Novel Method for Coupled Radiation-Conduction Simulations in Complex Geometries

Author(s):
Perraudin, David

Publication Date:
2014

Permanent Link:
https://doi.org/10.3929/ethz-a-010337016

Rights / License:
In Copyright - Non-Commercial Use Permitted
Novel Method for Coupled Radiation-Conduction Simulations in Complex Geometries

David Perraudin

Master Thesis
Professorship of Renewable Energy Carriers
Eidgenössische Technische Hochschule Zürich

Supervisors: Dr. A. Haselbacher, Prof. A. Steinfeld

July 29, 2014
Don’t matter if we lose our way,
It’s written that we’ll meet,
At least, that’s what I heard you say,
A thousand kisses deep

- Leonard Cohen
Contents

1 Introduction 1

2 ROCFLU 2
  2.1 Grids ................................................................. 2
  2.2 Preprocessing ....................................................... 3
  2.3 User Input .......................................................... 4
  2.4 Spatial Discretisation ............................................... 4
    2.4.1 Gradient Reconstruction ...................................... 4
    2.4.2 Reconstruction Close to Boundaries .......................... 6
    2.4.3 Least Squares Problems ...................................... 7
    2.4.4 Least Squares Problems with Constraints ................. 7
    2.4.5 Modifying Gradients .......................................... 8
  2.5 Computing Fluxes ................................................. 8
  2.6 Time Integration ................................................ 8

3 Radiative Heat Transfer 9
  3.1 Opaque Surfaces .................................................. 9
    3.1.1 Black Surfaces ............................................... 10
    3.1.2 Real Surfaces ................................................. 11
      3.1.2.1 Emission ................................................ 11
      3.1.2.2 Absorption .............................................. 11
      3.1.2.3 Reflection .............................................. 12
    3.1.3 Kirchhoff's Law ............................................. 13
  3.2 Analytical Solutions ........................................... 13
    3.2.1 Configuration Factors ...................................... 14
    3.2.2 Radiosity Method for Diffuse Reflection ................. 14
  3.3 Monte Carlo Ray Tracing in ROCFLU 15
    3.3.1 User Input .................................................. 15
    3.3.2 Diffuse Emission ........................................... 15
      3.3.2.1 Origin of Ray .......................................... 16
      3.3.2.2 Azimuthal Angle ....................................... 16
      3.3.2.3 Polar Angle ............................................ 16
    3.3.3 Ray Tracing ................................................ 17
7.1.1.2 Gray Diffuse Surfaces ........................................ 65
7.1.1.3 Gray Diffuse and Specular Surfaces ............................ 68
7.1.2 Generic Reactor Geometry ....................................... 72
7.2 Solid-Solid Interaction ........................................... 77
7.3 Fluid Solid Radiation Thermal Interaction ......................... 79

8 Conclusion .................................................................. 84

Appendices .................................................................... 88

A Thesis description ........................................................ 89
## Symbols

\(a\) \hspace{1cm} \text{speed of sound} \hspace{1cm} \text{m/s}

\(a_\lambda\) \hspace{1cm} \text{absorption coefficient} \hspace{1cm} 1/\text{m}

\(A_i\) \hspace{1cm} \text{surface } i \hspace{1cm} \text{m}^2

\(c_0 = 2.998 \cdot 10^8\) \hspace{1cm} \text{speed of light in vacuum} \hspace{1cm} \text{m/s}

\(c_v\) \hspace{1cm} \text{specific heat capacity} \hspace{1cm} \text{J/kg/\textdegree K}

\(d = \frac{D \cdot \Delta t}{\Delta x^2}\) \hspace{1cm} \text{diffusion number} \hspace{1cm} -

\(D = \frac{\kappa}{\rho c}\) \hspace{1cm} \text{thermal diffusivity} \hspace{1cm} \text{m}^2/\text{s}

\(F_{k-i}\) \hspace{1cm} \text{configuration factor} \hspace{1cm} -

\(F\) \hspace{1cm} \text{cumulative probability density} \hspace{1cm} -

\(h\) \hspace{1cm} \text{Planck’s constant} \hspace{1cm} \text{J} \cdot \text{s}

\(k_B = 1.381 \cdot 10^{-23}\) \hspace{1cm} \text{Boltzmann constant} \hspace{1cm} \text{J/\textdegree K}

\(\vec{n}\) \hspace{1cm} \text{normal vector} \hspace{1cm} \text{m}

\(q_c\) \hspace{1cm} \text{conductive heat flux} \hspace{1cm} \text{W/m}^2

\(q_{i,k}\) \hspace{1cm} \text{incoming radiative heat flux} \hspace{1cm} \text{W/m}^2

\(q_k\) \hspace{1cm} \text{net radiative heat flux} \hspace{1cm} \text{W/m}^2

\(q_{o,k}\) \hspace{1cm} \text{outgoing radiative heat flux} \hspace{1cm} \text{W/m}^2

\(Q_{i,k}\) \hspace{1cm} \text{incoming radiative power} \hspace{1cm} \text{W}

\(Q_k\) \hspace{1cm} \text{net radiative power} \hspace{1cm} \text{W}

\(Q_{o,k}\) \hspace{1cm} \text{outgoing radiative power} \hspace{1cm} \text{W}

\(r\) \hspace{1cm} \text{radial coordinate} \hspace{1cm} \text{m}

\(t\) \hspace{1cm} \text{time} \hspace{1cm} \text{s}

\(T\) \hspace{1cm} \text{temperature} \hspace{1cm} \text{K}

\(T_s\) \hspace{1cm} \text{surface temperature} \hspace{1cm} \text{K}

\(U\) \hspace{1cm} \text{uniform distribution in the interval } [0, 1) \hspace{1cm} -

\(\alpha\) \hspace{1cm} \text{absorptivity} \hspace{1cm} -

\(\delta_{ki}\) \hspace{1cm} \text{Kronecker delta} \hspace{1cm} -

\(\delta_x \ldots, \delta_y \ldots, \delta_z \ldots\) \hspace{1cm} \text{approximation of spatial gradients} \hspace{1cm} 1/\text{m}

\(\delta_t\) \hspace{1cm} \text{approximation of temporal gradient} \hspace{1cm} 1/\text{s}

\(\Delta \vec{r}_{0i}\) \hspace{1cm} \text{vector from point 0 to } i \hspace{1cm} \text{m}

\(\Delta t\) \hspace{1cm} \text{discrete time step} \hspace{1cm} \text{s}
\[ \Delta x, \Delta y, \Delta z \] discrete distance \[ \text{m} \]

\[ \epsilon = \sqrt{\kappa \rho c_v} \] effusivity \[ \frac{\text{W} \cdot \text{s}^{0.5}}{\text{m}^2 \cdot \text{K}} \]

\[ \varepsilon \] emissivity \[ \text{--} \]

\[ \kappa \] heat conductivity \[ \frac{\text{W}}{\text{m} \cdot \text{K}} \]

\[ \lambda \] wavelength \[ \text{m} \]

\[ \mu \] dynamic viscosity \[ \text{Pa} \cdot \text{s} \]

\[ \vec{\nabla} \] nabla operator \[ \left[ \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right]^T \] \[ \frac{1}{\text{m}} \]

\[ \varphi \] azimuthal angle \[ \text{rad} \]

\[ \phi \] generic quantity \[ \text{--} \]

\[ \rho \] density \[ \frac{\text{kg}}{\text{m}^3} \]

\[ \rho \] reflectivity \[ \text{--} \]

\[ \sigma = 5.6704 \cdot 10^{-8} \] Stefan-Boltzmann constant \[ \frac{\text{W}}{\text{m}^2 \cdot \text{K}^4} \]

\[ \tau \] characteristic time scale \[ \text{s} \]

\[ \theta \] polar angle \[ \text{rad} \]

\[ \xi \] weights \[ \text{--} \]

\[ j \] spatial index \[ \text{--} \]

\[ n \] temporal index \[ \text{--} \]

\[ i, k \] indices \[ \text{--} \]

\[ i \] evaluated at point i \[ \text{--} \]

\[ F, S \] in the fluid or solid domain \[ \text{--} \]

\[ T \] tranposed \[ \text{--} \]
Abstract

The starting point of this master thesis is an in-house, unsteady, unstructured, three dimensional, parallel CFD tool called ROCFLU. In order to be able to simulate radiative heat transfer between discretised surfaces of arbitrary temperature profiles, a Monte Carlo ray tracing algorithm is implemented within this code and successfully verified using analytical results and the in-house Monte Carlo ray tracing software VeGaS+.

A novel algorithm for fluid structure thermal interaction (also called conjugate heat transfer) is presented. In contrast to existing approaches that exchange boundary conditions between a fluid and a solid solver, this novel approach aims to use one solver only and to enforce continuity of temperature and heat fluxes at the interface. A simplified model problem of two solids with different properties and at different temperatures brought into contact is used to assess the stability and accuracy of such an algorithm in one dimension. Based on good results, the algorithm is implemented in the unstructured and three dimensional CFD code ROCFLU. The implementation is successfully verified using the model problem and the corresponding analytical solution.

To conclude, a three dimensional, unsteady simulation of a generic reactor geometry coupling radiation with fluid structure thermal interaction is presented to illustrate the applicability of the developed software.
Chapter 1

Introduction

With increasing availability of computational resources, CFD simulations gain importance in research and other areas. Many algorithms and software packages to simulate fluid flow exist. Recently, it has become possible to simulate fluid flow coupled with other phenomena, including radiative heat exchange, fluid structure interaction, and others with commercial software. The case of a solid material in contact with a fluid is usually modelled by prescribing a constant temperature or heat flux at the boundary of the fluid domain together with a no slip condition. In certain cases it is of interest to compute the temperature profile in a solid material in contact with a fluid. The coupling at the interface between a fluid and a solid is called fluid structure thermal interaction or conjugate heat transfer. Fluid structure interaction is of special interest in the context of solar reactors where, in addition, radiative heat exchange is the dominant mode of heat transfer. The aim of this thesis is to add the capability of simulating radiative heat transfer and fluid structure thermal interaction as well as the coupling of these to ROCFLU, an in-house CFD tool. An important advantage of in-house software is that the source code, and thus the numerical methods, are known and can be adapted, whereas the detailed functioning of commercial software is usually a company secret.

The common approach for fluid structure interaction is based on two separate solvers that exchange boundary conditions [21]. In the framework of this thesis, a novel approach is introduced that aims at implementing the simulation of the temperature profile in solid materials into an existing CFD code. This has the advantage that the same methods for gradient reconstruction and time integration are used in the fluid and solid domains, which ensures a maximal level of consistency from a numerical point of view. It is the goal to implement the coupling by enforcing physical constraints at the interface, meaning continuity of temperature and heat fluxes. While this may seem to be the most natural way of coupling, it is not the state of the art in the CFD community. Therefore, it is important to assess the stability and accuracy properties of such an algorithm.

In order to be able to simulate solar reactors, it is necessary to compute the radiative heat transfer. It is the goal to implement a Monte Carlo ray tracing algorithm directly in ROCFLU. This allows to couple the radiative heat transfer with the fluid structure interaction and will finally make it possible to simulate realistic solar devices using ROCFLU.
Chapter 2

ROCFLU

ROCFLU is an unsteady, parallel, finite volume solver for the compressible Navier Stokes equations written in FORTRAN 90. It was originally developed at “The University of Illinois Center for Simulation of Advanced Rockets” and later at the “University of Florida Department of Mechanical and Aerospace Engineering” under the supervision of Dr. Andreas Haselbacher.

This chapter focuses on describing the aspects that are relevant for the modifications that are performed in the framework of this thesis. The gradient reconstruction is of particular interest. What is described here is valid for ROCFLU before the implementation of the changes performed in the framework of this thesis and described in section 3.3 and chapter 6. For a more complete description the reader is referred to the ROCFLU manual [18].

2.1 Grids

ROCFLU is written for unstructured grids. It is capable of processing grids consisting of hexahedra, prisms, pyramids and tetrahedra (figure 2.1). Different file formats can be imported. In the framework of this thesis the grid generator “CENTAUR” (https://www.centaursoft.com) is used and more specifically, the CENTAUR in-house format.

Figure 2.1: Control volumes: prism, pyramid, tetrahedron and hexahedron (clockwise from left bottom)
2.2 Preprocessing

Before starting a simulation, the input files must be preprocessed. The original files are not changed, but new ones created. This is because the grid is converted to the ROCFLU grid format and for parallel simulations, the grid must be partitioned into smaller entities, called regions. Also before starting a simulation, the flow field must be initialised. The preprocessing consists of three steps that are performed using the three executables rflumap, rflupart and rfluinit.

In a first step, the mapping file is created. This file determines into how many regions the grid will be partitioned and how many cores or processors will be used for the simulation. Note that several regions can be processed on one single core or processor, if the user chooses to do so (figure 2.2).

In a next step, the grid is partitioned. To name just a few of the tasks performed, the input grid is converted, split into several smaller grids, virtual cells are determined, information for communication between cores and processors, and renumbering lists are written out. Virtual cells are all those cells, for which the cell value is computed in one region and that same value is required to compute the fluxes for cells in another region. Figure 2.3 illustrates the necessity of virtual cells for the example of a domain inner face marked with a square point. The points at which the values are required for that given face are called stencil members and illustrated by connection with dashed lines. The region boundary is indicated in red.

Finally, the flow field is initialised according to user input.

Figure 2.2: Example of grid partitioned into three regions and attribution to two cores or processors
Figure 2.3: Illustration of virtual cells (in grey), serial case with region boundary indicated in red (left), partitioned grids with virtual cells highlighted in grey (middle, right), stencil members for domain inner face (square point) are indicated with dashed lines

2.3 User Input

The user input is read in from a file called \[\text{casename}.inp\] and the boundary conditions from \[\text{casename}.bc\]. A detailed description of the input is given in the manual [18]. The boundary conditions must be specified for all patches that are defined in the grid input. Patches are parts of the domain boundary and must be defined in the process of mesh generation.

2.4 Spatial Discretisation

ROCFLU uses a finite volume algorithm and a cell centered approach. Values are stored and computed at the center of the control volumes. For a second or higher order accurate method in space, the values of certain variables and their derivatives at control volume faces must be approximated. The process of finding those values is called reconstruction. In the following, the implementation of reconstruction in ROCFLU is described. Since it is relevant in that context, least squares problems and least squares problems with constraints are discussed.

2.4.1 Gradient Reconstruction

For gradient reconstruction, a method based on truncated Taylor series introduced by Barth [5] is used. For a point of interest \(0\), a cloud of points \(i\) around that point is defined (figure 2.4). These points are called stencil.
For each of the $m$ points in a stencil, a truncated Taylor series is formulated for a certain quantity of interest $\phi$

$$\phi_i = \phi_0 + (\nabla \phi)|_0 \cdot \Delta \vec{r}_{0i}$$  \hspace{1cm} (2.1)

$(\nabla \phi)|_0$ denotes the gradient at the point 0, and $\Delta \vec{r}_{0i}$ the vector pointing from point 0 to point $i$. The three components of the approximation of the spatial gradient $(\delta_x \phi, \delta_y \phi, \delta_z \phi)$ are written into the unknown vector $x$

$$x = [\delta_x \phi \hspace{0.2cm} \delta_y \phi \hspace{0.2cm} \delta_z \phi]^T$$  \hspace{1cm} (2.2)

For gradient reconstruction at cell centers, the $m$ equations are written in matrix form as $A \cdot x = c$ with

$$A = \begin{pmatrix}
\Delta x_{01} & \Delta y_{01} & \Delta z_{01} \\
\Delta x_{02} & \Delta y_{02} & \Delta z_{02} \\
\vdots & \vdots & \vdots \\
\Delta x_{0m} & \Delta y_{0m} & \Delta z_{0m}
\end{pmatrix}$$  \hspace{1cm} (2.3)

$$c = \begin{bmatrix}
(\phi_1 - \phi_0) \\
(\phi_2 - \phi_0) \\
\vdots \\
(\phi_m - \phi_0)
\end{bmatrix}^T$$  \hspace{1cm} (2.4)

At faces or vertices the value $\phi_0$ at the face or vertex is not known, and thus included in the vector of unknowns. Again, the equations are written as $A \cdot x = c$ with an adapted matrix $A$ and right hand side $c$.
\[
A = \begin{pmatrix}
1 & \Delta x_{01} & \Delta y_{01} & \Delta z_{01} \\
1 & \Delta x_{02} & \Delta y_{02} & \Delta z_{02} \\
\vdots & \vdots & \vdots & \vdots \\
1 & \Delta x_{0m} & \Delta y_{0m} & \Delta z_{0m}
\end{pmatrix}
\]  
(2.5)

\[
x = \begin{bmatrix}
\phi_0 \\
\delta_x \phi \\
\delta_y \phi \\
\delta_z \phi
\end{bmatrix}^T
\]  
(2.6)

\[
c = \begin{bmatrix}
\phi_1 \\
\phi_2 \\
\vdots \\
\phi_m
\end{bmatrix}^T
\]  
(2.7)

In practice it is advantageous to weight the equations, to ensure that equations between points that are far apart from each other have less impact on the reconstructed gradients than points that are closer. The weights \(\xi_{0i}\) for the equation relating the point 0 to the point \(i\) can be chosen according to \([16], [17]\)

\[
\xi_{0i} = \|\Delta \vec{r}_{0i}\|^{-1}
\]  
(2.8)

In matrix form this can be rewritten as

\[
\tilde{W} \cdot A \cdot x = \tilde{W} \cdot c
\]  
(2.9)

with the \(m\)-by-\(m\) matrix \(\tilde{W}\) defined as

\[
\tilde{W} = \begin{pmatrix}
\xi_{01} & 0 & \cdots & 0 \\
0 & \xi_{02} & \vdots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \xi_{0m}
\end{pmatrix}
\]  
(2.10)

Note that there are more points contained in the stencils than there are unknowns. This is naturally the case when using unstructured grids and can be recognised in figure 2.4 for a two dimensional case. In three dimensions, the stencils are even bigger. Thus equation 2.9 is an overdetermined system of equations and must be solved in a least squares manner (section 2.4.3).

2.4.2 Reconstruction Close to Boundaries

If certain variables or their derivatives are prescribed at boundaries, they can be included in the reconstruction as constraints. This is a consistent way of including boundary conditions in the gradient reconstruction. More on that matter can be found in \([16]\) and \([17]\). The idea is that, in addition to the equations obtained from the formulation of the truncated Taylor series, which are solved in a least squares manner, equations are found that must be fulfilled exactly. In ROCFLU constrained reconstruction is implemented for Dirichlet temperature boundary conditions.

This idea of constrained reconstruction is central to the implementation of the fluid structure interaction in ROCFLU (chapter 6). The mathematical background is given in section 2.4.4.
2.4.3 Least Squares Problems

When solving a system of equations it can occur that the number of equations exceeds the number of unknowns. In that case, no unique solution can be found. Instead, a criterion is defined, that must be minimised. Assume that a system of equations is given as

$$A \cdot x = c \quad (2.11)$$

where \(A\) is a \(m\)-by-\(n\) matrix, \(x\) a \(n\)-by-1 vector, \(c\) a \(m\)-by-1 vector and \(m > n\) holds.

It makes sense to require that the two norm of the vector of the error is minimised

$$\min_x \| A \cdot x - c \|_2 \quad (2.12)$$

Different methods exist to solve equation 2.12. The number of equations can be reduced by writing equation 2.11 in its normal form

$$A^T \cdot A \cdot x = A^T \cdot c \quad (2.13)$$

It can be shown that the solution of 2.13 fulfills 2.12 [6]. The solution to equation 2.12 can also be found using the QR-decomposition or the single value decomposition of \(A\) [6].

2.4.4 Least Squares Problems with Constraints

If in addition to solving \(m\) equations in a least square manner, \(p\) constraints are imposed on the same \(n\) unknowns, one talks of a least squares problem with constraints. It is required that the solution fulfills the constraints exactly.

$$\min_{x} \| A \cdot x - c \|_2 \text{ subject to } B \cdot x = d \quad (2.14)$$

\(A\) is a \(m\)-by-\(n\) matrix, \(B\) a \(p\)-by-\(n\) matrix, \(c\) a \(m\)-by-1 vector, \(d\) a \(p\)-by-1 vector and \(x\) a \(n\)-by-1 vector.

Different methods exist to solve the problem 2.14. For the following considerations, the method of direct elimination is relevant [6]. The routine \texttt{ddglse} in the \texttt{LAPACK} [2] package implements this method. The following restrictions apply:

- \(p \leq n \leq m + p\)
- \(B\) has full row rank \(p\)
- the matrix \(\begin{pmatrix} A \\ B \end{pmatrix}\) has full column rank \(n\)

\(p \leq n\) expresses the fact that the number of constraints is limited by the number of unknowns. This must be the case since constraints are fulfilled exactly. \(m + p\) is the total number of equations. If \(n \leq m + p\) is not fulfilled, the problem is underdetermined and can not be solved using this algorithm. If \(B\) does not have full rank at least two constraints are linearly dependent. Thus at least two constraints are either identical or contradictory. In this case the algorithm fails.
2.4.5 Modifying Gradients

Different methods for modifying gradients are implemented in ROCFLU. In the framework of this thesis “weighted essentially non-oscillatory (WENO) reconstruction in which gradient components are weighted according to the inverse square of their magnitude” [18] will be used in the final, three dimensional simulations.

2.5 Computing Fluxes

Different schemes for flux computation are implemented in ROCFLU. In the framework of this thesis, the AUSM+ scheme [25] will be used.

2.6 Time Integration

For time integration, the fourth-order accurate classical Runge-Kutta scheme is used. The Butcher tableau is given in table 2.1. Given an expression for the right hand side

\[
\frac{d\phi}{dt} = RHS(\phi, t) \quad (2.15)
\]

the full method reads

\[
\phi^{n+1} = \phi^n + \frac{\Delta t}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \quad (2.16)
\]

with the following values for the \(k_i\)

\[
\begin{align*}
\phi_1 &= \phi^n & k_1 &= RHS(\phi_1, t^n) \\
\phi_2 &= \phi^n + \frac{1}{2} \Delta t \cdot k_1 & k_2 &= RHS(\phi_2, t^n + \frac{1}{2} \Delta t) \\
\phi_3 &= \phi^n + \frac{1}{2} \Delta t \cdot k_2 & k_3 &= RHS(\phi_3, t^n + \frac{1}{2} \Delta t) \\
\phi_4 &= \phi^n + \Delta t \cdot k_3 & k_4 &= RHS(\phi_4, t^n + \Delta t)
\end{align*}
\]

Table 2.1: Butcher tableau for classical Runge-Kutta scheme, from [7]

\[
\begin{array}{c|cccc}
0 & 0 & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\
\frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\
1 & 0 & 0 & 1 & \\
\hline
\end{array}
\]
Chapter 3

Radiative Heat Transfer

The aim of this chapter is to summarise the theoretical background on radiative heat exchange between opaque surfaces in non-participating media required to understand the simulation of radiative heat transfer in ROCFLU.

It describes the implementation of the Monte Carlo ray tracing, points out the most important simplifications applied, and may be a source of ideas for more sophisticated features that could be integrated into ROCFLU.

3.1 Opaque Surfaces

The following focuses on opaque surfaces. Radiative energy incident on an opaque surface can either be absorbed or reflected. Transmission is excluded.

In the most general case, the radiative surface properties are assumed to depend on the surface temperature \(T_S\), the wavelength \(\lambda\) and direction of incident and emitted or reflected radiation. In practical cases, it is common to use integral quantities.

Different definitions of spherical coordinates exist. Figure 3.1 illustrates the convention used in this thesis together with the orientation of the coordinate system with respect to a differential surface \(dS\).

This section starts with the description of black surfaces. Black surfaces are ideal absorbers and emitters. The properties of real surfaces are commonly related to those of black bodies by means of emissivity \(\varepsilon\), absorptivity \(\alpha\) and reflectivity \(\rho\).

The notation is elaborate. A subscript \(\lambda\) indicates that the respective quantity is wavelength dependent and in the case of heat fluxes or intensities defined per unit wavelength \(d\lambda\). Quantities with a subscript \(b\) are emitted from a black body. An apostrophe indicates a directional dependency and in the case of heat fluxes or intensities that the quantity is defined per unit solid angle \(d\omega\). Intensities \((i)\) are given per unit projected area \(\cos(\theta)dS\) whereas powers \((e)\) are given per unit area \(dS\).
3.1.1 Black Surfaces

Table 3.1 summarises the important definitions used in the context of radiation emitted from black surfaces. A black surface is an ideal absorber, such that it absorbs all incoming radiation. Furthermore, a black surface is an isotropic emitter, meaning that the directional spectral intensity $i_{\lambda b}'$ shows no directional dependency. It is given by Planck’s law [22]

$$i_{\lambda b}' = \frac{2hc_0^2}{\lambda^5 \left[ \exp \left( \frac{hc_0}{k_B T} \right) - 1 \right]} \quad (3.1)$$

c_0$ is the speed of light in vacuum, $h$ is Planck’s constant, and $k_B$ is the Boltzmann constant. The directional spectral emitted power is dependent on the polar angle $\theta$ and given by Lambert’s cosine law [22]

$$e'_{\lambda b} = i'_{\lambda b} \cdot \cos(\theta) \quad (3.2)$$

The remaining definitions given in table 3.1 are integral quantities. The corresponding expressions can be obtained by integration over all wavelengths or over all solid angles, starting from equation 3.1 or 3.2.
Table 3.1: Summary of definitions for black bodies

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Dependencies</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i'_\lambda)</td>
<td>directional spectral intensity</td>
<td>(T_S, \lambda)</td>
<td>(i'_\lambda = \frac{2hc^2}{\lambda^5 \left[ \exp \left( \frac{hc}{\lambda kT_S} \right) - 1 \right]})</td>
</tr>
<tr>
<td>(i'_b)</td>
<td>directional intensity</td>
<td>(T_S)</td>
<td>(i'_b = \frac{\pi}{4} T_S^4)</td>
</tr>
<tr>
<td>(i_\lambda)</td>
<td>spectral intensity</td>
<td>(T_S, \lambda)</td>
<td>(i_\lambda = \frac{2}{\pi} \cdot i'_\lambda)</td>
</tr>
<tr>
<td>(e'_\lambda)</td>
<td>directional spectral power</td>
<td>(T_S, \lambda, \theta)</td>
<td>(e'<em>\lambda = i'</em>\lambda \cdot \cos(\theta))</td>
</tr>
<tr>
<td>(e'_b)</td>
<td>directional power</td>
<td>(T_S, \theta)</td>
<td>(e'_b = i'_b \cdot \cos(\theta))</td>
</tr>
<tr>
<td>(e_\lambda)</td>
<td>spectral power</td>
<td>(T_S, \lambda)</td>
<td>(e_\lambda = \frac{\pi}{4} \cdot i'_\lambda)</td>
</tr>
<tr>
<td>(e_b)</td>
<td>power</td>
<td>(T_S)</td>
<td>(e_b = \pi \cdot i'_b = \sigma T_S^4)</td>
</tr>
</tbody>
</table>

3.1.2 Real Surfaces

Radiative properties of real surfaces are not trivial to describe if all dependencies are taken into account. The following sections start from the most basic definitions and show how integral quantities are defined. A big amount of definitions is required, such that they are organised in tables.

3.1.2.1 Emission

A black body is an ideal emitter. The fraction of radiative power emitted by a real body to that emitted by a black body at the same temperature is defined as the emissivity. The emissivity in its most basic form depends on wavelength and direction. Table 3.2 summarises the most important definitions.

Table 3.2: Summary of definitions for emission

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Dependencies</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\varepsilon'_\lambda)</td>
<td>directional spectral emissivity</td>
<td>(T_S, \lambda, \theta, \varphi)</td>
<td>(\varepsilon'<em>\lambda = \frac{\varepsilon'</em>\lambda}{\varepsilon'_b})</td>
</tr>
<tr>
<td>(\varepsilon')</td>
<td>directional emissivity</td>
<td>(T_S, \theta, \varphi)</td>
<td>(\varepsilon' = \frac{\varepsilon'_b}{\varepsilon'_b})</td>
</tr>
<tr>
<td>(\varepsilon_\lambda)</td>
<td>spectral emissivity</td>
<td>(T_S, \lambda)</td>
<td>(\varepsilon_\lambda = \frac{\varepsilon_\lambda}{\varepsilon_b})</td>
</tr>
<tr>
<td>(\varepsilon)</td>
<td>emissivity</td>
<td>(T_S)</td>
<td>(\varepsilon = \frac{\varepsilon}{\varepsilon_b})</td>
</tr>
</tbody>
</table>

3.1.2.2 Absorption

A black body is a perfect absorber. Real materials do not absorb all incoming radiation. The fraction of absorbed radiative power to the total incoming radiative power is the definition of absorptivity. The absorptivity in its most basic form depends on direction and wavelength. Table 3.3 gives a summary of important definitions.
Table 3.3: Summary of definitions for absorption

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Dependencies</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d^3Q'_{\lambda,i}$</td>
<td>power incident within $d\omega,d\lambda$, per unit projected surface</td>
<td>$\lambda, \theta, \varphi$</td>
<td>$d^3Q'<em>{\lambda,i} = i'</em>{\lambda,i}d\lambda dS \cos(\theta)d\omega$</td>
</tr>
<tr>
<td>$d^3Q'_{\lambda,a}$</td>
<td>power absorbed within $d\omega,d\lambda$, per unit projected surface</td>
<td>$T_S, \lambda, \theta, \varphi$</td>
<td></td>
</tr>
<tr>
<td>$\alpha'_\lambda$</td>
<td>directional spectral absorptivity</td>
<td>$T_S, \lambda, \theta, \varphi$</td>
<td>$\alpha'<em>\lambda = \frac{d^3Q'</em>{\lambda,a}}{d^3Q'_{\lambda,i}}$</td>
</tr>
<tr>
<td>$\alpha'$</td>
<td>directional absorptivity</td>
<td>$T_S, \theta, \varphi$</td>
<td>$\alpha' = \frac{d^3Q'<em>{\lambda,a}}{d^3Q'</em>{\lambda,i}}$</td>
</tr>
<tr>
<td>$\alpha_\lambda$</td>
<td>spectral absorptivity</td>
<td>$T_S, \lambda$</td>
<td>$\alpha_\lambda = \frac{d^3Q_{\lambda,a}}{d^3Q_{\lambda,i}}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>absorptivity</td>
<td>$T_S$</td>
<td>$\alpha = \frac{dQ_a}{dQ_i}$</td>
</tr>
</tbody>
</table>

3.1.2.3 Reflection

The definition of the reflectivity is more complicated than that of emissivity and absorptivity since it depends on the direction of incidence ($\theta, \varphi$) as well as on that of reflection ($\theta_r, \varphi_r$). The following considerations are restricted to two cases relevant in reality: diffuse and specular reflection. For a more detailed analysis the reader is referred to [22].

In the diffuse case, the reflection of a given intensity incident in a certain direction is equal in all directions. The relevant quantity is called directional hemispherical, indicating that the incident direction is relevant but the reflected intensity is isotropic.

Table 3.4: Summary of definitions for reflection

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Dependencies</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d^3Q'_{\lambda,i}$</td>
<td>power incident within $d\omega,d\lambda$, per unit projected surface</td>
<td>$\lambda, \theta, \varphi$</td>
<td>$d^3Q'<em>{\lambda,i} = i'</em>{\lambda,i}d\lambda dS \cos(\theta)d\omega$</td>
</tr>
<tr>
<td>$d^3Q'_{\lambda,r}$</td>
<td>power reflected into the whole hemisphere as a result of an incident intensity from direction ($\theta, \varphi$)</td>
<td>$T_S, \lambda, \theta, \varphi$</td>
<td></td>
</tr>
<tr>
<td>$\rho'_\lambda$</td>
<td>directional hemispherical spectral reflectivity</td>
<td>$T_S, \lambda, \theta, \varphi$</td>
<td>$\rho'<em>\lambda = \frac{d^3Q'</em>{\lambda,r}}{d^3Q'_{\lambda,i}}$</td>
</tr>
<tr>
<td>$\rho'$</td>
<td>directional hemispherical reflectivity</td>
<td>$T_S, \theta, \varphi$</td>
<td>$\rho' = \frac{d^3Q'<em>{\lambda,r}}{d^3Q'</em>{\lambda,i}}$</td>
</tr>
<tr>
<td>$\rho_\lambda$</td>
<td>hemispherical spectral reflectivity</td>
<td>$T_S, \lambda$</td>
<td>$\rho_\lambda = \frac{d^3Q_{\lambda,r}}{d^3Q_{\lambda,i}}$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>reflectivity</td>
<td>$T_S$</td>
<td>$\rho = \frac{dQ_r}{dQ_i}$</td>
</tr>
</tbody>
</table>
When considering specular reflection, the concept of reflectivity is simplified by the fact that the reflection angle is given by the incidence angle as

\[
\theta_r = \theta \\
\varphi_r = \varphi + \pi
\]

The definition of the same quantities as in the case of diffuse reflection is straightforward.

### 3.1.3 Kirchhoff’s Law

Kirchhoff’s law in its most basic form states that [22]

\[
\alpha'_\lambda(T_S, \lambda, \theta, \varphi) = \varepsilon'_\lambda(T_S, \lambda, \theta, \varphi)
\]

(3.5)

Simplifications can be made if it is assumed that surface properties are independent of the wavelength of incident and emitted light. Such surfaces are termed “grey” referring to human perception. In that case it holds that

\[
\alpha'(T_S, \theta, \varphi) = \varepsilon'(T_S, \theta, \varphi)
\]

(3.6)

The properties of diffuse surfaces are independent of direction such that Kirchhoff’s law simplifies to

\[
\alpha_\lambda(T_S, \lambda) = \varepsilon_\lambda(T_S, \lambda)
\]

(3.7)

Finally, for diffuse and grey surfaces, Kirchhoff’s Law becomes

\[
\alpha(T_S) = \varepsilon(T_S)
\]

(3.8)

The dependency on the surface temperature is often neglected.

### 3.2 Analytical Solutions

Analytical solutions for radiative heat exchange in complex geometries are close to impossible to find. The radiosity method, also called net-radiation method for enclosures [22] can provide analytical solutions for the radiative heat exchange in simple geometries between diffuse grey surfaces. However, strong restrictions that cannot be fulfilled without knowledge of the solution limit the practical applicability of this method.

In this section, the radiosity method is described as it will be used later for validation of the Monte Carlo ray tracing implementation in ROCFLU.
3.2.1 Configuration Factors

In formulating the radiosity method, configuration factors play an important role. Configuration factors assume diffuse emission from all surfaces. For two surfaces $A_1$ and $A_2$, the configuration factor $F_{1\rightarrow2}$ is defined as the ratio of radiative power, diffusely emitted from surface $A_1$, incident on surface $A_2$.

The configuration factors are defined by geometry solely. An important property of configuration factors is the reciprocity relation that holds for any two surfaces

$$A_1 \cdot F_{1\rightarrow2} = A_2 \cdot F_{2\rightarrow1} \quad (3.9)$$

If the considered setup is an enclosure defined by $N$ different surfaces, the following must hold for $k = 1 \ldots N$ due to conservation of energy

$$\sum_{i=1}^{N} F_{k\rightarrow i} = 1 \quad (3.10)$$

3.2.2 Radiosity Method for Diffuse Reflection

The radiosity method is based on several simplifications. The following four requirements must be fulfilled

- uniform temperature and properties over each surface
- uniform incident and emitted radiative heat flux over each surface
- black or gray-diffuse surfaces
- enclosure ($N$ surfaces)

The radiosity ($q_{o,k}$) is defined as

$$q_{o,k} = \varepsilon_k \sigma T_k^4 + \rho_k q_{i,k} = \varepsilon_k \sigma T_k^4 + (1 - \varepsilon_k) q_{i,k} \quad (3.11)$$

where $q_{i,k}$ denotes the incoming radiative heat flux. With the above requirements, the radiosity is constant over each surface. This is the backbone of the method.

The incoming radiative heat fluxes can be expressed as a function of the outgoing heat fluxes and configuration factors as

$$q_{i,k} = \sum_{i=1}^{N} F_{k\rightarrow i} q_{o,i} \quad (3.12)$$

In many applications, the net heat flux $q_k$ for each surface $k$ is of interest

$$q_k = q_{o,k} - q_{i,k} \quad (3.13)$$
The radiosity method can be written in one compact equation [22] as

\[ \sum_{i=1}^{N} \left( \frac{\delta_{ki}}{\varepsilon_i} - F_{k-i} \frac{1 - \varepsilon_i}{\varepsilon_i} \right) q_i = \sum_{i=1}^{N} \left( \delta_{ki} - F_{k-i} \right) \sigma T_i^4 \]

where \( \delta_{ik} \) is the Kronecker delta defined as

\[ \delta_{ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \]

The requirements for that method to be applicable are quite strict. As will be shown in section 7.1.1, the incoming radiative heat flux is not constant in almost any real case. In principle, surfaces with varying temperature or surface properties can be discretised into several smaller surfaces. This could also be done for varying incoming heat fluxes, but requires knowledge of the solution. Non-participating surroundings can be considered as surfaces with \( \rho = 0 \Rightarrow \varepsilon = 1 \) and \( T = 0 \) in order to create an enclosure.

3.3 Monte Carlo Ray Tracing in ROCFLU

A fully functional Monte Carlo ray tracing algorithm for diffuse, grey surfaces has been implemented in ROCFLU in the framework of this thesis, based on an existing basic implementation of ray tracing. Monte Carlo ray tracing is well suited for simulating radiative heat exchange in complex geometries and thus the method of choice. In this section the implementation details are described.

3.3.1 User Input

All surfaces are assumed to be diffuse emitters and grey. For each patch defined on the mesh, the user must provide surface properties assumed constant \((\alpha, \varepsilon, \rho)\) and the mode of reflection (diffuse or specular). Note that since the surfaces are assumed diffuse and grey, Kirchhoff’s law must be fulfilled (equation 3.8). The dependency of surface properties on the surface temperature is neglected.

In the case of a constant temperature boundary, the temperature of the patch is set constant. In all other cases, the wall temperature is a result of the simulation. Also, for each patch the user must specify the number of rays emitted.

3.3.2 Diffuse Emission

The number of rays prescribed is emitted from each patch. In a first step, the total energy emitted from a patch is computed. Each ray carries the same amount of radiative power, namely the total power emitted by this patch divided by the number of rays emitted from this patch. A ray is defined by a point of origin and a direction. The algorithm for ray emission is described in the following.
3.3.2.1 Origin of Ray

Due to the discretisation of the volume, surfaces are automatically discretised into smaller entities, called faces. For a given patch, and for each ray to be emitted from that patch the location of ray emission is determined by first selecting a face. For each ray a face is drawn from a probability distribution. The probability to select a certain face is given by the ratio of power emitted from that face to the total power emitted by that patch. The location on a face is drawn from a uniform distribution.

If a patch is partitioned the ratio of rays emitted from each part of the patch to the total number of rays emitted from that patch is chosen proportionally to the ratio of power emitted from each respective part of the patch to the total power emitted from that patch. The number of rays obtained in that way is rounded towards the nearest integer.

3.3.2.2 Azimuthal Angle

The azimuthal angle $\varphi$ is drawn from a uniform distribution between $[0, 2\pi)$.

3.3.2.3 Polar Angle

The distribution of the polar angle $\theta$ is more intricate to find. The cumulative distribution function $F(\theta)$ can be found by expressing the ratio of the amount of energy emitted between $\theta^* = (0, \theta)$ to the amount of energy emitted between $\theta^* = (0, \pi/2)$

$$F(\theta) = \frac{\int_{\varphi^*=0}^{2\pi} \left( \int_{\theta^*=0}^{\theta} \{ \varepsilon'_{\lambda}(\lambda^*, \theta^*, T_S) \int_{\lambda^*=0}^{\infty} \{ \varepsilon'_{\lambda b}(\lambda^*, T_S) \cos(\theta^*) \} d\lambda^* \sin(\theta^*) \} d\theta^* \right) d\varphi^*}{\int_{\varphi^*=0}^{2\pi} \left( \int_{\theta^*=0}^{\pi/2} \{ \varepsilon'_{\lambda}(\lambda^*, \theta^*, T_S) \int_{\lambda^*=0}^{\infty} \{ \varepsilon'_{\lambda b}(\lambda^*, T_S) \cos(\theta^*) \} d\lambda^* \sin(\theta^*) \} d\theta^* \right) d\varphi^*}$$

Assuming that emission is independent of the azimuthal angle ($\varphi$) leads to

$$F(\theta) = \frac{2\pi}{2\pi} \left( \int_{\theta^*=0}^{\theta} \{ \varepsilon'_{\lambda}(\lambda^*, \theta^*, T_S) \int_{\lambda^*=0}^{\infty} \{ \varepsilon'_{\lambda b}(\lambda^*, T_S) \cos(\theta^*) \} d\lambda^* \sin(\theta^*) \} d\theta^* \right)$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

It is assumed that emission occurs from a grey body and thus the emissivity is not wavelength dependent. Using that

it follows for $F$ that

$$F(\theta) = \frac{2\pi \frac{\sigma T^4}{\pi}}{2\pi \frac{\sigma T^4}{\pi}} \left( \int_{\theta^*=0}^{\theta} \{ \varepsilon'(\theta^*, T) \cos(\theta^*) \sin(\theta^*) \} d\theta^* \right)$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$

$$\int_{\lambda^*=0}^{\infty} \varepsilon'_{\lambda b}(\lambda^*, T) d\lambda^* = \int_{\lambda^*=0}^{\infty} \varepsilon'(\lambda^*, T) d\lambda^* = \frac{e_b}{\pi} = \frac{\sigma T^4}{\pi}$$
Since diffuse emission is assumed, the emissivity is also independent from the polar angle and it follows that

\[
F(\theta) = \frac{2\pi a T^4}{\pi} \varepsilon(T) \int_{\theta=0}^{\theta} \cos(\theta^*) \sin(\theta^*) d\theta^*
\]

(3.20)

\[
= \frac{\int_{\theta=0}^{\theta} \cos(\theta^*) \sin(\theta^*) d\theta^*}{\int_{\theta=0}^{\theta} \cos(\theta^*) \sin(\theta^*) d\theta^*}
\]

(3.21)

\[
= \frac{-\frac{1}{2} \cos^2(\theta^*)|_{\theta}}{-\frac{1}{2} \cos^2(\theta^*)|_{\theta}} = -\cos^2(\theta) + \cos^2(0) = \sin^2(\theta)
\]

(3.22)

In order to be able to draw from that distribution, the inverse cumulative density function must be found.

\[
F^{-1}(U) = \arcsin \left( \sqrt{U} \right)
\]

(3.23)

If \(U\) denotes a realisation of the uniform distribution in the interval \([0, 1)\) then \(F^{-1}(U) = \arcsin(\sqrt{U})\) is a realisation of the distribution that is required.

### 3.3.3 Ray Tracing

In principle, a ray must be traced until it hits a surface. In the present implementation, a ray is traced from the point it enters a cell to the point it leaves that cell using an efficient algorithm that is also applied in the context of particle tracking [20]. This is done until the ray hits a face of a cell, that belongs to a surface. At this point the ray is either reflected or absorbed (section 3.3.4). The distance travelled in each cell is computed. This is done to make it possible to add the effect of participating media easily, should this be required in the future.

**Participating media**

The change in intensity due to absorption \((di'_{\lambda,a})\) along a differential path \(dS\) between \(S\) and \(S + dS\) is given by [22]

\[
di'_{\lambda,a} = -a_{\lambda}(S, \lambda) \cdot i'_{\lambda}(S) \cdot dS
\]

(3.24)

The material property \(a_{\lambda}(S, \lambda)\) is called absorption coefficient and can be dependent on wavelength and spatial position.

If the absorption coefficient \(a_{\lambda}\) is assumed constant, the relative change of intensity of a ray traveling a distance \(S\) can be derived from equation 3.24 as

\[
\frac{i'_{\lambda}(S)}{i'_{\lambda}(S = 0)} = \exp \left( -a_{\lambda}S \right)
\]

(3.25)
Knowing the distance $S$ that a ray travels in a cell and the ray intensity when entering the cell $i'_\lambda(S = 0)$, the heat addition to that cell is computed as

$$i'_\lambda(S = 0) \cdot [1 - \exp(-a_\lambda S)]$$

(3.26)

and could be included as volumetric heat source in the simulation of the flow.

### 3.3.4 Absorption and Reflection

A ray incident on an opaque surface can be either reflected or absorbed. The probability of absorption is given by $\alpha$ and that of reflection by $\rho = 1 - \alpha$. Whether a ray is reflected or absorbed is decided by drawing a random number $U$ from a uniform distribution in the interval $[0, 1)$ and absorbing the ray if $U \leq \alpha$, reflecting it otherwise.

### 3.3.5 Parallelisation

If the fluid domain is partitioned for the use of multiple cores or processors, it will occur that a ray must be communicated from one core or processor to another. If a ray is incident on a surface that marks the limit between two regions, this will be detected and the ray is stored in the ray buffer. The point of incidence, the direction and the power that is carried by the ray is stored.

Once all rays in all regions have been traced until they have either left the domain, been absorbed, or must be communicated on all cores and processors, it is checked, whether rays must be communicated. Should this be the case, the number of rays to be communicated between cores and processors is exchanged, receive buffers are created and the rays are exchanged using MPI. Should two regions be on the same core, it suffices to copy the buffer.
Chapter 4

Fluid Structure Thermal Interaction

When solving the Navier Stokes equation in conventional fluid dynamics simulations, Von Neumann or Dirichlet boundary conditions are usually prescribed on the temperature in addition to the no-slip condition to model solid walls. Such boundary conditions can be obtained from engineering simplifications.

In certain cases such as turbine blade cooling or solar reactors it can be of interest to investigate the fluid structure thermal interaction, also called conjugate heat transfer. This means that the fluid domain, or a part of it, is delimited by a solid which is discretised and included in the simulation. In the solid domain, the heat transfer equation is solved.

This chapter describes the common approach for implementation of fluid structure thermal interaction. It introduces the idea behind a novel approach based on physical constraints and outlines how that approach can be coupled with radiative heat transfer.

4.1 Coupling Two Separate Solvers

The most common approach for simulation of fluid thermal interaction starts from two existing solvers, one solving the Navier Stokes equation and one the heat equation. Both solvers accept Dirichlet or Von Neumann boundary conditions for temperature. It makes sense to think of ways to couple two such solvers by exchange of boundary conditions.

Heselhaus provides an overview of efforts which aim at simulating conjugate heat transfer [21]. In his work a finite elements solver for the heat equation is coupled with a finite volume fluid solver. This is done by exchanging the boundary heat flux from the fluid solver to the solid solver and the wall temperature resulting from the simulation of the temperature profile in the solid back to the fluid solver (figure 4.1).
Results of steady simulations are presented in [21] that are closer to the measurements than those of uncoupled simulations. While such an approach benefits from the advantages of the two solvers it is coupling in the respective domains, the coupling, especially ensuring energy conservation at the interface poses a challenge. For a consistent solution it is expected that continuity of the temperature profile and of heat fluxes at the boundaries is fulfilled. Heselhaus comments that for a converged simulation this is indeed the case. It is not demonstrated that this holds for a general case. It cannot be excluded that convergence could not be reached or numerical instabilities could occur. Giles in [10] shows that for a one dimensional case of two solids in contact and using a finite volume approach, stability depends on the choice of which type of boundary condition is communicated from which material. Finally, if using such an approach for an unsteady simulation, local conservativeness and continuity of temperature and heat fluxes at the interface is not trivial to impose at each time step.

**Implementation in ANSYS**

The implementation of fluid structure interaction in ANSYS is described in the manual with the following words [3]:

In a two-way transfer FSI analysis, the CFD analysis results (forces, temperatures, heat flows, or heat transfer coefficients and near wall temperatures) at the fluid-structure interface are transferred to the mechanical model and applied as loads. Within the same analysis, the subsequently calculated displacements, temperatures, or heat flows at the fluid-structure interface are transferred back to the CFD analysis. Two-way transfer is appropriate when displacements and temperature differentials calculated in the Mechanical application are large enough to have a significant impact on the fluid analysis.

This seems to be a similar algorithm as described in [21] but full clarification is not possible.
4.2 Novel Approach

In the framework of this thesis, a novel approach is investigated. It is based on the ideas of using one solver only and enforcing physical constraints at the interface. Using one solver has the advantage that conservativeness can be ensured in a consistent manner since the gradients, and thus the heat fluxes, are computed with the same numerical method in the fluid and the solid. The novel approach requires that the temperature profile and heat fluxes are continuous across the interface at all times and thus reflects first principles of physics. In fact only one temperature at the interface is stored, such that continuity of temperature is intrinsically fulfilled. This interface temperature is computed by enforcing continuity of fluxes. Such an algorithm can be used without restriction for unsteady and steady simulations.

The finite volume idea is extended to interfaces. Figure 4.2 illustrates a control volume constructed around a discretised interface.

![Figure 4.2: Extension of finite volume idea to a fluid-solid interface](image)

The thickness $\Delta x$ of such a control volume tends towards zero. As a consequence, it has no volume and mass. This implies that no flux imbalance can exist since no energy can be stored. Furthermore the heat addition due to fluxes parallel to the interface vanishes. When formulating energy conservation for such a control volume, indeed, the continuity of heat fluxes normal to the interface is enforced. It follows that

$$0 = q_{\text{solid}} - q_{\text{fluid}} \quad (4.1)$$

The interface temperature and heat fluxes across the interface are computed in the same step in a consistent manner. Note that $q_{\text{fluid}}$ and $q_{\text{solid}}$ are both conductive fluxes. This is trivially the case since the faces of the infinitely thin control volume coincide with the interface, where the fluid velocity vanishes.

The detailed implementation is described in section 6.3.2.
Coupling with Radiation

When coupling with radiation, one is looking at an implicit problem. This is because the interface temperature depends on the incoming and outgoing radiative heat fluxes. At the same time the emitted radiation depends on the interface temperature. Figure 4.3 illustrates the heat fluxes relevant for a heat balance, when coupling with radiation.

![Figure 4.3: Illustration of balance of heat fluxes at an interface](image)

Iterative algorithms exist to solve such problems. In the current framework, this may be avoided. The reasoning is that the change of the interface temperature is limited by the conduction of heat away or towards the interface. In many engineering problems, the time scales that must be resolved in the fluid flow are orders of magnitude smaller than the time scales governing heat transfer, and thus the temporal change of the interface temperature. The characteristic timescale for fluid phenomena is determined by the speed of sound $a$ and a characteristic length scale $\Gamma_{flow}$

$$\tau_{flow} = \frac{\Gamma_{flow}}{a} \quad (4.2)$$

The characteristic timescale for heat conduction is defined by the thermal diffusivity and a characteristic length scale $\Gamma_{cond}$

$$\tau_{cond} = \frac{\Gamma_{cond}^2}{D} \quad (4.3)$$

Characteristic timescales are given for some materials in table 4.1. It becomes apparent that $\tau_{flow} \ll \tau_{cond}$ holds. This means effectively that if the flow is well resolved, the changes in interface temperature are better resolved by orders of magnitude. Under the assumption that the interface temperature does not change much between single time steps, iteration can be avoided. The net radiative heat flux at the previous time step is taken as source term at the interface

$$q_{source} = q_{rad,in}^{n-1} - q_{rad,out}^{n-1} \quad (4.4)$$
In that context, the superscript \( n \) or \( n - 1 \) indicates the time at which a quantity is considered \((t = n \cdot \Delta t)\).

With the radiative heat fluxes given as a source term at the interface, the determination of the interface temperature reduces from solving an implicit problem to the solution of an algebraic equation which is obtained by requiring conservation of energy.

\[
q_{\text{fluid}}^n - q_{\text{solid}}^n = q_{\text{source}}
\]  

(4.5)

The detailed implementation is described in section 6.3.3.

It is important to note that depending on the materials and conditions, this simplification may not be applicable. In all cases, this must be checked before running a simulation. It should hold that

\[
\frac{\tau_{\text{cond}}}{\tau_{\text{flow}}} = \frac{\Gamma_{\text{cond}}^2}{\Gamma_{\text{flow}}} \cdot \frac{a}{D} > 10 \ldots 100
\]  

(4.6)

Should this not be fulfilled, the interface temperature determination can be applied iteratively. However it is not certain whether this would converge and will greatly affect efficiency.

**Table 4.1:** Overview of relevant time scales for flow and conduction phenomena in different materials, values for air from perfect gas relations \((\gamma = 1.4, R = 287 \frac{J}{kgK})\)

<table>
<thead>
<tr>
<th>Material</th>
<th>(D) ([m^2/s])</th>
<th>(a) ([m/s])</th>
<th>length scale ([m])</th>
<th>time scale ([s])</th>
</tr>
</thead>
<tbody>
<tr>
<td>air at 298K</td>
<td>-</td>
<td>346</td>
<td>0.001</td>
<td>2.9 (\cdot) 10^{-6}</td>
</tr>
<tr>
<td>air at 1000K</td>
<td>-</td>
<td>634</td>
<td>0.001</td>
<td>1.6 (\cdot) 10^{-6}</td>
</tr>
<tr>
<td>aluminum [8]</td>
<td>2.3 (\cdot) 10^{-3}</td>
<td>-</td>
<td>0.001</td>
<td>4.3 (\cdot) 10^{-4}</td>
</tr>
<tr>
<td>iron [8]</td>
<td>2.1 (\cdot) 10^{-5}</td>
<td>-</td>
<td>0.001</td>
<td>4.8 (\cdot) 10^{-2}</td>
</tr>
<tr>
<td>gold [8]</td>
<td>1.2 (\cdot) 10^{-4}</td>
<td>-</td>
<td>0.001</td>
<td>8.3 (\cdot) 10^{-3}</td>
</tr>
</tbody>
</table>
Chapter 5

Preliminary Study in One Dimension

Theoretical stability considerations in three dimensional, unstructured grids are very hard to perform, if not impossible in many cases. In order to minimise the risk of possible instabilities, a preliminary study is conducted to assess the stability of the novel algorithm, that should be implemented in ROCFLU (section 4.2). To do so, it is applied to a one dimensional problem and a structured grid. This will be done in theory and verified with numerical experiments.

Note that there is no guarantee that an algorithm that is stable when using a structured grid will also be stable when using an unstructured grid. The same applies when the dimensionality is increased.

The reduction of dimensionality allows to find an analytical solution for the described model problem. Knowing the analytical solution, the accuracy of the method can be assessed in that simplified case.

The chapter is concluded by the decision whether the method should be implemented in ROCFLU.

5.1 Model Problem

When looking for a model problem it is essential for assessing the accuracy of the method that there exists an analytical solution. The task of finding an analytical solution for a problem where a fluid is in contact with a solid can be achieved only with strong restrictions. No solution that could be of use in the context of this prestudy has been found.

A simplification that does not substantially alter the nature of the problem is found by recognising that, trivially, in no case there can be a convective heat flux across a fluid-solid interface. The case of two solids of different properties in contact is a very similar problem. It can be seen as the limit of an infinitely viscous and incompressible fluid in contact with a solid.

In the following, the analytical solution for one dimensional, semi-infinite solids at different temperatures brought into contact is discussed as well as the related problem of heat conduction in a semi-infinite solid.
5.1.1 Semi-Infinite Solid

A problem connected to that of two solids in contact is that of a semi-infinite solid. The governing equation for heat transfer is [4]

\[
\int_V \left[ \rho c_v \frac{\partial T}{\partial t} + \vec{\nabla} \cdot \vec{q}_c \right] dV = 0
\] (5.1)

It is assumed that the product \( \rho c_v \), the volume specific heat capacity, is constant. \( \vec{q}_c \), denoting the conductive heat flux, can be expressed using Fourier’s first law

\[
\vec{q}_c = -\kappa \cdot \vec{\nabla} T
\] (5.2)

where \( \kappa \) is the thermal conductivity.

The integral in equation 5.1 can only vanish for arbitrary volumes if the integrand itself vanishes, thus by assuming that \( \kappa \) is constant it is found that

\[
\rho c_v \frac{\partial T}{\partial t} = \kappa \vec{\nabla}^2 T
\] (5.3)

The domain of the model problem is \( x \in (0, \infty) \). It is assumed that at time zero, the temperature is constant everywhere

\[
T(x, t = 0) = T_2
\] (5.4)

In the limit of \( x \to \infty \) the temperature does not change.

\[
T(x \to \infty, t) = T_2
\] (5.5)

At \( x = 0 \) a jump in temperature is imposed at \( t = 0 \) such that

\[
T(x = 0, t > 0) = T_0
\] (5.6)

The solution to this problem is given by [4]

\[
T(x, t) = T_0 + (T_2 - T_0) \cdot \text{erf} \left( \frac{x}{2\sqrt{Dt}} \right)
\] (5.7)

where the error function is defined as

\[
\text{erf}(\phi) = \frac{2}{\sqrt{\pi}} \int_0^\phi \exp(-\xi^2) d\xi
\] (5.8)

and \( D \) denotes the thermal diffusivity

\[
D = \frac{\kappa}{\rho \cdot c_v}
\] (5.9)

An illustration of the analytical solution for gold at 300 K and a temperature jump to 493.56 K is illustrated in figure 5.1 at different times.
The following properties are used for gold [8]

\[ \kappa = 310 \frac{W}{m \cdot K} \]  \hspace{1cm} (5.10)

\[ \rho = 1930 \frac{kg}{m^3} \]  \hspace{1cm} (5.11)

\[ c_v = 130 \frac{J}{kg \cdot K} \]  \hspace{1cm} (5.12)

![Figure 5.1: Analytical solution for a semi-infinite body with temperature jump prescribed at time \( t = 0 \) s at \( x = 0 \) m, illustrated at times \( t = 0.5, 1.5, 3, 5, 7.5, 10.5, 14 \) s, initial condition in red]

Given the analytical solution of equation 5.7, the heat flux at a location \( x \) and a given time \( t \) can be expressed as [26]

\[ q_c = -\kappa \cdot \nabla T = \frac{\kappa \cdot (T_0 - T_1)}{\sqrt{\pi Dt}} \cdot \exp \left( \frac{-x^2}{4Dt} \right) = \epsilon \cdot \frac{(T_0 - T_1)}{\sqrt{\pi t}} \cdot \exp \left( \frac{-x^2}{4Dt} \right) \]  \hspace{1cm} (5.13)

The heat flux is proportional to a material property called effusivity \( \epsilon = \sqrt{\kappa \cdot \rho \cdot c_v} \).
5.1.2 Two Materials in Contact

A possible cause for a temperature jump at \( x = 0 \) could be contact with another body at different temperature. It is not a priori clear that the contact temperature would be constant in time. If it were constant, however, the solution in the two domains would be given by equation 5.7 with the corresponding material properties and boundary temperatures. The material properties and initial temperatures for \( x < 0 \) are denoted by the superscript one, those for \( x > 0 \) by the superscript two.

\[
T(x,t) = \begin{cases} 
T_0 + (T_0 - T_1) \cdot \text{erf}\left(\frac{x}{2\sqrt{D_1} t}\right) & x < 0 \\
T_0 + (T_2 - T_0) \cdot \text{erf}\left(\frac{x}{2\sqrt{D_2} t}\right) & x > 0 
\end{cases}
\tag{5.14}
\]

A solution must fulfill two requirements, namely continuity of temperature profile across the interface and continuity of heat fluxes at the interface. Continuity of temperature is fulfilled. This can be seen from equation 5.14 and considering that \( \text{erf}(0) = 0 \).

For continuity of fluxes at the interface, according to equation 5.13, it is required that

\[
\epsilon_1 \cdot (T_0 - T_1) = \epsilon_2 (T_2 - T_0) 
\tag{5.15}
\]

Thus the interface temperature must be [26], [4]

\[
T_0 = \frac{\epsilon_1 \cdot T_1 + \epsilon_2 \cdot T_2}{\epsilon_1 + \epsilon_2} 
\tag{5.16}
\]

To summarise: it has been assumed that the interface temperature is constant in time. In that case the analytical solution in the two domains is known with the exception of a constant, the interface temperature. By determining the interface temperature in an appropriate manner, it can be shown that the resulting solution is physically valid at the interface. Thus it has been shown that this is indeed the analytical solution fulfilling the heat equation in both domains and the physical constraints at the interface.

Figure 5.2 illustrates the solution for the temperature profile and heat flux if iron \((x < 0)\) at 600 K is brought into contact with gold \((x > 0)\) at 300 K at different times. The following properties have been used for iron [8]

\[
\kappa = 74 \quad \frac{\text{W}}{\text{m} \cdot \text{K}} 
\tag{5.17}
\]

\[
\rho = 7900 \quad \frac{\text{kg}}{\text{m}^3} 
\tag{5.18}
\]

\[
c_v = 440 \quad \frac{\text{J}}{\text{kg} \cdot \text{K}} 
\tag{5.19}
\]

The interface temperature evaluates to \( T_0 = 493.56 \text{ K} \).
It is important to note that the temperature gradient can be discontinuous at the interface. This is a consequence of the discontinuity in material properties. The discontinuity in gradient is such that it compensates the discontinuity in material properties, in order to have a continuous heat flux across the interface (figure 5.2).

**Figure 5.2:** Analytical solution for the temperature profile and heat flux in two semi-infinite bodies at different temperatures and with different properties brought into contact at $t = 0$ (initial condition in red), illustrated at times $t = 0.5, 1.5, 3.5, 7.5, 10.5, 14$ s, time progresses in direction of arrows.
5.2 Numerical Method

In this section the numerical method used for the prestudy in one dimension is described. The heat equation in its integral form is [4]

$$\int_V \left[ \rho c_v \frac{\partial T}{\partial t} + \vec{\nabla} \cdot \vec{q}_c \right] dV = 0 \quad (5.20)$$

It is assumed that the product $\rho c_v$, the volume specific heat capacity, is constant in both domains. $q_c$, denoting the conductive heat flux, can be expressed using Fourier's first law

$$\vec{q}_c = -\kappa \cdot \vec{\nabla} T \quad (5.21)$$

Using the divergence theorem and assuming constant heat conductivity $\kappa$ in both domains, equation 5.20 can be rewritten

$$\int_V \rho c_v \frac{\partial T}{\partial t} dV = \int_S \kappa \vec{\nabla} T \cdot \vec{n} dS \quad (5.22)$$

The notation used for time and space is

$$x_j = \begin{cases} j \cdot \Delta x & \text{if } j \leq 0 \\ j \cdot \Delta x & \text{if } j > 0 \end{cases} \quad (5.23)$$

$$t^n = n \cdot \Delta t \quad (5.24)$$

Figure 5.3 illustrates the spatial discretisation of the one dimensional domain. The domain where $x < 0$ is denoted as the minus domain, that where $x > 0$ as the plus domain. A cell centered scheme is used.

Note that the index notation is unusual for a cell centered scheme but this is done on purpose to make it possible to use the same notation when applying a Godunov Ryabenkii analysis to that method (section 5.4).

Figure 5.3: Spatial discretisation of two semi-infinite bodies brought into contact at $x = 0$ in one dimension, cell centered approach

29
5.2.1 Spatial Discretisation

The spatial discretisation of equation 5.22 in one dimension is

\[
RHS = -\kappa \cdot \frac{\delta T}{\delta x} \bigg|_{i-\frac{1}{2}} - \left[ -\kappa \cdot \frac{\delta T}{\delta x} \bigg|_{i+\frac{1}{2}} \right]
\]  

(5.25)

Using second order centered finite difference stencils this translates to the following equations

**Interior cells** \((j > \frac{1}{2})\)

\[
RHS_j = -\kappa_+ \cdot \frac{T_j - T_{j-1}}{\Delta x_+} - \left( -\kappa_+ \cdot \frac{T_{j+1} - T_j}{\Delta x_+} \right)
\]

(5.26)

leads to

\[
RHS_j = \frac{\kappa_+}{\Delta x_+} \cdot (T_{j+1} - 2 \cdot T_j + T_{j-1})
\]

(5.27)

**Interior cells** \((j < -\frac{1}{2})\)

\[
RHS_j = \frac{\kappa_-}{\Delta x_-} \cdot (T_{j+1} - 2 \cdot T_j + T_{j-1})
\]

(5.28)

**Cells adjacent to boundary** \((|j| = \frac{1}{2})\)

\[
RHS_{\frac{1}{2}} = -\kappa_+ \cdot \frac{T_{\frac{1}{2}} - T_0}{\Delta x_+} - \left( -\kappa_+ \cdot \frac{T_{\frac{3}{2}} - T_{\frac{1}{2}}}{\Delta x_+} \right)
\]

(5.29)

leads to

\[
RHS_{\frac{1}{2}} = \frac{\kappa_+}{\Delta x_+} \cdot \left(T_{\frac{3}{2}} - 3 \cdot T_{\frac{1}{2}} + 2 \cdot T_0\right)
\]

(5.30)

Furthermore

\[
RHS_{-\frac{1}{2}} = \frac{\kappa_-}{\Delta x_-} \cdot \left(2 \cdot T_0 - 3 \cdot T_{-\frac{1}{2}} + T_{-\frac{3}{2}}\right)
\]

(5.31)
Boundary temperature \((j = 0)\) Requiring that the heat flux across the interface is continuous leads to

\[
\kappa_- \cdot \lim_{x \to 0^-} \left( \frac{\partial T}{\partial x} \right) = \kappa_+ \cdot \lim_{x \to 0^+} \left( \frac{\partial T}{\partial x} \right)
\]  

(5.32)

Since a discontinuity in temperature gradient can exist at the interface the gradients in each domain must be approximated using values in the same domain only. Using a one sided, first order finite difference stencil leads to

\[
\kappa_- \cdot \frac{T_0 - T_{-\frac{1}{2}}}{\Delta x_-} = \kappa_+ \cdot \frac{T_{\frac{1}{2}} - T_0}{\Delta x_+}
\]  

(5.33)

and solving for the interface temperature gives

\[
T_0 = \frac{\kappa_- \cdot T_{-\frac{1}{2}} + \kappa_+ \cdot T_{\frac{1}{2}}}{\kappa_- / \Delta x_- + \kappa_+ / \Delta x_+}
\]  

(5.34)

5.2.2 Generalisation as Diffusion Problem with Variable Diffusivity

Instead of two materials in contact, one could assume another point of view and interpret the problem of the two interface adjacent cells as cells in one material with spatially varying heat conductivity. A similar type of problem is encountered in the field of porous media and discussed in the context of multi point flux approximation methods [1].

In the case of one material, the concept of interface temperature is not relevant. However, equality of fluxes at faces of a control volume must still hold. The spatial discretisation can indeed be rewritten by eliminating the interface temperature from the expression for the flux across the interface. This is done by inserting the expression for the interface temperature (equation 5.34) into the expression for the flux across the interface (left or right hand side of equation 5.33). This leads to

\[
\text{flux}_0 = -\frac{2}{\Delta x_- / \kappa_- + \Delta x_+ / \kappa_+} \cdot \left( T_{\frac{1}{2}} - T_{-\frac{1}{2}} \right)
\]  

(5.35)

This could be applied to all faces that have adjacent cells with different heat conductivity or grid spacing. Note that if two cells that share a face have the same properties, the expected result is obtained by inserting \(\kappa_- = \kappa_+ = \kappa\) and \(\Delta x_- = \Delta x_+ = \Delta x\) into equation 5.35

\[
\text{flux}_0 = -\kappa \cdot \frac{T_{\frac{1}{2}} - T_{-\frac{1}{2}}}{\Delta x}
\]  

(5.36)

5.2.3 Time Integration

The forward Euler method is used

\[
T_j^{n+1} = T_j^n + \Delta t \cdot \frac{RHS_j^n}{\Delta x \rho c_v}
\]  

(5.37)
5.2.4 Full Discretisation

The full discretisation is summarised. The following notation is used

\[ r = \frac{k_+}{\Delta x_+ + k_-} \]

\[ d_\pm = \frac{D_\pm \cdot \Delta t}{\Delta x^2} \]

\[ j < -\frac{1}{2} \]

\[ T_j^{n+1} = T_j^n + d_- \cdot (T_{j+1}^n - 2 \cdot T_j^n + T_{j-1}^n) \] (5.40)

\[ j = -\frac{1}{2} \]

\[ T_{-\frac{1}{2}}^{n+1} = T_{-\frac{1}{2}}^n + d_- \cdot \left(2 \cdot T_0^n - 3 \cdot T_{-\frac{1}{2}}^n + T_{-\frac{1}{2}}^n\right) \] (5.41)

\[ j = \frac{1}{2} \]

\[ T_{\frac{1}{2}}^{n+1} = T_{\frac{1}{2}}^n + d_+ \cdot \left(T_{\frac{3}{2}}^n - 3 \cdot T_{\frac{1}{2}}^n + 2 \cdot T_0^n\right) \] (5.42)

\[ j > \frac{1}{2} \]

\[ T_j^{n+1} = T_j^n + d_+ \cdot (T_{j+1}^n - 2 \cdot T_j^n + T_{j-1}^n) \] (5.43)

\[ j = 0 \]

\[ T_0^{n+1} = r \cdot T_{\frac{1}{2}}^{n+1} + (1 - r) \cdot T_{-\frac{1}{2}}^{n+1} \] (5.44)

Note that the interface temperature at a given time is computed from the temperature values in the domains at that same time. It is possible to rewrite the equation to obtain an expression for the interface temperature at a given time \( (t_n^{n+1}) \) as a function of the domain and interface temperatures at the previous time step \( (t_n^n) \) by inserting equations 5.41 and 5.42 into equation 5.44

\[ T_0^{n+1} = (1 - r) \cdot d_- \cdot T_{-\frac{1}{2}}^n + r \cdot d_+ \cdot T_{\frac{1}{2}}^n + \]

\[ (1 - r) \cdot (1 - 3d_-) \cdot T_{-\frac{1}{2}}^n + r \cdot (1 - 3d_+)T_{\frac{1}{2}}^n + \]

\[ 2 \cdot [(1 - r) \cdot d_- + r \cdot d_+] \cdot T_0^n \]

5.3 Von Neumann Analysis

The Von Neumann analysis is a method for predicting the stability of fully discretised numerical methods. It assumes periodic boundaries or an infinite domain and introduces the Ansatz [24]

\[ T_i^n = T(x_i, t^n) = \sum_k \hat{T}_k^n \cdot e^{ikx_i} \] (5.46)
Due to linearity, it suffices to consider one mode. Inserting the Von Neumann Ansatz 5.46 into the fully discrete approximation used in the domain (equation 5.40 or 5.43) repeated here

\[ T_{i+1}^n = T_i^n + d \cdot (T_{i-1}^n - 2T_i^n + T_{i+1}^n) \]  

leads to

\[ \frac{T_{k+1}^n}{T_k^n} = 1 + 2d \left[ \cos(k\Delta x) - 1 \right] \]  

For stability it must hold that \[ \left| \frac{T_{k+1}^n}{T_k^n} \right| \leq 1 \]. This leads to

\[ |1 - 4d| \leq 1 \]  

and finally

\[ d \leq 0.5 \]  

In the determination of this stability limit, the interface was not taken into account. It cannot be excluded that the presence of a discontinuity in material properties, an interface, introduces additional instabilities. A Von Neumann analysis is not capable of finding such instabilities. To do so, a Godunov Ryabenkii analysis is carried out.

### 5.4 Godunov Ryabenkii Analysis

The Godunov Ryabenkii analysis is a mathematical analysis developed by two Russian mathematicians in the sixties and first published in [11] and [12]. A good overview is given in [13]. The Godunov Ryabenkii analysis assumes that a linear numerical scheme can be written as

\[ \phi^{n+1} = Q \phi^n \]  

where the operator \( Q \) is applied to the values \( \phi \). \( Q \) can be thought of as a matrix and \( \phi \) as a vector. If a subscript \( h \) is used (\( Q_h \)), a constant grid spacing \( h \) is assumed. Godunov and Ryabenkii recognised the shortcomings of determining the stability by an analysis of the eigenvalues of \( Q \). They introduced the concept of a “family of operators”. The following definition is given [13].

**Definition 5.4.1.** A point \( \lambda \) is a spectral point of \( \{Q_h\} \) if for every \( \epsilon > 0 \) and \( h_0 > 0 \) we can give a number \( h, h < h_0 \), such that the inequality \( \|Q_h\phi - \lambda\phi\| < \epsilon \|\phi\\| \) has a solution \( \phi \). We call the aggregate of all spectral points the spectrum of \( \{Q_h\} \).

With this definition, the following theorem holds

**Theorem 5.4.1.** For the stability of a problem of the form

\[ \phi^{n+1} = Q_h \phi^n \quad n = 0, 1, \ldots \]  

it is necessary that the spectrum of \( \{Q_h\} \) should lie in the unit disc.
This powerful theorem is not easy to understand in its most general form. More references include [33] and [14] and [31]. The application to a fully discretised numerical scheme is demonstrated in [13], [28], and in applications closer to engineering numerics in [10], [29], and [30]. In the latter three references the analysis leads to correct stability limits, even though, strictly speaking, the Godunov Ryabenkii criterion is not sufficient for stability. Following [10], [29] and [30] the following Ansatz is introduced to perform the Godunov Ryabenkii Analysis

\[ \phi_j^n = z^n \cdot k^j \]  (5.53)

It is assumed that \( |k| < 1 \) holds, such that the solution is stable in space. The stability limit is found by inserting \( z = \pm 1 \).

In the following an exemplary application of the Godunov Ryabenkii criterion is shown to illustrate how the method is applied. The Godunov Ryabenkii analysis is then applied to the model problem fully described by the equations given in section 5.2.4. Finally, the result of the stability analysis is verified by means of numerical experiments.

### 5.4.1 Exemplary stability determination

An exemplary determination of the stability limit is presented in order to demonstrate the power of the Godunov Ryabenkii analysis. A one dimensional heat conduction problem in a domain of length \( l \) is considered. The domain, discretised using a cell centered approach, is illustrated in figure 5.4.

![Figure 5.4: Illustration of spatial discretisation of model problem for Godunov Ryabenkii analysis](image)

The discretisation is such that the following holds

\[ x_j = j \cdot \Delta x \]  (5.54)

\[ t^n = n \cdot \Delta t \]  (5.55)
The fact that cell centers have not integer numbers as indices may seem unusual. It is done in order to be able to use the same notation for the Godunov Ryabenkii analysis.

Initially, the domain is at constant temperature

\[ T(x, t = 0) = T_1 \]  

(5.56)

At \( t = 0 \) a jump in temperature is prescribed at the domain boundary \( x = 0 \)

\[ T(x = 0, t > 0) = T_0 \] \hspace{1cm} \[ T(x = l, t) = T_1 \]  

(5.57) \hspace{1cm} (5.58)

This problem is similar in character to that encountered when computing the interface temperature.

As in equation 5.27, the flux balance for cells inside the domain is computed as

\[ RHS^n_j = \kappa \cdot \frac{T^n_{j+1} - 2T^n_j + T^n_{j-1}}{\Delta x} \]  

(5.59)

When computing the flux balance for the cell adjacent to the left wall \( (j = \frac{1}{2}) \), a one sided second order stencil for flux computation at the wall is used. This leads to

\[ RHS^{\frac{1}{2}}_n = -\kappa \cdot \frac{1}{\Delta x} \cdot \left( \frac{\Delta x}{3} \cdot \left( -8T^n_0 + 9T^n_{\frac{1}{2}} - T^n_{\frac{3}{2}} \right) - (T^n_{\frac{3}{2}} - T^n_{\frac{1}{2}}) \right) \]  

(5.60)

For time integration, the explicit Euler method is used

\[ T^{n+1}_j = T^n_j + \Delta t \cdot \frac{RHS^n_j}{\Delta x \rho c_v} \]  

(5.61)

Using \( d = \frac{D \Delta t}{\Delta x^2} \) the full discretisation is summarised as

\( (j > \frac{1}{2}) \)

\[ T^{n+1}_j = T^n_j + d \cdot \left( T^n_{j+1} + T^n_{j-1} - 2T^n_j \right) \]  

(5.62)

\( (j = \frac{1}{2}) \)

\[ T^{n+1}_{\frac{1}{2}} = T^n_{\frac{1}{2}} + d \cdot \left( \frac{4}{3} T^n_{\frac{3}{2}} - 4T^n_{\frac{1}{2}} + \frac{8}{3} T^n_0 \right) \]  

(5.63)

The Godunov Ryabenkii analysis uses a modal Ansatz and introduces two unknowns \( z, k \)

\[ T^n_j = z^n \cdot k^j \]  

(5.64)

For ease of the analysis it is assumed without loss of generality that \( T_0 = 0 \). Inserting the Godunov Ryabenkii Ansatz 5.64 into the full discretisation (5.62 and 5.63) leads to

\[ z = 1 + d \cdot (k - 2 + k^{-1}) \]  

(5.65)

\[ z = 1 + d \left( \frac{4}{3} k - 4 \right) \]  

(5.66)
Equation 5.65 is a quadratic equation for $k$. It can be solved to give

$$k = 1 - \frac{1 - z}{2d} \pm \frac{1}{2d} \sqrt{(1 - z)^2 - 4d(1 - z)}$$ \hspace{0.5cm} (5.67)

When investigating temporal stability, it must hold that $|k| < 1$ such that $k^j$ does not diverge as $j \to \infty$. Therefore the negative sign in front of the square root must be chosen. This leads to [10], [30]

$$k(d, z) = 1 - \frac{1 - z}{2d} \cdot \left(1 - \sqrt{1 - \frac{4d}{1 - z}}\right)$$ \hspace{0.5cm} (5.68)

Inserting equation 5.68 into equation 5.66 and solving for $d$ leads to

$$d = \frac{\pm \sqrt{3z^2 - 6z + 3}}{8} \hspace{0.5cm} (5.69)$$

$$d \neq 0 \hspace{0.5cm} (5.70)$$

For finding the stability limit, $z = \pm 1$ is inserted. $z = 1$ leads to $d = 0$, which is excluded as solution. Inserting $z = -1$ leads to

$$d_{\text{crit}} = \frac{\sqrt{3}}{4}$$ \hspace{0.5cm} (5.71)

In order to verify that this is indeed the stability limit, the numerical algorithm (equation 5.62 and 5.63) is implemented. The following setup is used

$$\kappa = 16 \text{ W m}^{-1} \text{ K} \hspace{0.5cm} (5.72)$$

$$\rho = 490 \text{ kg m}^{-3} \hspace{0.5cm} (5.73)$$

$$c_v = 7500 \text{ J kg}^{-1} \text{ K} \hspace{0.5cm} (5.74)$$

$$l = 1 \text{ m} \hspace{0.5cm} (5.75)$$

$$\Delta x = 0.01 \text{ m} \hspace{0.5cm} (5.76)$$

$$T_0 = 600 \text{ K} \hspace{0.5cm} (5.77)$$

$$T_1 = 300 \text{ K} \hspace{0.5cm} (5.78)$$

This leads to a predicted critical time step of $\Delta t_{\text{crit}} = 9.94576 \text{ s}$. Figure 5.5 shows results at different times obtained with a diffusion number of $d = \frac{\sqrt{3}}{4} - 10^{-6}$. Some spurious oscillations are visible after $5 \cdot 10^6$ steps which disappear later. The numerical solution converges to the analytical solution expected as $t \to \infty$, a linear temperature profile between the prescribed boundary values.
The results obtained when using a diffusion number slightly above the predicted stability limit \( d = \frac{\sqrt{3}}{4} + 10^{-6} \) are shown in figure 5.6. Huge oscillations that clearly originate at the boundary are observed that do not at all disappear, in the contrary spread into the domain, as time progresses. The solution is unstable indeed. It was predicted that the limiting case is at \( z = -1 \). For \( z = -1 \) and \( d = \frac{\sqrt{3}}{4} \), \( k \) evaluates to \( k = -0.464 \) according to equation 5.68. As a consequence, due to the Godunov Ryabenkii Ansatz, the instabilities are predicted to be of oscillating character in space. This is the case (figure 5.6). The solution also oscillates in time. This is illustrated in figure 5.7 where the solution at \( x = 0.05 \) m is illustrated as a function of time.

Figure 5.5: Numerical solution for the model problem at different times, using a diffusion number of \( d = \frac{\sqrt{3}}{4} - 10^{-6} \)

Figure 5.6: Numerical solution for the model problem at different times, using a diffusion number of \( d = \frac{\sqrt{3}}{4} + 10^{-6} \)
Figure 5.7: Numerical solution for the model problem at $x = 0.05$ m, using a diffusion number of $d = \frac{\sqrt{3}}{4} \times 10^{-6}$

5.4.2 Stability limit for Model Problem

The Godunov Ryabenkii analysis is now applied to the algorithm described in section 5.2.4. It is undertaken under the assumption that the algorithm is stable in the absence of boundaries and thus

$$\max(d_-, d_+ \leq 0.5$$

(5.79)

It is investigated, whether the interface introduces instabilities that would limit the time step more.

There are two domains in the case of an interface. This is taken into account when formulating the Godunov Ryabenkii Ansatz (see also [10], [29], [30])

$$T^m_j = \begin{cases} 
z^n \cdot k_+^j & \text{if } j \leq 0 \\
z^n \cdot k_-^j & \text{if } j > 0 \end{cases}$$

(5.80)

Inserting the Ansatz from equation 5.80 into equations 5.40 and 5.43 gives

$$z = 1 + d_+ (k_+ - 2 + k_+^{-1})$$

(5.81)

$$z = 1 + d_- (k_- - 2 + k_-^{-1})$$

(5.82)

From the form of equations 5.81 and 5.82 one can infer that the solution for $k_+$ will be identical to that of $k_-^{-1}$. The equations are quadratic in $k_+$ and $k_-^{-1}$ respectively and lead to two solutions.

$$k_+(d_+, z) = 1 - \frac{1 - z}{2d_+^2} \pm \frac{1}{2d_+} \cdot \sqrt{(1 - z)^2 - 4d_+(1 - z)}$$

(5.83)

$$k_-^{-1}(d_-, z) = 1 - \frac{1 - z}{2d_-^2} \pm \frac{1}{2d_-} \cdot \sqrt{(1 - z)^2 - 4d_-(1 - z)}$$

(5.84)
In equations 5.83 and 5.84 the plus or minus sign could be used in front of the square root. When investigating temporal stability, it is assumed that no unstable modes can exist, thus it must hold that \(|k_+| \leq 1\) and \(|k_-| \leq 1\). The positive sign in equation 5.83 and equation 5.84 must be used [10], [29], [30].

\[
k_+(d_+, z) = 1 - \frac{1 - z}{2d_+} \cdot \left(1 - \sqrt{1 - \frac{4d_+}{1 - z}}\right) \quad (5.85)
\]

\[
k_-^{-1}(d_-, z) = 1 - \frac{1 - z}{2d_-} \cdot \left(1 - \sqrt{1 - \frac{4d_-}{1 - z}}\right) \quad (5.86)
\]

Note that the above expressions can be misleading in that sense that they may seem to diverge at first sight for \(z = 1\) whereas they do not. In general, \(k_-\) and \(k_+\) are complex numbers. Figures 5.8 and 5.9 illustrate the real and imaginary part of \(k\) as a function of the diffusion number for different values of \(z\).

\[k_+(d_+, z) = k_-^{-1}(d_-, z) = k(d, z)\]

**Figure 5.8:** Real part of \(k\) (constant in Godunov Ryabenkii Ansatz 5.80) as a function of diffusion number for different values of \(z\) (constant in Godunov Ryabenkii Ansatz 5.80)
Figure 5.9: Imaginary part of $k$ (constant in Godunov Ryabenkii Ansatz 5.80) as a function of diffusion number for different values of $z$ (constant in Godunov Ryabenkii Ansatz 5.80)

When investigating temporal stability, one is interested in the limiting case where $z = \pm 1$. In both limiting cases, the imaginary parts of $k_-$ and $k_+$ vanish.

\[
k_+(d_+, z = 1) = 1 \quad k_-^{-1}(d_-, z = 1) = 1 \quad (5.87)
\]
\[
k_+(d_+, z = -1) = 1 - \frac{1}{d_+} \cdot \left(1 - \sqrt{1 - 2d_+}\right) < 0 \quad (5.89)
\]
\[
k_-^{-1}(d_-, z = -1) = 1 - \frac{1}{d_-} \cdot \left(1 - \sqrt{1 - 2d_-}\right) < 0 \quad (5.90)
\]

For $z = -1$ the values of $k$ are real and negative.

5.4.2.1 Case $d_- = d_+$

The case where $d_- = d_+$ corresponds to that of one domain with constant grid spacing and same properties. No interface exists, since there is no discontinuity in the diffusion number. As outlined in section 5.2.2 the numerical method in this case corresponds to that which would be used for one domain with constant properties. The stability limit can be found using a Von Neumann analysis (section 5.3).
5.4.2.2 Case \( d_- \neq d_+ \)

In a general case, if the two domains are different and the spatial discretisation is not chosen such that \( d_- = d_+ \), the diffusion numbers are different in both domains. In such a case where \( d_- \neq d_+ \) a Godunov Ryabenkii analysis is carried out to investigate stability. The Ansatz (5.80) is inserted into equation (5.45) and simplified to give

\[
\begin{align*}
z = & (1 - r) \cdot d_- k_-^{\frac{3}{2}} + r d_+ k_+^{\frac{3}{2}} \\
  & + (1 - r)(1 - 3d_- )k_-^{\frac{1}{2}} + r(1 - 3d_+ )k_+^{\frac{1}{2}} + \\
  & 2[(1 - r)d_- + r d_+] \\
\end{align*}
\]

Under the assumption that \( d_- \neq d_+ \) equation (5.91) can be solved for \( r \)

\[
\begin{align*}
  r = & \frac{d_- k_-^{\frac{3}{2}} + (1 - 3d_- )k_-^{\frac{1}{2}} + 2d_- - z}{d_- k_-^{\frac{3}{2}} - d_+ k_+^{\frac{3}{2}} + (1 - 3d_- )k_-^{\frac{1}{2}} - (1 - 3d_+ )k_+^{\frac{1}{2}} - 2(d_+ - d_- )} \\
\end{align*}
\]

A stability limit is usually formulated as a relation between the diffusion number and the problem parameters, in this case \( r \). Since in this case there are two diffusion numbers, it is more elegant to solve for the parameter \( r \) as a function of the two diffusion numbers. Note that \( r \) is restricted between \( 0 < r < 1 \) (see equation (5.38)).

**Case \( z = 1 \)** For \( z = 1 \) it follows that (5.87 and 5.88)

\[
k_+ = k_- = 1
\]

Inserting these values into equation (5.92) leads to a division of two expressions that evaluate to zero. In order to obtain the correct solution, one must insert equations (5.85 and 5.86) into equation (5.92) and take the limit of \( z \to 1 \). This done with the software Mathematica 9 and leads to

\[
\begin{align*}
  r = & \sqrt{d_+} \\
  & \sqrt{d_+ - d_- } \\
\end{align*}
\]

If a stability limit exists, a combination of \((d_-, d_+)\) must exist such that \( r \) is in the possible range of \( 0 < r < 1 \). Figure 5.10 shows \( r \) as a function of the two diffusion number \( d_- \) and \( d_+ \). It is not clearly visible, whether such a combination of diffusion numbers exist or not. For further investigation a new system of coordinates \( x \) and \( y \) is introduced (figure 5.10).
Figure 5.10: Illustration of $r$ (equation 5.38) as a function of the two diffusion numbers $d_-$ and $d_+$ for the case of $z = 1$ (equation 5.80)

Expressing the values of $r$ along the diagonal in the $(d_-, d_+)$ domain

$$d_+ = m - d_- \quad 0 < m < 1$$ (5.95)

and using the coordinates

$$x = \sqrt{2} \cdot d_+ - \frac{m}{\sqrt{2}}$$ (5.96)

$$y = \frac{m}{\sqrt{2}}$$ (5.97)

gives the following expression for $r$

$$r(x, y) = \frac{\sqrt{\sqrt{2}(y - x)}}{\sqrt{\sqrt{2}(y - x) - \sqrt{2}(y + x)}}$$ (5.98)

This expression for $r$ is illustrated as a function of the $x$-coordinate at different $y$ values in figure 5.11
Figure 5.11: Values of \( r \) as a function of the \( x \)-coordinate according to equation 5.98 with \( z = 1 \) for different positions on the \( y \)-coordinate, with envelopes, possible range for values of \( r \) highlighted in green

The envelope curves have been found by evaluating equation 5.98 at the boundaries where

\[
x = \begin{cases} 
\pm y & m \leq 0.5 \\
\pm (\frac{\sqrt{2}}{2} - y) & m > 0.5
\end{cases}
\] (5.99)

It is clearly visible that no solution for \( r \) crosses the possible range of \( r \) values. Note that the curves seem to touch the limit. This is the case for either \( d_- = 0 \) or \( d_+ = 0 \). Both cases are excluded as the diffusion number cannot be zero. It is concluded that no spatially stable mode with \( z = 1 \) can exist.

Case \( z = -1 \) For \( z = -1 \) the expression for \( r \) (equation 5.92) is complex since the values of \( k_-^{-1} \) and \( k_+ \) are strictly negative (equations 5.89 and 5.90). Figures 5.12 and 5.13 illustrate the imaginary and real part of \( r \) according to equation 5.92 where equations 5.89 and 5.90 were inserted for \( k_- \) and \( k_+ \).

\( r \) is a real quantity. The zero contour illustrates that there exist combinations of \( d_- \) and \( d_+ \) such that the imaginary part vanishes. It must be further investigated whether the real part of \( r \) can assume values in the possible range.
As in the case of $z = 1$, a new coordinate system that is $45^\circ$ rotated in counterclockwise direction is introduced and the values of $r$ are plotted as a function of the $x$-coordinate for different values of $y$. In this case the boundary values ($d_- = 0$ and $d_+ = 0.5$) do not form envelopes but define the starting points of the curves for $r$ that increase as $x$ approaches zero and diverge for $x = 0$. It is clearly visible in figure 5.14, that no solution exists for the possible range of $r$ values highlighted in green. Note that this is shown in general, whereas it would be enough if that was the case for the $d_-$ and $d_+$ combinations where the imaginary part vanishes.

Also in the case of $z = -1$, no stability limit can be found.
Figure 5.14: Values of $r$ as a function of the $x$-coordinate according to equation 5.98 with $z = -1$ for different positions on the $y$-coordinate, possible range for values of $r$ highlighted in green

5.4.2.3 Summary

It has been shown that inserting the Godunov Ryabenkii Ansatz (equation 5.80) into the method for interface treatment (section 5.2.4), no stability limit can be found. Based on the results of the Godunov Ryabenkii analysis it is expected that the interface treatment as described in section 5.2.4 does not introduce any instabilities being more restrictive than the stability limit given by the Von Neumann criterion (equation 5.50).

5.4.3 Verification

In order to verify the result of the Godunov Ryabenkii analysis, numerical experiments are conducted. A case with a temperature difference of 1600K is run using 512 points, 256 in each domain. The relevant parameters ($d_-, d_+, r$) are varied systematically between

$$0 < d_-, d_+ < 0.6$$  \hspace{1cm} (5.100)

$$0 < r < 1$$  \hspace{1cm} (5.101)
The diffusion numbers are discretised into hundred points, the interval for \( r \) into two hundred. For each combination of parameters, a simulation is run until the outer boundary adjacent cells (not interface adjacent) have changed by two Kelvin or a maximum of \( 10^6 \) steps is reached.

The maximum and minimum temperatures at the end of each simulation are written out. It is checked whether they are within the interval defined by the initial values. An unstable solution is characterised by the occurrence of temperatures that are above the initial maximum or below the initial minimum temperature. Based on that criterion it can be confirmed that indeed no unstable solution can be observed where both diffusion numbers are below 0.5.

### 5.5 Accuracy

When investigating accuracy, excellent results have been obtained. A representative case where iron is brought into contact with gold is illustrated in figure 5.15. The spatial discretisation is as in figure 5.3. The setup is the following

\[
\begin{align*}
\max (d_-, d_+) &= 0.1 \\
\ell_- &= 1 \text{m} \\
\Delta x_- &= 0.04 \text{ m} \\
\ell_+ &= 0.25 \text{ m} \\
\Delta x_+ &= 0.01 \text{ m} \\
T(x < 0, t = 0) &= 300 \text{ K} \\
T(x > 0, t = 0) &= 1000 \text{ K}
\end{align*}
\]

The following material properties are used for gold [8]

\[
\begin{align*}
\kappa &= 310 \frac{\text{W}}{\text{m} \cdot \text{K}} \\
\rho &= 1930 \frac{\text{kg}}{\text{m}^3} \\
c_v &= 130 \frac{\text{J}}{\text{kg} \cdot \text{K}}
\end{align*}
\]
and iron [8]

\[
\kappa = 74 \frac{W}{m \cdot K} \tag{5.113}
\]

\[
\rho = 7900 \frac{kg}{m^3} \tag{5.114}
\]

\[
c_v = 440 \frac{J}{kg \cdot K} \tag{5.115}
\]

The recomputed time step is 1.2953 \cdot 10^{-3} s. This means for the diffusion numbers that

\[
d_- = 0.1 \tag{5.117}
\]
\[
d_+ = 0.02757 \tag{5.118}
\]

The numerical result after 15444 steps (20 s) together with the initial condition is illustrated in figure 5.15.

The analytical solution for the interface temperature is \( T_0 = 751.6451 \) K for all times (section 5.1.2). The relative error in interface temperature as a function of time is illustrated in figure 5.16. The error decreases quickly and the interface temperature converges to the analytical value. This is representative and has always been the case in numerical simulations that were performed.

The error distribution in space, normalised with the analytical value for the interface temperature, is illustrated in figure 5.17 for different times. It decreases and diffuses as time progresses.
Figure 5.15: Gold brought into contact with iron, initial condition and numerical solution after 20 s (15444 time steps)

Figure 5.16: Relative error in interface temperatures in dependence of time
5.6 Conclusion

In conclusion it can be said that this prestudy in one dimension has shown that the novel algorithm works well in this simplified setup. It has been shown that the interface treatment does not introduce instabilities that would be more restrictive than those that apply within the domain. Finally it has been shown that the method leads to accurate results. While this is not a guarantee that these conclusions apply to three dimensions for unstructured grids, there are no indications of problems on any level. Based on this study, it will be proceeded with the implementation in ROCFLU.
Chapter 6

Implementation of Fluid Structure Thermal Interaction in ROCFLU

In this chapter it is described how the heat equation is solved in the solid and how the coupling at the fluid-structure interface is implemented. This is done using a novel algorithm described in section 4.2 and has been implemented in the framework of this thesis.

6.1 Preprocessing

The preprocessing is adapted to allow for two different materials. The different materials are distinguished by the domain index. The domain index is an information that must be contained in the grid and read in by ROCFLU. This is implemented for grids in the CENTAUR in-house format. Negative indices indicate that the domain is a solid, positive indices stand for fluid.

When partitioning, it is made sure that every region consists of one material only such that in each cell of any region the same numerical algorithm can be applied. The domain index of every region is written into a file called [casename].reg2dom. Note that virtual cells may be of different material than the region, if there is an interface. For that reason, for every region the domain indices of the cells are written into a file called [casename].cell2dom_[region number].

Limitations

The partitioning is limited to two different materials at the moment. In principle it could handle more materials, but the case of different material limits crossing has not been thought of.

The interface patch is reconstructed from the knowledge of domain indices. An interface patch may intersect a regular patch, but the case where an interface patch is intersected by a regular patch has not been considered in the implementation. The partitioning algorithm does run in that case, but the simulation leads to wrong results.
6.2 Heat Equation

The heat equation that is solved in the solid in its integral form is [4]

\[
\int_V \left[ \rho c_v \frac{\partial T}{\partial t} + \vec{\nabla} \cdot \vec{q}_c \right] dV = 0 \tag{6.1}
\]

It is assumed that the product \( \rho c_v \), the volume specific heat capacity, is constant. \( q_c \), denoting the conductive heat flux, can be expressed using Fourier’s first law

\[
\vec{q}_c = -\kappa \cdot \vec{\nabla} T \tag{6.2}
\]

Using the divergence theorem and assuming constant thermal conductivity \( \kappa \), equation 6.1 can be rewritten

\[
\int_V \rho c_v \frac{\partial T}{\partial t} dV = \int_S \kappa \cdot \vec{\nabla} T \cdot \vec{n} dS \tag{6.3}
\]

On the left hand side stands an expression for the temporal change of the thermal energy contained in a given volume. The right hand is the integral over the boundaries of the same volume of the projection of heat fluxes onto the outward pointing normal. The physical interpretation is that an imbalance of the sum of fluxes at the boundaries of a volume causes a temporal change of energy contained in that volume.

6.2.1 Discretisation

ROCFLU uses a cell centered approach. It computes the average cell temperatures. Note that since a second order accurate scheme in space is used, the difference between the cell average temperature and temperature at the cell centroid can be neglected [19].

6.2.1.1 Spatial

The right hand side of equation 6.3 must be discretised in space. Formulated for a finite control volume \( \delta V \), delimited by \( i = 1 \ldots N \) faces with face surfaces \( \delta S_i \) this reads

\[
RHS = \sum_{i=1}^{N} -\kappa \cdot \begin{pmatrix} (\delta_x T)|_i \\ (\delta_y T)|_i \\ (\delta_z T)|_i \end{pmatrix} \cdot \begin{pmatrix} n_{x,i} \\ n_{y,i} \\ n_{z,i} \end{pmatrix} \cdot \delta S_i \tag{6.4}
\]

where \((\delta_x T)|_i, (\delta_y T)|_i, (\delta_z T)|_i\) denote the three components of the approximated temperature gradient at face \( i \) and \( n_{x,i}, n_{y,i}, n_{z,i} \) the components of the outward pointing normal vector of face \( i \).

The reconstruction algorithm for face gradients used in the fluid is adopted in the solid (see section 2.4.1).
6.2.1.2 Temporal

The temporal discretisation approximates the temporal gradient of the temperature $\frac{\delta T}{\delta t}$ with

$$\frac{\delta T}{\delta t} = \frac{RHS}{\rho \cdot c_v \cdot \delta V} \quad (6.5)$$

As for the fluid, the classical Runge Kutta scheme is used for time integration (see section 2.6).

6.3 Interface Temperature

The algorithm for interface temperature computation is inspired by the gradient reconstruction described in section 2.4.1 and especially the implementation of constrained reconstruction for Dirichlet temperature boundary conditions in ROCFLU (section 2.4.2). It was outlined in section 5.1.2 that a discontinuity in temperature gradient can exist at an interface between two materials. This must be taken into account when reconstructing the temperature gradients at the interface. Two gradients must be defined for each interface face, one for the fluid and one for the solid. This is indicated by the superscript $F$ or $S$ in the following. Hence two stencils must be built for an interface face (figure 6.1).

![Figure 6.1: Interface face and adjacent cells (left), solid stencil (middle), fluid stencil (right)](image)

6.3.1 Building Stencils

The task of building stencils for interface faces is very similar to that of building a stencil for a boundary face. One difference is that it must be made sure that only cells from the fluid domain or solid domain respectively are used.

Another requirement that applies is that the fluid stencil must be identical when built in
the fluid domain or solid domain. This is possible, since interface adjacent cells that are of different material than the region are included as virtual cells (see section 6.3.4). The original algorithm in ROCFLU for building face stencils is given in algorithm 1

Algorithm 1 ROCFLU algorithm for building boundary face stencils

1: Build first layer of stencil consisting of face adjacent cell
2: for iLayer = 2, nLayersMax do
3:   if stencilSize >= stencilSizeMin then
4:     build A matrix, invert A ▷ See equation 2.5
5:   end if
6:   if A is singular or stencilSize < stencilSizeMin then
7:     for isl = 1, nCellsInLastLayer do ▷ cells in last layer of stencil
8:       for iv2c = 1, nCellsShareVertex do ▷ cells sharing vertex with cell isl
9:         if cell iv2c not in stencil and stencilSize < stencilSizeMax then
10:            add cell
11:           stencilSize=stencilSize+1
12:       end if
13:     end for
14:   end for
15:   else
16:     exit for loop
17:   end if
18: end for

Two problems can be identified in that algorithm. First it could occur that the matrix $A$ is singular and no error message is issued. Also the cell members of a stencil depend on the order of the cell to vertex and vertex to cell list. These lists can be different for different partitionings. Therefore the stencils depend on the partitioning of the grid. It is of paramount importance that both of these problems are solved at the interface. Note that the dependence of the stencil on the partitioning could cause small differences between parallel and serial results of fluid only simulations.

The algorithm is slightly adapted for interfaces. In algorithm 2 compared to algorithm 1 lines seven to nine are new, line thirteen has been adapted, and lines seventeen to twenty-four are new.
Algorithm 2 ROCFLU algorithm for building interface face stencils

1: Build first layer of stencil consisting of face adjacent cell
2: for iLayer = 2, nLayersMax do
3:     if stencilSize >= stencilSizeMin then
4:         build A matrix, invert A
5:     end if
6:     if A is singular or stencilSize < stencilSizeMin then
7:         if stencilSize >= stencilSizeMax then
8:             error, stop program
9:         end if
10:    end if
11:     for isl = 1, nCellsInLastLayer do
12:         for iv2c = 1, nCellsShareVertex do
13:             if cell iv2c not in stencil and stencilSize < stencilSizeMax then
14:                 add cell to help stencil
15:             end if
16:         end for
17:     end for
18:     sort cells in help stencil according to ascending distance from face center
19:     for isl = 1, stencilHelpSize do
20:         if stencilSize < StencilSizeMax then
21:             add cell isl in help stencil to real stencil
22:             stencilSize = stencilSize + 1
23:         end if
24:     end for
25:     remove all cells from help stencil
26: else
27:     exit for loop
28: end if

Another requirement for interface stencils becomes apparent when considering a corner (figure 6.2). It must be made sure that only cells that are on the right side of an infinite (imaginary) plane defined by the face considered are contained in the stencil. If that is not ensured, the truncated Taylor series is formulated for cells between which the direct connection crosses another material.
6.3.2 Computing the Interface Temperature

In analogy to the gradient reconstruction in the fluid (section 2.4.1), truncated Taylor series are formulated. This is done separately for the fluid and solid side

\[
T_i^F = T_0 + (\vec{\nabla} T^F)|_0 \Delta \vec{r}_{0i}^F
\]

\[
T_i^S = T_0 + (\vec{\nabla} T^S)|_0 \Delta \vec{r}_{0i}^S
\]

The equations for the \(i = 1 \ldots m^F\) points in the fluid and \(i = 1 \ldots m^S\) points in the solid totaling to \(m = m^F + m^S\) equations are written in matrix form as \(A \cdot x = c\) with

\[
x = \begin{bmatrix}
T_1^F & T_2^F & \ldots & T_{m^F}^F & T_1^S & T_2^S & \ldots & T_{m^S}^S
\end{bmatrix}^T
\]

\[
x = \begin{bmatrix}
T_0 & (\delta_x T^F)|_0 & (\delta_y T^F)|_0 & (\delta_z T^F)|_0 & (\delta_y T^S)|_0 & (\delta_z T^S)|_0
\end{bmatrix}^T
\]

\[
A = \begin{bmatrix}
1 & \Delta x_{01}^F & \Delta y_{01}^F & \Delta z_{01}^F & 0 & 0 & 0 \\
1 & \Delta x_{02}^F & \Delta y_{02}^F & \Delta z_{02}^F & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
1 & \Delta x_{0m^F}^F & \Delta y_{0m^F}^F & \Delta z_{0m^F}^F & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & \Delta x_{01}^S & \Delta y_{01}^S & \Delta z_{01}^S \\
1 & 0 & 0 & 0 & \Delta x_{02}^S & \Delta y_{02}^S & \Delta z_{02}^S \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
1 & 0 & 0 & 0 & \Delta x_{0m^S}^S & \Delta y_{0m^S}^S & \Delta z_{0m^S}^S
\end{bmatrix}
\]

\[
c = \begin{bmatrix}
T_1^F & T_2^F & \ldots & T_{m^F}^F & T_1^S & T_2^S & \ldots & T_{m^S}^S
\end{bmatrix}^T
\]

\[
A \text{ is a } m\text{-by-}n \text{ matrix, } x \text{ a } n\text{-by-}1 \text{ vector, and } c \text{ a } m\text{-by-}1 \text{ vector. It is assumed that the stencils are built such that } m > n \text{ holds (section 2.4.1).}
\]
The equations are weighted with weights $\xi_{0i}^F$ and $\xi_{0i}^S$ that are inversely proportional to the distance between the point 0 and $i$ [16], [17]

$$\xi_{0i}^F = \left\| \Delta \vec{r}_{0i}^F \right\|^{-1}$$

$$\xi_{0i}^S = \left\| \Delta \vec{r}_{0i}^S \right\|^{-1} \quad (6.11)$$

Using the $m$-by-$m$ matrix $\tilde{W}$ this can be rewritten as

$$\tilde{W} \cdot A \cdot x = \tilde{W} \cdot c \quad (6.13)$$

with $\tilde{W}$ defined as

$$\tilde{W} = \begin{pmatrix}
\xi_{01}^F & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & \xi_{02}^F & \ddots & & & & \\
\vdots & \ddots & \ddots & \ddots & & & \\
0 & \ldots & 0 & \xi_{0m}^F & 0 & \ldots & 0 \\
0 & \ldots & \ldots & 0 & \xi_{01}^S & 0 & \ldots & 0 \\
\vdots & & & & \ddots & \ddots & \ddots & \\
0 & \ldots & \ldots & \ldots & \ldots & 0 & \xi_{0m}^S
\end{pmatrix} \quad (6.14)$$

The constraints in the case of interface temperature computation are derived from first principles (section 4.2). The continuity of temperature across the interface is intrinsically fulfilled. The continuity of fluxes must be formulated as constraint.

The convention for normal vectors used in the ROCFLU source and in this thesis is that the normals are uniquely defined for each face and point outwards of the fluid domain. Two parallel vectors forming a right handed coordinate system with the normal are also uniquely defined for each face (figure 6.3).

![Figure 6.3: Normal and parallel vectors](image-url)
If the heat conductivity in the fluid is denoted with $\kappa^F$ and in the solid with $\kappa^S$ and with the above convention for normal vectors, the continuity of fluxes requires that

$$\kappa^F \cdot (\vec{\nabla} T^F)|_0 \cdot \vec{n} = \kappa^S \cdot (\vec{\nabla} T^S)|_0 \cdot \vec{n}$$  \hspace{1cm} (6.15)$$

While a discontinuity in temperature gradient normal to an interface can occur, this is not the case parallel to an interface if the heat conductivity is assumed constant in both domains. Continuity of gradients parallel to an interface face is enforced by requiring that the following two equations are exactly fulfilled

$$(\vec{\nabla} T^F)|_0 \cdot \vec{p}_1 = (\vec{\nabla} T^S)|_0 \cdot \vec{p}_1$$  \hspace{1cm} (6.16)$$

$$(\vec{\nabla} T^F)|_0 \cdot \vec{p}_2 = (\vec{\nabla} T^S)|_0 \cdot \vec{p}_2$$  \hspace{1cm} (6.17)$$

In matrix form the three constraints can be written as $B \cdot x = d$ for the same $x$ as in equation 6.8

$$B = \begin{bmatrix} 0 & p_{1,x} & p_{1,y} & p_{1,z} & -p_{1,x} & -p_{1,y} & -p_{1,z} \\ 0 & p_{2,x} & p_{2,y} & p_{2,z} & -p_{2,x} & -p_{2,y} & -p_{2,z} \\ 0 & \kappa^F \cdot n_x & \kappa^F \cdot n_y & \kappa^F \cdot n_z & -\kappa^S \cdot n_x & -\kappa^S \cdot n_y & -\kappa^S \cdot n_z \end{bmatrix}$$

$$d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$$  \hspace{1cm} (6.19)$$

Using the above algorithm, the determination of the interface temperature has been reduced to a least squares problem with constraints in a manner that is consistent with the gradient reconstruction as it is used in the fluid in ROCFLU.

In its most compact form, the computation of the interface temperature has been reduced to a matrix equation for $n$ unknowns in the form

$$\begin{pmatrix} \tilde{W} \cdot A \\ B \end{pmatrix} \cdot x = \begin{pmatrix} \tilde{W} \cdot c \\ d \end{pmatrix}$$  \hspace{1cm} (6.21)$$

where $\tilde{W} \cdot A = \tilde{W} \cdot c$ are equations that must be solved in a least squares manner with the constraints $B \cdot x = d$ imposed.

**In Two Dimensions**

In two dimensions the algorithm must be slightly adapted. This is because the three dimensional formulation would lead to a singular matrix $A$ and thus the least squares problem could not be solved. Also in two dimensions the requirement of continuous gradients parallel to interfaces reduces from two equations to one.

In the following it is assumed that two dimensional computations are performed in the $x−y$--plane as done in ROCFLU. The equations from the formulation of the truncated Taylor series in matrix form become $A \cdot x = c$ with

57
\[
A = \begin{bmatrix}
1 & \Delta x^F_{01} & \Delta y^F_{01} & 0 & 0 \\
1 & \Delta x^F_{02} & \Delta y^F_{02} & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & \Delta x^F_{0mF} & \Delta y^F_{0mF} & 0 & 0 \\
1 & 0 & 0 & \Delta x^S_{02} & \Delta y^S_{02} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 0 & 0 & \Delta x^S_{0ms} & \Delta y^S_{0ms}
\end{bmatrix}
\]
\[
B = \begin{bmatrix}
0 & p_{1,x} & p_{1,y} & -p_{1,x} & -p_{1,y} \\
0 & \kappa^F \cdot n_x & \kappa^F \cdot n_y & -\kappa^S n_x & -\kappa^S n_y
\end{bmatrix}
\]
\[
d = [0 \ 0 \ 0]^T
\]

A is a \(m\)-by-\(n\) matrix, \(x\) a \(n\)-by-1 vector, and \(c\) a \(m\)-by-1 vector.

The constraints written in matrix form become

\[
B \cdot x = d
\]

with

\[
\begin{align*}
\kappa^F \cdot (\nabla T^F)|_0 \cdot \vec{n} - \kappa^S \cdot (\nabla T^S)|_0 \cdot \vec{n} &= q_{\text{source}} \\
B \cdot x &= d
\end{align*}
\]

In three dimensions the constraints can be written in matrix form as

\[
B \cdot x = d
\]
with the same \( B \) as in the case without radiation but a modified the vector \( d \)

\[
B = \begin{bmatrix}
0 & p_{1,x} & p_{1,y} & p_{1,z} & -p_{1,x} & -p_{1,y} & -p_{1,z} \\
0 & p_{2,x} & p_{2,y} & p_{2,z} & -p_{2,x} & -p_{2,y} & -p_{2,z} \\
0 & \kappa^F \cdot n_x & \kappa^F \cdot n_y & \kappa^F \cdot n_z & -\kappa^S n_x & -\kappa^S n_y & -\kappa^S n_z
\end{bmatrix}
\]

(6.31)

\[
d = \begin{bmatrix} 0 \ 0 \ q_{source} \end{bmatrix}^T
\]

(6.32)

Note that since only the right hand side of the constraints and not the equation as such is modified compared to the case without radiation, no numerical problems are expected if the source term should be very small and oscillating around zero.

### 6.3.4 Parallelisation

It was mentioned in section 6.1 that regions consist of one single material. As a consequence, interfaces must be region limits. This means that if the interface temperature is computed in the fluid region, it must be communicated to the solid region. The same applies in the opposite case.

Another possibility is to determine the interface temperatures twice, once in each region. This requires the knowledge of the temperatures and heat conductivities of the cells that are in both stencils associated with each interface face. Figure 6.4 illustrates the case of partitioning along an interface with the stencil members for an interface face (square point) indicated by dashed lines and virtual cells highlighted in grey.

The second option is chosen because it can be seamlessly integrated in the ROCFLU parallelisation process as it is used for the fluid.

Computing interface temperature and fluxes in both regions implies that the stencils for the interface face must be the same in both fluid and solid regions. It is made sure that this is the case (see section 6.3.1). If a simulation is run serially, it is explicitly checked that the stencils are identical.

![Figure 6.4: Illustration of interface stencils (dotted lines) for an interface face (square point) in the serial case (left) and partitioning with virtual cells highlighted in grey (middle, right)](image-url)
Chapter 7

Results

In this chapter, results obtained with ROCFLU including the new features are presented. For the purpose of verification, simple cases are shown and compared to analytical or reference solutions. In order to demonstrate the capabilities to simulate complex geometries, a generic reactor geometry is simulated.

7.1 Radiative Heat Exchange

The radiosity method for radiative heat exchange in an enclosure has been presented in section 3.2.2. It is important to keep in mind, that this method is applicable only if the incoming heat flux is uniform. This may not seem very restrictive at first, but it is. It will be shown that for two parallel, rectangular, finite plates at a given temperature the incoming heat fluxes are not uniform if the plates are grey.

In the following, the radiative heat exchange between two parallel plates is used as a verification case. In the case of two black plates, ROCFLU results are compared to the solution obtained with the radiosity method. In the case of grey surfaces, a solution obtained using the in-house ray tracing software VeGaS+ [27] is used as a reference solution. As a more complex geometry, the solution for the radiative heat transfer in a generic reactor geometry is presented.

7.1.1 Two Parallel Plates

The case of two identical parallel, rectangular plates with non participating surroundings is illustrated in figure 7.1.
According to [22] the configuration factors are given by equation 7.3

\[ X = \frac{w}{h} \]  
\[ Y = \frac{l}{h} \]  
\[ F_{1-2} = F_{2-1} = \frac{2}{\pi XY} \cdot \left[ \frac{1}{2} \ln \left( \frac{(1 + X^2)(1 + Y^2)}{1 + X^2 + Y^2} \right) + \left( X \sqrt{1 + Y^2} \arctan \frac{X}{\sqrt{1 + Y^2}} + Y \sqrt{1 + X^2} \arctan \frac{Y}{\sqrt{1 + X^2}} - X \arctan(X) - Y \arctan(Y) \right) \right] \] (7.3)

In order to use the radiosity method, the non participating surroundings are treated as a third surface (see section 3.2.2). Using conservation of energy and the reciprocity between \( F_{k-j} \) and \( F_{j-k} \) (equation 3.9), the matrix of configuration factors is found (table 7.1).

**Figure 7.1:** Two parallel, rectangular plates with non participating surroundings, dimensions \( h \times w \times l \), diffuse emission from plate two
Table 7.1: Configuration factors for two identical parallel rectangular plates (indices 1 and 2) and surroundings (3)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>$F_{1-2}$</td>
<td>$1 - F_{1-2}$</td>
</tr>
<tr>
<td>2</td>
<td>$F_{2-1}$</td>
<td>0</td>
<td>$1 - F_{2-1}$</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{A_1}{A_3}(1 - F_{1-2})$</td>
<td>$\frac{A_2}{A_3}(1 - F_{1-2})$</td>
<td>$1 - \frac{A_1}{A_3}(1 - F_{1-2}) - \frac{A_2}{A_3}(1 - F_{1-2})$</td>
</tr>
</tbody>
</table>

Applying the radiosity method (equation 3.14), using $\varepsilon_3 = 1, T_3 = 0$ for the non participating surroundings and simplifying leads to the following expressions for the net heat fluxes $q_i$

$$q_1 = \varepsilon_1 \cdot \frac{\sigma T_1^4 [1 - F_{1-2} F_{2-1}(1 - \varepsilon_2)] - \varepsilon_2 \cdot F_{1-2} \cdot \sigma T_2^4}{1 - (1 - \varepsilon_1)(1 - \varepsilon_2) F_{1-2} F_{2-1}}$$  \hspace{1cm} (7.4)

$$q_2 = \varepsilon_2 \left[ \frac{1 - \varepsilon_1}{\varepsilon_1} F_{2-1} q_1 - F_{2-1} \sigma T_1^4 + \sigma T_2^4 \right]$$  \hspace{1cm} (7.5)

$$q_3 = \frac{1}{A_3} \cdot (-A_1 q_1 - A_2 q_2) =$$  \hspace{1cm} (7.6)

$$= F_{3-1} \left[ \frac{1 - \varepsilon_1}{\varepsilon_1} q_1 - \sigma T_1^4 \right] + F_{3-2} \left[ \frac{1 - \varepsilon_2}{\varepsilon_2} q_2 - \sigma T_2^4 \right]$$

For the test cases in this section, the following dimensions are used

$$h = 0.1m$$  \hspace{1cm} (7.7)

$$w = l = 0.5m$$  \hspace{1cm} (7.8)

The numerical values for the configuration factors are given in table 7.2.

Table 7.2: Configuration factors for two identical parallel plates with dimensions given in 7.7 and 7.8

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.6902446941</td>
<td>0.3097553059</td>
</tr>
<tr>
<td>2</td>
<td>0.6902446941</td>
<td>0</td>
<td>0.3097553059</td>
</tr>
<tr>
<td>3</td>
<td>0.3871941324</td>
<td>0.3871941324</td>
<td>0.2256117352</td>
</tr>
</tbody>
</table>

7.1.1.1 Black Surfaces

In the case of black surfaces, the expressions for the net heat fluxes simplify considerably. This result can be found without the use of the radiosity method. Inserting $\varepsilon_1 = \varepsilon_2 = 1$
into equations 7.4 to 7.6 leads to

\[ q_1 = \sigma T_1^4 - F_{1-2} \cdot \sigma T_2^4 \]  
\[ q_2 = \sigma T_2^4 - F_{2-1} \cdot \sigma T_1^4 \]  
\[ q_3 = -F_{3-1} \cdot \sigma T_1^4 - F_{3-2} \cdot \sigma T_2^4 \]

(7.9)  
(7.10)  
(7.11)

For this verification case, the temperatures for the two plates are chosen as

\[ T_1 = 1000 \text{ K} \]  
\[ T_2 = 900 \text{ K} \]

(7.12)  
(7.13)

The values for the net heat transferred evaluate to

\[ Q_1 = A_1 \cdot q_1 = 7756.121 \text{ W} \]  
\[ Q_2 = A_2 \cdot q_2 = -484.035 \text{ W} \]  
\[ Q_3 = A_3 \cdot q_3 = -7272.086 \text{ W} \]

(7.14)  
(7.15)  
(7.16)

For verification, the total number of rays used is varied and the error is investigated. The total number of rays \((N_R)\), is the sum of the rays emitted from plate one \((N_{R,1})\) and two \((N_{R,2})\). For the rays to carry the same power, the number of rays is distributed according to the power emitted from the two plates

\[ \frac{N_{R,1}}{N_{R,2}} \approx \frac{T_1^4}{T_2^4} \]

(7.17)

The approximately equal sign is there since the number of rays is rounded to the nearest integer, which leads to small differences in ray power between the rays emitted from plate one and plate two.

When investigating convergence, the standard deviation of the result is at least as important as the actual error in the result. Given Monte Carlo results \(X_i\) from \(i = 1 \ldots N\) runs performed with the same number of rays \((N_R)\), the average \((\mu)\) and variance \((\sigma^2)\) can be estimated as [9]

\[ \text{estimation}(\mu) = \bar{X}_N = \frac{1}{N} \sum_{i=1}^{N} X_i \]  
\[ \text{estimation}(\sigma^2) = S_N^2 = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X}_N)^2 \]

(7.18)  
(7.19)

Figure 7.2 illustrates the results for the net heat exchanged between the plates and surroundings. The above described setup is used.

For each given number of rays, ten runs were performed with a different set of random numbers used for ray generation and surface interaction. The relative error is computed as the absolute value of the difference between the analytical solution and the average of
the ten runs divided by the exact solution. The standard deviation is estimated according to equation 7.19 and normalised.

\[
\text{relative error} = \frac{\text{abs}(Q_{\text{exact}} - \bar{X}_N)}{Q_{\text{exact}}} \quad (7.20)
\]

\[
\text{normalised standard deviation} = \frac{\sqrt{S^2_N}}{Q_{\text{exact}}} \quad (7.21)
\]

**Figure 7.2:** Relative error and normalised standard deviation for net radiative heat fluxes exchanged between two identical rectangular parallel plates in dependency of the number of rays, estimated from ten runs.
It is clearly visible that the standard deviation decreases proportionally to the inverse square root of the number of rays. This is the rate of convergence that is expected [15]. The decrease of the error does not form such a clear line. However for all three net heat fluxes it can be seen that the maximum error is limited by a decrease proportional to $N_{R}^{-0.5}$.

From this study it can be concluded that the Monte Carlo implementation in ROCFLU leads to the results that are expected for this most simple setup.

### 7.1.1.2 Gray Diffuse Surfaces

In a next step the same setup of two identical parallel plates is used but the plates are grey. The following emissivities are used

\[ \varepsilon_1 = 0.2 \quad (7.22) \]
\[ \varepsilon_2 = 0.3 \quad (7.23) \]

For grey plates, it does not make sense to compare ROCFLU results to results obtained with the radiosity method. This is because the radiative incoming heat flux is not constant in space (figure 7.3) and thus the radiosity method is not applicable. Instead, ROCFLU results are compared with results obtained with the in-house software VeGaS+ [27].

The setup is chosen in analogy to that used for the black plates (section 7.1.1.1). The total number of rays is distributed to the plates according to the emitted energy

\[ \frac{N_{R,1}}{N_{R,2}} \approx \frac{\varepsilon_1 T_1^4}{\varepsilon_2 T_2^4} \quad (7.24) \]

**Figure 7.3:** Incoming radiative heat flux for grey, parallel, identical plates, plate one and two (from the top), in $\text{W m}^{-2}$, obtained with ROCFLU using $10^8$ rays ($T_1 = 1000 \text{ K}, \varepsilon_1 = 0.2, T_2 = 900 \text{ K}, \varepsilon_2 = 0.3, h = 0.1 \text{ m}, w = l = 0.5 \text{ m}$)
Since there is no analytical solution, the error cannot be computed. Instead the resulting values for the net powers $Q_i$ are shown with the standard deviation illustrated with error bars (figures 7.4 and 7.5). The standard deviation is computed as in the previous section (equation 7.19). Ten runs were performed for each number of rays.

It is clearly visible that ROCFLU and VeGaS+ results converge to the same value and the standard deviation decreases and is similar.

**Figure 7.4:** Comparison of ROCFLU and VeGas+ results for heat exchange between two identical rectangular parallel diffuse grey plates, results from ten runs for each number of rays, standard deviation indicated with error bars
In the following, the decrease of standard deviation is investigated (figures 7.6 and 7.7). It is not normalised. For ROCFLU and VeGaS+ results, the standard deviation is very similar and decreases as expected proportionally to the square root of the total number of rays used.

**Figure 7.6:** Comparison of ROCFLU and VeGas+ results, standard deviation for heat exchange between two identical rectangular grey diffuse parallel plates, estimated from ten runs.
Figure 7.7: Comparison of ROCFLU and VeGas+ results, standard deviation for heat exchange between two identical rectangular parallel grey diffuse plates, estimated from ten runs.

From this study it can be concluded that the Monte Carlo implementation in ROCFLU leads to the results that are expected for this setup.

7.1.1.3 Gray Diffuse and Specular Surfaces

What has been done for gray diffuse plates is repeated for the case of diffuse emission, diffuse reflection from plate two and specular reflection from plate one. The results are illustrated in figures 7.8 to 7.11. Based on this study, it can be concluded that the Monte Carlo implementation in ROCFLU leads to the expected, correct results.
Figure 7.8: Comparison of ROCFLU and VeGas+ results for heat exchange between two identical rectangular diffuse grey and diffuse specular parallel plates, results from ten runs for each number of rays, standard deviation indicated with error bars.
**Figure 7.9:** Comparison of ROCFLU and VeGas+ results for heat exchange between two identical rectangular diffuse grey and diffuse specular parallel plates, results from ten runs for each number of rays, standard deviation indicated with error bars

**Figure 7.10:** Comparison of ROCFLU and VeGas+ results, standard deviation for heat exchange between two identical rectangular parallel grey diffuse and grey specular plates, estimated from ten runs
Figure 7.11: Comparison of ROCFLU and VeGas+ results, standard deviation for heat exchange between two identical rectangular parallel grey diffuse and grey specular plates, estimated from ten runs.
7.1.2 Generic Reactor Geometry

A key advantage of Monte Carlo ray tracing is that complex geometries can be simulated. In order to demonstrate the capabilities of ROCFLU, a generic reactor geometry is simulated. A planar cut through the geometry is given in figure 7.12. The reactor is axi-symmetric.

Figure 7.12: Planar cut through axisymmetric geometry of generic reactor, dimensions are in mm, solid is hatched
The geometry was drawn in Siemens NX 8.5, exported in the IGES format, read in by CENTAUR, meshed with that tool and exported in the CENTAUR in-house format. This format can be read in by ROCFLU.

The shape of the compound parabolic concentrator is given by the following parametrisation with parametric angle $\phi \in (\theta_A, \frac{\pi}{2})$

\[
z_{\text{CPC}}(\phi) = \frac{2 \cdot \cos(\phi)}{1 - \cos(\phi + \theta_A)} \cdot f
\]  
\[
r_{\text{CPC}}(\phi) = \frac{2 \cdot \sin(\phi)}{1 - \cos(\phi + \theta_A)} \cdot f - r_e
\]

where $\theta_A$ is the acceptance angle, $r_e$ the exit radius, $f = [1 + \sin(\theta_A)] \cdot r_e$ the focal length. This formula has been extracted from the VeGaS+ source code.

In the present case, an acceptance angle of $\theta_A = 45^\circ$ and an exit radius of $r_e = 0.02$ m is chosen (figure 7.13).

For the simulation, incoming light is assumed at the reactor inlet highlighted in green in figure 7.12. It is diffuse but restricted between a polar angle relative to the vertical axis of $0 \leq \theta < 45^\circ$. A radiative incoming flux of $7 \cdot 10^6$ W m$^{-2}$ is prescribed. The inlet is non-participating, thus an absorptivity of $\alpha = 1$ is used.

Surface properties must be prescribed. The surfaces of the concentrator part, highlighted in blue in figure 7.12, are modeled as perfect specular reflector with $\rho = 1$. The cavity, highlighted in red, is modeled as the surface of aluminum oxide. It is a diffuse reflector and emitter with an emissivity of $\varepsilon = 0.55$. This value is taken from [32] at $T \approx 300$ K.

All surface temperatures are prescribed as $T = 293.15$ K. The radiative heat flux emitted from the cavity surface is neglected since it is several orders of magnitude below the incoming radiation.

The resulting incoming heat fluxes are illustrated in figure 7.14 and 7.15.
Figure 7.14: ROCFLU result for incoming radiative heat fluxes in Watts, simulation with 150 million incoming rays, side view

Figure 7.15: ROCFLU result for incoming radiative heat fluxes in Watts, simulation with 150 million incoming rays, bottom view
At first, it seems surprising that just below the concentrator part, the incoming heat flux is so big. However, it is not spurious, a clear result of the simulation and can be explained. Consider directional incoming radiation in vertical direction. In that case a considerable part of the radiation is reflected by the concentrator part, such that it hits exactly the most upper part of the cavity. This is illustrated in figure 7.16 that is drawn to scale and with the correct angles. Similar effects are responsible for the observed peak in the case of diffuse incoming radiation.

![Illustration of directional incoming light, drawn to scale](image)

**Figure 7.16:** Illustration of directional incoming light, drawn to scale

The geometry is axially symmetric. Therefore the incoming radiation at the cavity wall is a function of the vertical coordinate only. The values of $q_i$ can be plotted against the vertical coordinate, where the origin of the coordinate system is chosen at the bottleneck of the geometry, the limit between concentrator part and cavity. This is done in figure 7.17 for simulation results obtained with $1.5 \cdot 10^6$, $15 \cdot 10^6$ and $150 \cdot 10^6$ incoming rays. The decrease in variance and the converging behaviour of the solution is clearly visible.

The radiative power leaving the cavity is illustrated in figure 7.18. The power that is rejected by the reactor is considerable. According to this simulation, only 31.7% are absorbed by the cavity, 68.3% are rejected. This is due to the diffuse incoming light, in that case the CPC does not concentrate the light.
Figure 7.17: Incoming radiative heat flux distribution at cavity wall as a function of the vertical coordinate for simulations performed with different numbers of incoming rays.

Figure 7.18: ROCFLU result for rejected light, in Watts, simulation with 150 million incoming rays, top view.
7.2 Solid-Solid Interaction

In chapter 5, the case of two solids in contact was considered as test case in one dimension. The ROCFLU solution to that problem is shown and compared to the analytical solution for verification.

Note that ROCFLU cannot in general compute solutions for two solids in contact. For this special case, the mass and momentum fluxes were set to zero in the fluid domain in a special version of the source code. In that way, the temperature profile for the limit of an incompressible, infinitely viscous fluid is computed which is equivalent to a solid.

Two Materials in Contact

A two dimensional grid is used in ROCFLU (figure 7.19). In order to approximate a one dimensional problem, a vanishing heat flux normal to the boundaries is prescribed at the limits in $y$ direction. The two domains are initialised to two different, constant initial temperatures such that no gradients in $y$ direction should exist.

For comparison, the temperature values in the middle cell in $y$ direction are used. This is a detail as gradients in $y$ direction in the result are indeed negligible.

The same setup is used as described in section 5.5, namely gold brought into contact with iron and a temperature difference of 700 K. The only exception is that here a time step of $\Delta t = 10^{-3}$ s is used which is slightly below the time step of $\Delta t = 1.2953 \cdot 10^{-3}$ s in section 5.5. Five cells are used in $y$-direction with an extent of 0.04 m. The extent of the grid in $z$ direction is not relevant but also chosen as 0.04 m.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{grid.png}
\caption{Two dimensional grid used for simulation of a one dimensional test case in ROCFLU, red is the gold domain, blue is iron}
\end{figure}
For comparison, the one dimensional formula for determination of the interface temperature (equation 5.34) is implemented in ROCFLU. Figure 7.20 illustrates the spatial error distribution normalised with the analytical value for the interface temperature. The ROCFLU result with that modification is very close to that obtained with a one dimensional code.

When using the algorithm described in section 6.3 for determining the interface temperature, the error is larger. This can be explained. Reasons for that include the fact that a two dimensional domain is used, for a one dimensional problem and a method designed for an unstructured grid is applied to a structured grid. Also the steep local temperature gradients at early times lead to the fact that the one dimensional formula estimates the gradients at the interface more accurately, since it uses only one adjacent point, whereas in the case of constrained reconstruction three layers of cell are contained in the stencil used for interface temperature computation.

Figure 7.21 shows the comparison of relative error in interface temperatures. Clearly using the one dimensional formula the error decreases very quickly. The convergence when using constrained reconstruction is slower.

As a conclusion it can be said that the implementation for interface temperature determination in ROCFLU leads to the expected results in a one dimensional test case, simulated using a two dimensional grid and appropriate boundary and initial conditions.

![Graph showing the comparison of relative error in interface temperatures.](image)

**Figure 7.20:** Normalised spatial error distribution for numerical solutions of the one dimensional test case described in section 5.5 obtained using a one dimensional code and ROCFLU
Figure 7.21: Relative error in interface temperature for numerical solutions of the one-dimensional test case described in section 5.5 obtained using a one-dimensional code and ROCFLU

7.3 Fluid Solid Radiation Thermal Interaction

In order to demonstrate the capabilities of ROCFLU to simulate fluid solid radiation interaction, the generic reactor geometry presented in section 7.1.2 is simulated. The contained fluid is air with the following properties

\[
c_p = 880 \, \frac{J}{kg \cdot K} \quad (7.27)
\]

\[
\gamma = 1.4 \quad (7.28)
\]

\[
Pr = 0.712 \quad (7.29)
\]

\[
\mu = 0.00001511 \, \text{Pa} \cdot \text{s} \quad (7.30)
\]

The surrounding material is aluminum oxide with the following properties [23]

\[
\rho = 3940 \, \frac{kg}{m^3} \quad (7.31)
\]

\[
c_v = 880 \, \frac{J}{kg \cdot K} \quad (7.32)
\]

\[
\kappa = 30 \, \frac{W}{m \cdot K} \quad (7.33)
\]

The initial condition is that the fluid is at rest and fluid and solid domains are at ambient temperature (293.15 K) and the fluid at ambient pressure (101432 Pa).
The time step is specified via the CFL number in the input. A CFL number of 2.9 is used. Note that the concept of CFL number is not uniquely extendable from structured to unstructured grids. The maximum time step computed by ROCFLU in the fluid domain is limiting, as expected. This leads to a time step of $\Delta t = 4.4233 \cdot 10^{-7}$ s which is constant during the $10^{-2}$ s of physical time simulated.

The radiation from the cavity walls is neglected since it is orders of magnitude below the incoming radiative power. Only the incoming radiative fluxes are relevant. The result for incoming radiative power from section 7.1.2 is used at each time step. Note that this can be done only because the simulation is not run for a very long time. Should the walls heat up and emission from the cavity walls contribute significantly to the radiative heat exchange, the radiative heat transfer would have to be computed at every time step (or at least more often). This is implemented but was disabled for this case to allow for a computationally more efficient simulation.

As a consequence of the incoming radiation, the walls heat up. Figure 7.22 Shows the computed interface temperature after $10^{-2}$ s. The concentrator part remains at the initial temperature since it reflects all incoming radiation. The temperature profile in the cavity is consistent with the incoming radiative heat fluxes (figure 7.15).

A problem seems to occur at the lower cavity corners. It is physically impossible that the wall temperature drops below the initial temperature. It is not clear why that happens. Grid refinement could help but this is most likely not the main reason.

Close to the wall, a temperature boundary layer develops. Figure 7.23 Illustrates the temperature profile after $10^{-2}$ s at a height of $z = -0.04$ m as a function of the radial coordinate ($r = \sqrt{x^2 + y^2}$). The interface is located at $r = 0.04$ m. Figure 7.23 was created by extracting from the full geometry all cells that have their center of gravity in the interval $z = (-0.0403$ m, $-0.0397$ m) and plotting all those temperature values as a function of the radius. Since the geometry is axially symmetric, the values at a given radius are very similar, even hard to distinguish, as expected.
Figure 7.22: ROCFLU results for interface temperatures for fully coupled simulation of generic reactor geometry after $10^{-2}$ s
Figure 7.23: Temperature profile as a function of the radial coordinate \( r = \sqrt{x^2 + y^2} \) at a vertical height of \( z = -0.04 \) m after \( 10^{-2} \) s

It was outlined in section 4.2 that the novel algorithm enforces continuity of fluxes at the interface. This can be verified by plotting the normalised error in heat balance (see figure 4.3 for notation)

\[
\frac{q_{\text{rad,in}} - q_{\text{rad, out}} - q_{\text{fluid}} + q_{\text{solid}}}{q_{\text{rad,in}}}
\]  

(7.34)

During the simulation it is checked that this error never exceeds a value of \( 10^{-5} \). Figure 7.24 illustrates this quantity after \( 10^{-2} \) s. Indeed the normalised error in energy conservation at the interface is very small.

In the fluid the full compressible Navier Stokes equations are solved. Figure 7.25 illustrates a pressure distribution after \( 4 \cdot 10^{-4} \) s. Pressure waves due to the incident radiative heat fluxes, disturbing a setup that was at equilibrium, are visible. The initial pressure was \( 101432 \) Pa.
Figure 7.24: Normalised error in heat balance at the interface after $10^{-2}$ s

Figure 7.25: Pressure distribution after $4 \cdot 10^{-4}$ s, solid in black
Chapter 8

Conclusion

In summary, it can be said that the implementation of a novel algorithm for fluid structure thermal interaction and coupling with radiative heat transfer in an in-house CFD tool has been successful.

The implementation of a Monte Carlo Ray tracing algorithm into a CFD tool that is capable of importing unstructured meshes allows to fully use the advantage of the Monte Carlo method that imposes no restriction whatsoever on the geometry to be simulated. After formulating a novel approach for fluid structure thermal interaction, it was investigated whether this shows good stability and accuracy properties. This was indeed the case, so that it was implemented in ROCFLU. This was a very time consuming process. The data structure being adapted, possibilities of adapting the algorithm for fluid structure interaction and radiative heat exchange or testing new ideas in a reasonable amount of time are given.

The results presented show one minor issue that should be further investigated. It would also make sense to look into computational efficiency as this was not a major goal of the present work. It is expected that the Monte Carlo ray tracing algorithm could be made more efficient. Furthermore, it is desired that more sophisticated features such as several solids in contact are implemented. The implementation of fluid structure interaction coupled with radiation in the implicit solver that is already implemented in ROCFLU for the case of pure fluid simulations would be of great use.

Finally, it would be most interesting to compare ROCFLU results of a fluid structure thermal interaction simulation coupled with radiation to those obtained with other softwares, such as ANSYS.

I wish to express my gratitude to Men Wirz for his kind support with CENTAUR, Tom Cooper for his help with VeGaS+ and radiative heat transfer, Philipp Furler for the inputs when designing the generic reactor geometry, Silvan Siegrist, Fabian Wermelinger for saving me hours with computer issues, my brother for his help with NX, Prof. Patrick Jenny for his input on multi-point flux approximation methods, Dr. Jimmy Olsson for a brilliant lecture on Monte Carlo methods given at Lund University and all the people that have supported me in one way or another.
Bibliography


Appendices
Appendix A

Thesis description
1. Background

The performance of many engineering systems is partly determined by the interplay of convection, conduction, and radiation heat-transfer mechanisms. The equations governing the interplay of these modes, especially in the complex geometries typical of engineering systems, are not amenable to theoretical approaches. Therefore, simulations based on numerical solutions of these equations are necessary. As a first step toward coupled radiation-convection-conduction simulations, the simpler coupled radiation-conduction and conjugate heat-transfer simulations can be considered. In conjugate heat-transfer simulations, numerical stability is often a problem. For example, the stability of conjugate heat-transfer simulations was analyzed in references [1,2]. A novel approach to solving the conjugate heat-transfer problem using the finite-volume method has been tested in one dimension and shown good stability properties. A sample solution obtained using this approach is compared to the analytical solution in Figure 1. This novel approach has the virtue of simplicity and is easily implemented in existing finite-volume methods.

The long-term goal of this work is to enable coupled radiation-convection-conduction simulations of participating media such as particulate multiphase flows in complex geometries typical of chemical reactors. This master thesis represents the first step toward this goal. The goal will be reached by building on top of an existing finite-volume code for three-dimensional simulations in complex geometries. This code is already equipped with a Monte-Carlo ray-tracing capability for non-participating media. With this first step achieved, it will be possible to simulate the radiative heating of a solid body such as a reactor structure and the accompanying conduction of heat within this body.
2. Objectives
The objectives of this master thesis are:

1. Implement conjugate heat-transfer approach in three-dimensional finite-volume code.
2. Extend interface conditions to include radiative fluxes computed by Monte-Carlo ray tracing.
3. Verify coupled radiation-conduction capability using exact solutions in [3].

Pre-requisite: Course “Theoretical and Applied Computational Fluid Dynamics” by Dr. A. Haselbacher.

3. List of Assignments

1. Analyze the stability of the novel conjugate heat-transfer approach in one dimension.
2. Extend the novel conjugate heat-transfer approach to three dimensions.
3. Implement the approach in an existing three-dimensional finite-volume code.
4. Couple the approach to an existing Monte-Carlo ray-tracing capability in the existing code to enable coupled radiation-conduction simulations.
5. Verify coupled radiation-conduction capability using exact solutions.
7. Seminar (to be held on December 20, 2013)
8. Report: 2 hardcopies +1 e-copy (MS-Word + PDF format + CD with raw data)

Note: List of assignments serves as guideline; adjustments may be required according to progress and results.

4. References