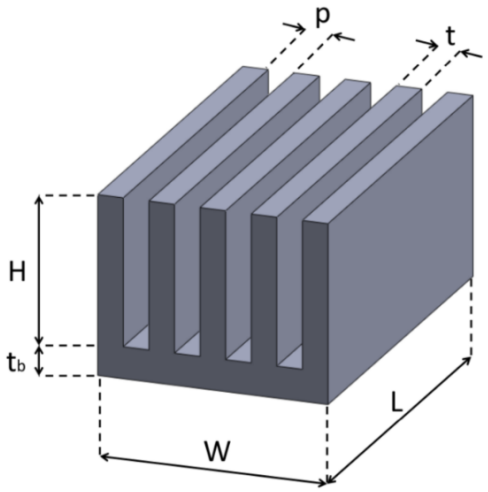



Figure 10: Geometry of a heat sink [27].

1. Coarse mesh ≈ 217 200 cells,
2. Medium mesh ≈ 710 300 cells,
3. Fine mesh ≈ 2 200 000 cells.

Two views of the medium mesh near the heat sink and the transistor are presented in Fig. 12.

Convergence is presented in Fig. 13 for all three meshes. Fig. 13a presents the convergence of average temperature at the interface between the heat source and the heat sink. The coarse and medium mesh results are almost identical, yielding $T = 330.95$ L, while a slightly higher temperature of $T = 331.11$ K is obtained with the fine mesh. The maximum difference between the results is approximately $\Delta T = 0.16$ K, indicating that a very good solution can be obtained with a coarse mesh. The discrepancy between results obtained with coarse/medium and the fine mesh can be explained by non–uniform refinement since unstructured meshes have been used. The residuals for temperature, velocity and pressure are presented in Fig. 13b, Fig. 13c and Fig. 13d, respectively. All equations converge to a lower tolerance on coarser meshes, which is expected considering that the numerical diffusion damps–out naturally occurring flow oscillations.

Ventola *et al.* [27] report the junction temperature as:

$$T_j = T_i + PR_{jc}\,, \tag{42}$$

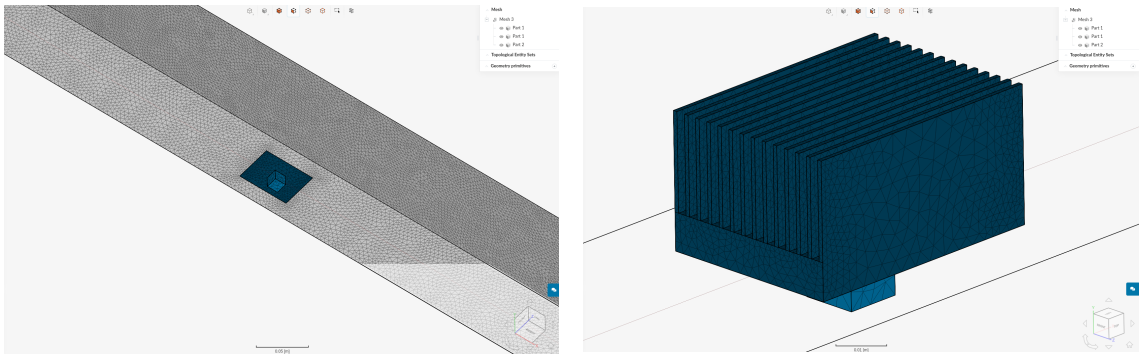
where $T_i$ is the average interface temperature, $P = 60.24W$ is the power output of the transistor and $R_{jc} = 0.5K/W$ is the thermal resistance. Furthermore, the thermal resistance between the cooling air and the junction is obtained with:

$$R_{ja} = \frac{T_j - T_a}{P}\,, \tag{43}$$

where $T_a = 229.6$ K is the ambient temperature. Comparison with experimental results is presented in Tab. 5. The junction temperature obtained with the present numerical framework ranges from 361.07 to 361.23 K, while the experimentally measured result is 357.55 K. The resulting thermal resistance is approximately 1.02 as compared to the experimentally measured value of 0.961. The comparison is encouraging considering the complexity of performing such experimental measurements and the possible differences between the numerical and the experimental setup.

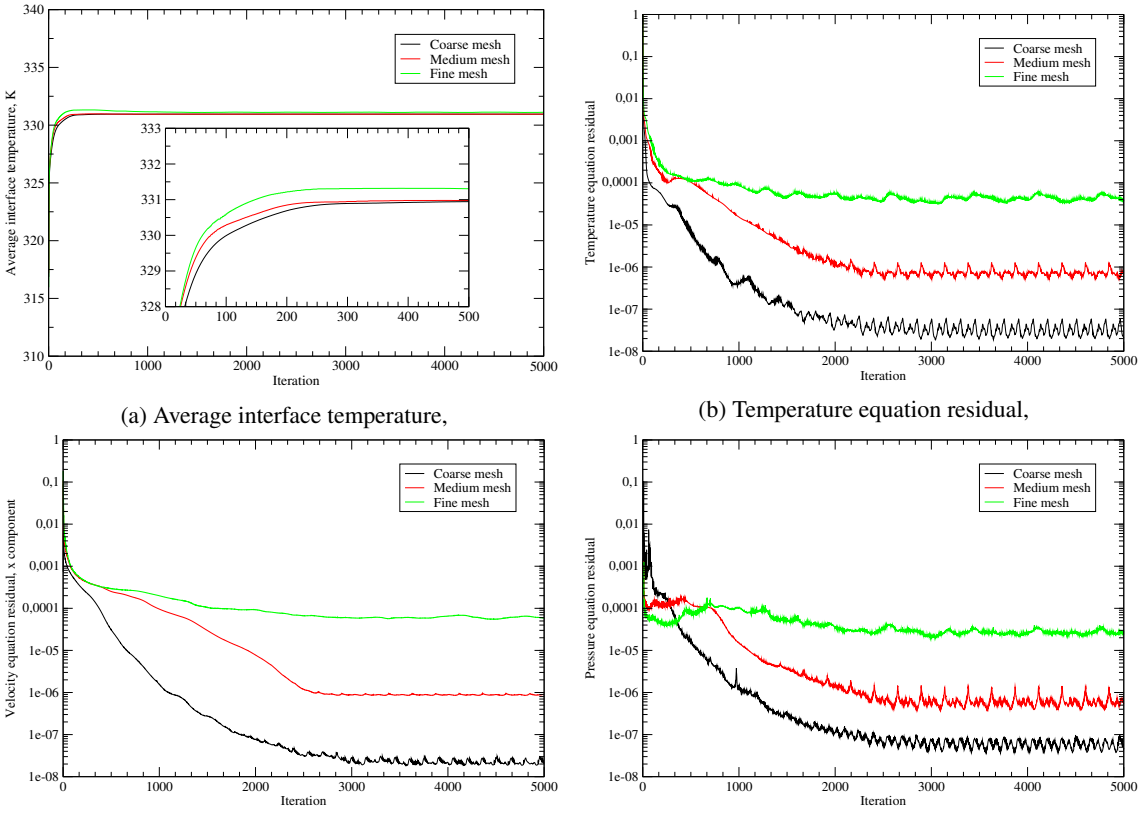



Figure 11: Sketch of the computational domain.



(a) Bottom view of the complete mesh, (b) A view presenting heat sink and power source meshes,

Figure 12: Medium mesh details.

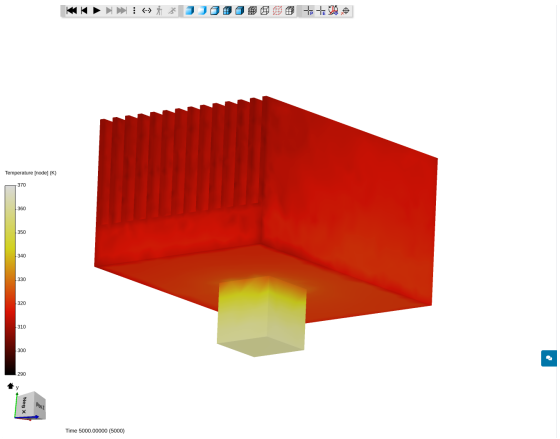



(a) Average interface temperature, (b) Temperature equation residual,

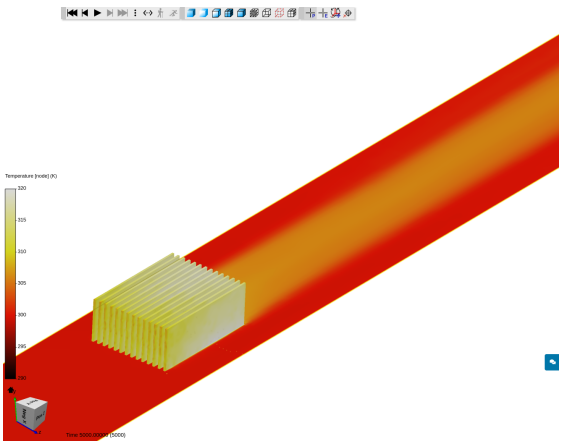
(c) Velocity equation residual, $x$ component, (d) Pressure equation residual,

Figure 13: Convergence plots.

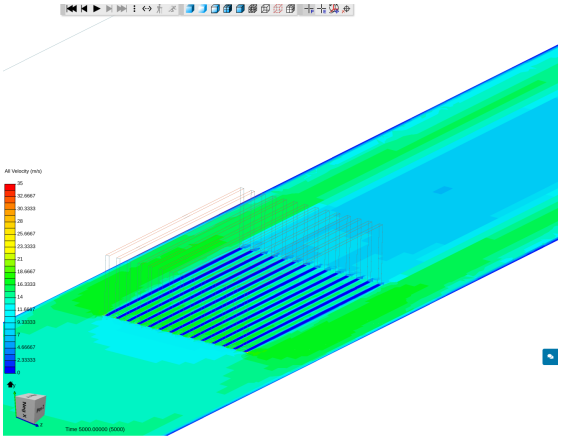
Solution fields are presented in Fig. 14. The temperature field in the heat source and the heat sink is shown in Fig. 14a. The transistor heats up to 370 K while the maximum temperature in the heat sink is approximately 330 K. The heat sink transfers the heat to the surrounding air that convects it further downstream, as seen in Fig. 14b. The flow through the heat sink slows down along the plates (see Fig. 14c) due to viscous resistance and thus decreases the



(a) Temperature distribution in the heat sink and the heat source,



(b) Temperature field in a section through the heat sink,



(c) Velocity field in a section through the heat sink,

Figure 14: Visualisation of the solution fields.

heat transfer efficiency.

## 5. Conclusion

This work presents a region–monolithic numerical framework for CHT industrial–scale problems that often involve complex geometries. The numerical framework is based on second–order accurate, arbitrary polyhedral FV method with the GFM for handling discontinuities at the interfaces between different regions. The method is integrated within

Table 5: Comparison of simulation results with experimental data.

| Item | Coarse mesh | Medium mesh | Fine mesh | Experiment |
|---|---|---|---|---|
| $T_j$ | 361.07 | 361.07 | 361.23 | 357.55 |
| $R_{ja}$ | 1.020 | 1.020 | 1.023 | 0.961 |

SimScale's cloud platform [28], offering ease of use for complex geometries, automatic contact detection, automated meshing and virtually limitless scaling capabilities.

The framework is validated on three cases with ranging complexity. The first two cases are 1–D and 2–D, respectively, which are performed to verify and validate the numerical solution with analytical solutions. The results show that the solution is exact for 1–D heat conduction through a rod consisting of two solids with different conductivities. Furthermore, the 2–D flow case shows that the numerical uncertainties are less than 0.2% and that the convergence with grid refinement is achieved. Finally, a 3–D transistor cooling simulation has been performed and compared to experimental results, yielding good results. The convergence of junction temperature is shown to be fast, giving satisfactory results within the first two hundred iterations.

The usefulness of the present method can be summarized as follows:

- Compared to analytical and experimental results, the numerical solution offers insights into flow physics that can often lead to better designs and avoidance of trial a error approaches,
- The region–monolithic treatment offers faster convergence of the temperature field compared to region–partitioned approaches,
- The integration within SimScale offers scalability and ease of use without having to invest in the infrastructure.

## References


[1] ESI, OpenFOAM: The open source CFD toolbox, https://www.openfoam.com/, [Online; accessed 08 June 2020] (2020).

[2] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College of Science, Technology & Medicine, London (1996).

[3] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object oriented techniques, Comput. Phys. 12 (1998) 620–631.

[4] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416.

[5] W. Bo, J. W. Grove, A volume of fluid method based ghost fluid method for compressible multi-fluid flows, Comput. Fluids 90 (2014) 113–122.

[6] B. Lalanne, L. R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, J. Comput. Phys. 301 (2015) 289–307.

[7] V. Vukčević, H. Jasak, I. Gatin, Implementation of the Ghost Fluid Method for Free Surface Flows in Polyhedral Finite Volume Framework, Comput. Fluids 153 (2017) 1–19. doi:10.1016/j.compfluid.2017.05.003.

[8] V. Vukčević, Numerical modelling of coupled potential and viscous flow for marine applications, Ph.D. thesis, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, PhD Thesis (2016). doi:10.13140/RG.2.2.23080.57605.

[9] F. R. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, Turb. Heat Mass Tran. 4 (2003) 625–632.

[10] D. D. Gray, A. Giorgini, The validity of the Boussinesq approximation for liquids and gases, Int. J. Heat Mass Transfer 19 (1976) 545–551.

[11] J. Huang, P. M. Carrica, F. Stern, Coupled ghost fluid/two–phase level set method for curvilinear body–fitted grids, Int. J. Numer. Meth. Fluids 44 (2007) 867–897. doi:10.1002/fld.1499.

[12] H. K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education Limited, 1995.

[13] J. H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer, 1996.

[14] H. Rusche, Computational fluid dynamics of dispersed two - phase flows at high phase fractions, Ph.D. thesis, Imperial College of Science, Technology & Medicine, London (2002).

[15] M. Arienti, M. Susman, An Embedded Level Set Method for Sharp–Interface Multiphase Simulations of Diesel Injectors, International Journal of Multiphase Flow 59 (2014) 1–14. doi:10.1016/j.ijmultiphaseflow.2013.10.005.

[16] S. Wang, J. Glimm, R. Samulyak, X. Jiao, C. Diao, An Embedded Boundary Method for Two Phase Incompressible Flow, ArXiv e-prints.arXiv:1304.5514.

[17] R. K. Crockett, P. Colella, D. T. Graves, A Cartesian Grid Embedded Boundary Method for Solving the Poisson and Heat Equations with Discontinuous Coefficients in Three Dimensions, J. Comput. Phys. 230 (2010) 613–628. doi:10.1016/j.jcp.2010.12.017.

[18] H. Johansen, P. Colella, A Cartesian Grid Embedding Boundary Method for Poisson's Equation on Irregular Domains, J. Comput. Phys. 147 (1998) 60–85. doi:10.1006/jcph.1998.5965.

[19] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A Front-Tracking Method for Computations of Multiphase Flow, J. Comput. Phys. 169 (2001) 708–759. doi:10.1006/jcph.2001.6726.

[20] H. Jasak, H. G. Weller, Application of the finite volume method and unstructured meshes to linear elasticity, Int. J. Numer. Methods Eng. 48 (2000) 267–287.

[21] I. Demirdžić, On the Discretization of the Diffusion Term in Finite–Volume Continuum Mechanics, Numer. Heat Transfer, Part B 68 (2015) 1–10. doi:10.1080/10407790.2014.985992.

[22] S. V. Patankar, D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transf. 15 (1972) 1787–1806.

[23] C. M. Rhie, W. L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.

[24] Y. Saad, Iterative Methods for Sparse Linear Systems (2nd Edition), Society for Industrial and Applied Mathematics Philadelphia, SIAM, 2003.

[25] M. Vynnycky, S. Kimura, K. Kanev, I. Pop, Forced convection heat transfer from a flat plate: The conjugate problem, Int. J. Heat Mass Transfer 41 (1) (1998) 45–59.

[26] L. Eca, M. Hoekstra, A procedure for the estimation of the numerical uncertainty of cfd calculations based on grid refinement studies, J. Comput. Phys. 262 (2014) 104–130. doi:10.1016/j.jcp.2014.01.006.

[27] Ventola, L. and Curcuruto, G. and Fasano, M. and Fotia, S. and Pugliese, V. and Chiavazzo, E. and Asinari, P., Unshrouded Plate Fin Heat Sinks for Electronics Cooling: Validation of a Comprehensive Thermal Model and Cost Optimization in Semi–Active Configuration, Energies 9 (608). doi:10.3390/en9080608.

[28] SimScale, Simulation software reinvented for the web, www.simscale.com, [Online; accessed 08 July 2020] (2020).