

Gradient calculations of non-orthogonal meshes in the finite volume method

N. van der Westhuizen
20546092

Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Engineering in Nuclear Engineering at the Potchefstroom Campus of the
North-West University

Supervisor: Dr. J-H Kruger
Co-Supervisor: Prof. C.G. Du Toit

September 2013

Opsomming

Die hantering van gradiënt berekeninge op nie-ortogonale roosters in die eindige volume metode is van besondere belang met betrekking tot die modellering van komplekse geometriese modelle, aangesien verskillende metodes die akkuraatheid en stabiliteit van die oplossing beïnvloed.

Die toepassing in die huidige studie behels 'n numeriese oplossing van hitte-oordrag in 'n komplekse geometrie. Die modellering van komplekse geometriese modelle het betrekking op baie ingenieurswese toepassings, soos byvoorbeeld 'n mikro-kanaal hitteruiler wat gebruik word in die kragkringloop van hoë temperatuur reaktore om helium te voor-verhit.

'n Program gebaseer op die eindige volume metode was ontwikkel in Excel om die diffusie vergelyking op 'n nie-ortogonale rooster op te los. 'n Toetsgeval van hitte-oordrag in 'n reghoekige blok met 'n tetraëdriese rooster was opgelos in Excel. Dieselfde toetsgeval was ook opgelos met OpenFOAM. Die resultate van die twee kodes was vergelyk en klein verskille was gevind as gevolg van effense verskillende implementeringsmetodes. Kennis oor die verskillende implementerings metodes het gehelp om die aspekte wat 'n invloed het op die akkuraatheid en stabiliteit van die oplossing beter te verstaan.

Toetsgevalle op roosters met die teenwoordigheid van skeefheid en met nie-ortogonale lyne by die wande van die domein was getoets en daar is waargeneem dat die akkuraatheid afgeneem het. Daar is gevind dat die standaard eindige volume metode soos geïmplementeer in Excel en in OpenFOAM, gevorderde metodes benodig om te kompenseer vir die skeefheid van roosters en nie-ortogonaliteit op die wande.

Gedurende die studie is dieper insig en kennis verkry oor die uitdaging om akkurate oplossings van hitte-oordrag op nie-ortogonale roosters te verkry. Hierdie kennis mag lei tot die algehele verbetering van die simulاسie van hitte-oordrag probleme en meer spesifiek vir die mikro-kanaal hitteruiler.

Sleutelwoorde: Nie-ortogonale roosters, eindige volume metode, mikro-kanaal hitteruiler, OpenFOAM

Abstract

The handling of gradient calculations on non-orthogonal meshes in the Finite Volume Method (FVM) is important in the modelling of complex geometries, since different implementation methods have an influence on the accuracy and the stability of the solution.

The application in the current study is the numerical solution of heat conduction in a complex geometry. It finds relevance in many engineering applications such as the Micro-Channel Heat Exchanger (MCHE) that acts as a recuperator in a High Temperature Reactor (HTR) power generation cycle.

A program based on the FVM was developed in Excel for the solution of the diffusion equation on a non-orthogonal mesh. A test case of heat conduction in a rectangular block, meshed with a tetrahedral mesh, was solved with the Excel code. The same test case was solved with OpenFOAM. The results of the two codes were compared. Small differences were found and their origins were traced to slightly different implementation methods. Knowledge of the differences in implementation between the two codes resulted in a better understanding of the aspects that influenced accuracy and stability.

Computations on meshes with the presence of mesh skewness and non-orthogonal mesh lines at boundaries were performed and an accompanying decrease in accuracy was observed. The results showed that the standard FVM as implemented in the Excel code and in OpenFOAM will need advanced methods to compensate for mesh skewness and non-orthogonality found at boundaries.

During the study, a deeper knowledge and understanding was gained of the challenge of obtaining accurate solutions of heat conduction on non-orthogonal meshes. This knowledge may lead to the overall improvement of the simulation of heat transfer models in general and for the MCHE specifically.

Keywords: Non-orthogonal meshes, finite volume method, micro-channel heat exchanger, computational fluid dynamics, OpenFOAM

Index

Opsomming.....	i
Abstract.....	ii
List of figures.....	vi
List of tables.....	ix
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Problem statement.....	5
1.3 Objective of study	5
1.4 Outline of study	6
Chapter 2 Literature study	8
2.1 CFD analysis stages	8
2.1.1 Pre-processing.....	9
2.1.2 Solving	9
2.1.3 Post-processing	9
2.2 Control Volumes	10
2.3 Orthogonal versus non-orthogonal meshes	11
2.4 Generation of meshes for complex geometries	12
2.4.1 Unstructured meshes	13
2.5 Solution techniques for non-orthogonal meshes	14
2.5.1 Minimum correction approach.....	16
2.5.2 Orthogonal correction approach	17
2.5.3 Over-relaxed approach.....	18
2.5.4 Comparison of the different non-orthogonality treatments	18
2.6 Boundaries.....	19
2.6.1 Boundary conditions	19
2.7 Alternative methods used for the calculation of gradients on non-orthogonal meshes.....	20
2.8 Error associated with non-orthogonal meshes	23
2.9 Summary	24
Chapter 3 Governing equations and discretisation	26
3.1 General transport equation	26
3.2 Diffusion Term.....	28

3.2.1	Diffusion coefficient	28
3.2.2	Face area vector	28
3.2.3	Gradient of ϕ at the face of the Control Volume	30
3.3	Source term	35
3.4	Numerical representation of the transport equation	35
3.4.1	Boundary Conditions	36
3.5	Summary	37
Chapter 4	Implementation in Excel and OpenFOAM	39
4.1	Introduction	39
4.2	Heat conduction case study	39
4.3	Implementation of the FVM in Excel	40
4.3.1	Mesh generation	40
4.3.2	Implementation	41
4.3.3	Solution	42
4.4	Implementation in OpenFOAM version 2.1	43
4.4.1	Data Structure	43
4.4.2	Mesh definition	44
4.4.3	Physical properties	45
4.4.4	Boundary and initial conditions	45
4.4.5	Specification of the files in the System directory	45
4.4.6	Viewing the mesh	46
4.4.7	Solution	47
4.5	Summary	47
Chapter 5	Results and discussion	48
5.1	Excel and OpenFOAM results	48
5.2	Comparison between the results of the calculations in Excel and OpenFOAM	50
5.2.1	Comparison of the coefficients for the internal faces based on the different non-orthogonality treatments in Excel with OpenFOAM	50
5.2.2	Coefficients of the internal and boundary faces used in the matrix for the calculation of the temperature in each CV	51
5.2.3	Gradient of ϕ over a Control Volume	53
5.2.4	Face area vectors	54
5.2.5	Volumes of Control Volumes	55

5.2.6	Interpolation factors	56
5.3	Implementing the different implementation methods in the new Excel model	59
5.4	Treatment of non-orthogonality and mesh skewness	60
5.4.1	Test for validity of non-orthogonal treatment.....	61
5.4.2	Mesh skewness.....	64
5.4.3	Comparison between the decomposition methods for non-orthogonal meshes	66
5.5	Summary	70
Chapter 6	Conclusions.....	71
Appendix A	- Excel and OpenFOAM calculations	77
Appendix B	- Excel and OpenFOAM results	94
Appendix C		100

List of figures

Figure 1: Recuperator used in a Brayton cycle from Pra et al., (2008:3161) where HP indicates the high pressure circuit, BP the low pressure circuit and MP the medium pressure circuit.....	2
Figure 2: MCHE plate with flow channels from Heatric (2012).....	3
Figure 3: Example of a MCHE with stacked plates to form primary and secondary sides from Kruger et al. (2008:2).....	3
Figure 4: Solving the heat conduction through a MCHE using CFD techniques modified from Kruger et al. (2008:8)	4
Figure 5: Illustration of the face, face area vector and the centre of a CV modified from Sezai (2011:22)	10
Figure 6: CV shapes where: (a) triangle, (b) tetrahedron, (c) quadrilateral, (d) hexahedron, (e) prism and (f) is a pyramid from Murthy & Mathur (2002:28) ..	11
Figure 7: Illustration of an orthogonal mesh modified from Sezai (2011:20).....	12
Figure 8: Illustration of a non-orthogonal mesh modified from Ubbink (2011:18)	12
Figure 9: Illustration of a triangular unstructured mesh from Versteeg and Malalasekera (2007:311)	13
Figure 10: Illustration of a hybrid unstructured mesh for the calculation of flow in a tube bank from Versteeg & Malalasekera (2007:312).....	14
Figure 11: Vertex-centred (a), circum-centred (b) and cell-centred (c) CVs from Date (2005:1118).....	14
Figure 12: Illustration of the non-orthogonal corrector term on a non-orthogonal mesh modified from Ubbink (2011:18).....	15
Figure 13: The "minimum correction approach" used for the treatment of non-orthogonality from Jasak (1996:85).....	17
Figure 14: The "orthogonal correction approach" used for the treatment of non-orthogonality from Jasak (1996:85).....	17
Figure 15: The "over-relaxed approach" for the treatment of non-orthogonality from Jasak (1996:85)	18
Figure 16: CV of a non-orthogonal mesh adjacent to a boundary from Jasak (1996:93)	20
Figure 17: Control-volume-based finite element method from Ferziger and Perić (2001:245).....	21

Figure 18: Illustration of different vectors from Versteeg and Malalasekera (2007:317).....	22
Figure 19: Skewness error on the face from Juretić & Gosman (2010:418)	23
Figure 20: Illustration of the face area vector modified from Apsley (2012:5).....	29
Figure 21: Illustration of fN and PN used in the calculation of the interpolation factors.....	31
Figure 22: A typical triangular control volume illustrating face centres	32
Figure 23: A triangular control volume with the centre point added.....	33
Figure 24: CV for the calculation of the volume of a CV	34
Figure 25: Domain with specified boundary conditions.....	39
Figure 26: Domain divided into non-orthogonal CVs	40
Figure 27: Case directory structure in OpenFOAM of a typical OpenFOAM case	43
Figure 28: Mesh generated in OpenFOAM	46
Figure 29: Temperature distribution through domain.....	48
Figure 30: Comparison between the Excel, OpenFOAM and analytical results	48
Figure 31: Comparison of the calculated coefficients based on the different non-orthogonal treatments with the method used in OpenFOAM	51
Figure 32: Comparison between the coefficients of the orthogonal components of the face area vectors between the Excel and OpenFOAM models.....	52
Figure 33: Comparison of the x-components of the gradients of ϕ over the CVs with regard to the Excel and OpenFOAM models that were used.....	53
Figure 34: Comparison of the y-components of the gradients of ϕ over the CVs between the Excel and OpenFOAM models.....	54
Figure 35: Comparison of the calculated magnitudes of the face area vectors between Excel and OpenFOAM	55
Figure 36: Comparison between the volumes of the CVs between the Excel and OpenFOAM models.....	56
Figure 37: Comparison of the interpolation factor, L_P between the Excel and OpenFOAM models for the internal faces of the domain.....	57
Figure 38: Comparison of the interpolation factor, L_N between the Excel and OpenFOAM models for the internal faces of the domain.....	57
Figure 39: Calculation of the interpolation factors in OpenFOAM.....	58

Figure 40: Comparing the temperature results for the new Excel model with the OpenFOAM results	59
Figure 41: Non-orthogonal hexahedral mesh	61
Figure 42: Temperature profiles on a non-orthogonal hexahedral mesh.....	62
Figure 43: Comparison between the analytical and OpenFOAM results of the temperatures for the CVs	62
Figure 44: Error in temperature gradient calculation near boundaries	63
Figure 45: Constant temperature gradient over the domain.....	64
Figure 46: Non-orthogonal mesh with mesh skewness	65
Figure 47: Distribution of the temperature gradient on a non-orthogonal mesh with mesh skewness.....	65
Figure 48: Linear temperature distribution in a rectangular block	66
Figure 49: Convergence behaviour of the different decomposition methods on a skew mesh.....	67
Figure 50: Convergence behaviour of the different decompositions on a skew mesh specifically for under-relaxation values of 0.9 and 0.8.....	67
Figure 51: Convergence behaviour of the different decomposition methods on a mesh with no skewness	68

List of tables

Table 1: Comparison between the Excel, OpenFOAM and analytical results	49
Table 2: Comparison between the results of the new Excel model and the results from OpenFOAM	60
Table 3: Node coordinates	77
Table 4: Owners and neighbours of a face.....	77
Table 5: Face area vectors.....	78
Table 6: Face centres	78
Table 7: CV centres	79
Table 8: Interpolation factors.....	79
Table 9: Volumes of CVs	80
Table 10: The value of ϕ at the CV face.....	80
Table 11: Gradient of ϕ over the CV - Owner.....	81
Table 12: Gradient of ϕ over the CV - Neighbour.....	81
Table 13: Gradient of ϕ at the face of a CV	82
Table 14: Orthogonal component of the face area vector for each of the faces	82
Table 15: Non-orthogonal corrector factor for each of the CV faces	83
Table 16: Non-orthogonal corrector term for each of the CV faces	83
Table 17: Neighbouring coefficient for each of the faces of a CV	84
Table 18: Calculation of the central coefficient.....	84
Table 19: Source term added to the matrix source	85
Table 20: Node coordinates specified in OpenFOAM	85
Table 21: Faces specified in the Faces file in the polyMesh subdirectory of the Constant directory in OpenFOAM.....	87
Table 22: Owners of faces specified in the Owners file in the polyMesh subdirectory of the Constant directory.....	89
Table 23: Neighbours of faces specified in the Neighbour file in the polyMesh subdirectory of the Constant directory.....	91
Table 24: Comparison of the calculated coordinates of the face area vectors between Excel and OpenFOAM	94
Table 25: Comparison of the magnitudes of the faces area vectors between Excel and OpenFOAM	95

Table 26: Comparison of the coordinates of the CV centres between Excel and OpenFOAM	96
Table 27: Comparison of the calculated volumes between Excel and OpenFOAM ...	97
Table 28: Comparison of the calculated interpolation factors between Excel and OpenFOAM	98
Table 29: Comparison between the coefficients of the Excel and OpenFOAM models	99

Nomenclature

Symbol Definition

a_{nb}	Neighbouring coefficients
a_p	Central coefficient
b	Boundary
C	Location of CV centre used in OpenFOAM
C_f	Location of the face centre used in OpenFOAM
$\underline{d}_f$	Distance vector between CV centres
$\underline{d}_n$	Distance vector between CV centre and the boundary in a direction normal to the boundary
$\underline{e}_\xi$	Unit vector in direction of ξ
f	Face of a CV
$\underline{fN}$	Distance vector between the centre of a face of a CV and the centre of the neighbouring CV
i	Face index in OpenFOAM
k	Thermal conductivity
$\underline{k}_f$	Non-Orthogonal Corrector Factor
L_N	Interpolation factor at CV N
L_P	Interpolation factor at CV P
$\underline{m}$	Skewness correction vector
$\underline{n}_f$	Unit normal vector at the face of a CV
n	Number of faces
N	CV Neighbour
N_v	Number of vertices
P	CV Owner
$\underline{PN}$	Distance vector between the centres of the owner and neighbouring CVs
S	Source term
S_u	Source term added to the matrix source
$\underline{S}_f$	Face area vector
S_p	Source term added to central coefficient

S_{ϕ}	Source term of field variable ϕ
$S_{\phi P}$	Volumetric source term of field variable ϕ
T	Temperature
$\underline{u}$	Velocity vector
V	Volume
V_P	Volume of control volume P
w	Weight factor
x	Cartesian x-coordinate
y	Cartesian y-coordinate
z	Cartesian z-coordinate

Mathematical Operators

$\frac{\partial}{\partial t}$	Partial derivative with respect to time
div	Divergence of a vector field
$\int_{CV} dV$	Integral over the volume of a CV
$\int_S dS$	Integral over a surface
$\sum_{f=1}^n$	Summation of property values over n faces
$\underline{\nabla}$	Del, geometric specific gradient operator

Greek Symbols

Γ	Diffusion coefficient
Γ_b	Diffusion coefficient at a boundary face
Γ_f	Diffusion coefficient at the face of a CV
μ	Viscosity
ρ	Density
ξ	Coordinate along the line joining P and A
ϕ	Field variable - Temperature
ϕ_b	Field variable at the boundary
ϕ_{d_n}	Field variable of a fixed gradient at the boundary
ϕ_f	Field variable at the face
ϕ_N	Field variable at the centre of the neighbour CV

ϕ_p	Field variable at the centre of the owner CV
Δ_f	Orthogonal component
$(\nabla\phi)_f$	Gradient of the field variable at the face of a CV
$(\nabla\phi)_N$	Gradient of the field variable over CV N
$(\nabla\phi)_P$	Gradient of the field variable over CV P
$(\nabla\phi)_b$	Gradient of the field variable over CV at the boundary

Acronyms

BP	Low Pressure
CAD	Computer Aided Design
CV	Control Volume
CFD	Computation Fluid Dynamics
D_{Cross}	Cross Diffusion Term
DIC	Diagonal Incomplete-Cholesky
FVM	Finite Volume Method
GT-MHR	Gas Turbine-Modular Helium Reactor
HTGR	High Temperature Gas Reactor
HP	High Pressure
MCHE	Micro-Channel Heat Exchanger
MP	Medium Pressure
OpenFOAM	Open source Field Operation And Manipulation
PCG	Preconditioned Conjugate Gradient

Chapter 1 Introduction

In the current study a heat conduction problem was solved using a numerical solution method. The numerical solution method applied was the Finite Volume Method (FVM) which is widely recognised in the engineering field for the simulation of fluid flows and heat transfer.

For complex geometries, numerical meshes of which the Control Volumes (CVs) are adjustable to the shape of the domain are used to model the geometry. These meshes are known as non-orthogonal meshes. The availability of non-orthogonal meshes allows for complex industrial flow and heat transfer problems to be solved. An example thereof is a Micro-Channel Heat Exchanger (MCHE) that acts as a recuperator in the gas cycle of a High Temperature Reactor (HTR).

The accurate modelling of heat conduction in the MCHE necessitates a thorough understanding of the treatment of non-orthogonal meshes and in particular gradient calculations. The focus of this study is to gain such an understanding of the aspects that influence the accuracy of modelling on non-orthogonal meshes.

By discretising a simple domain with a non-orthogonal mesh, the implementation methods of these meshes as well as the accuracy thereof can be investigated. The investigation was done by comparing the results obtained from a custom developed model, based on available literature, with results from the established Computational Fluid Dynamics (CFD) code OpenFOAM (Open source Field Operation And Manipulation), (Nebenführ, 2010:VII) and with the analytical solution.

1.1 Background

The use of a MCHE in a HTR such as the Gas Turbine-Modular Helium Reactor (GT-MHR) can improve the efficiency of the cycle by acting as an intermediate heat exchanger and preheating the helium at the core inlet.

GT-MHRs are based on the direct cycle concept where the primary coolant circuit of the nuclear core is the same circuit used as the driver circuit of the electric generator (Thonon & Breuil, 2000:151). The hot helium flowing from the outlet of the core flows directly through the turbine. Here it expands and before the helium returns to

the core, it is cooled before it is compressed. The electric generator is then driven by the turbine. Figure 1 is an illustration of the position of a recuperator in such a Brayton helium cycle.

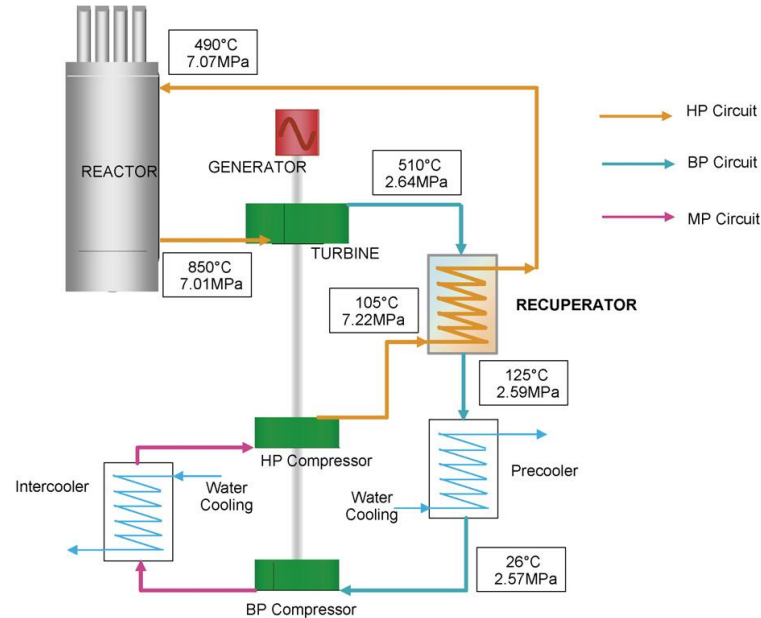


Figure 1: Recuperator used in a Brayton cycle from Pra et al., (2008:3161) where HP indicates the high pressure circuit, BP the low pressure circuit and MP the medium pressure circuit

These MCHEs have become a specific focus point of research due to their high heat transfer capabilities and the necessity for miniaturised light weight heat exchangers (Khan & Fartaj, 2011:553). MCHEs are up to 85% more compact than conventional heat exchangers (Heatric, 2012). The compactness of the MCHE is a very attractive feature since the recuperator and all of the other components, which are exposed to radioactive fluids, need to be installed inside the main containment building.

MCHEs consist of a large number of thin metal plate layers. These metal plate layers contain fluid channels that are etched in the surface as seen in Figure 2.

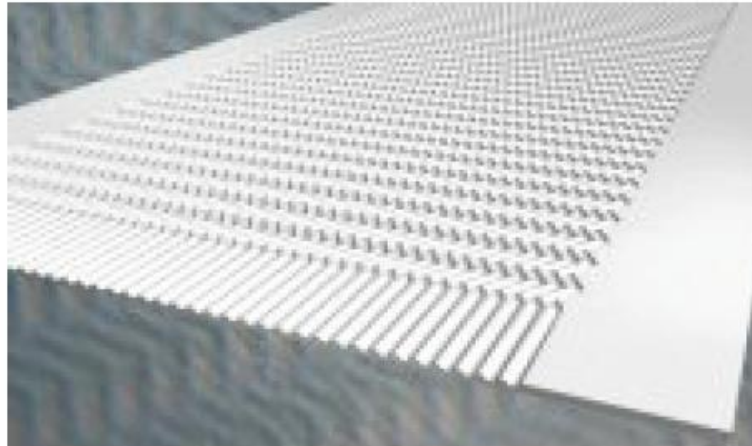


Figure 2: MCHE plate with flow channels from Heatric (2012)

The thin metal plates are stacked and diffusion bonded together to create a porous chunk of material containing the channels (Kruger et al., 2008:2). The plates are stacked in such a manner that the primary and secondary plates are alternated as seen in Figure 3.

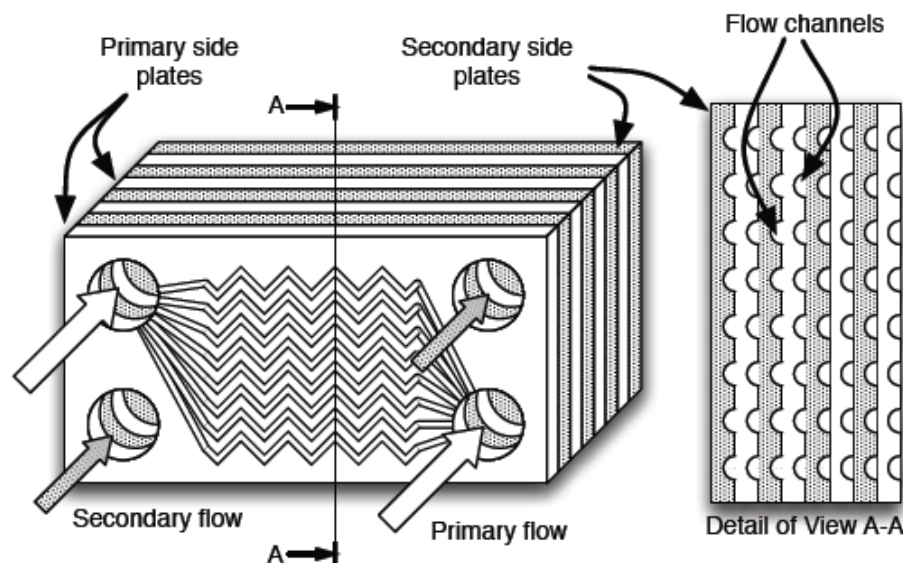


Figure 3: Example of a MCHE with stacked plates to form primary and secondary sides from Kruger et al. (2008:2)

The recuperator requires high efficiency characteristics and will operate under very harsh conditions such as high pressures and high temperatures (Pra et al., 2008:3161). In order to select the proper materials that should be used for the recuperator, it is necessary to accurately determine conditions such as the temperature distribution throughout the MCHE. Severe thermal gradients and cyclic loads can cause failures

due to thermal fatigue. However, since the design and construction of MCHEs are usually proprietary processes and fluid flow correlations for these conditions are not readily available in open literature, it is necessary to use alternative methods to determine these.

A Systems CFD approach was proposed for the modelling of these heat exchangers during operation (Kruger et al., 2008:2). The fluid flow inside the channels was modelled with one-dimensional empirical models and the heat distribution inside the solid core was modelled with three-dimensional CFD techniques for porous media. An effective porosity was calculated by tracing the micro channels through the CFD CVs and the connectivity between one-dimensional elements and three-dimensional CV volumes, were then determined. The coefficients of the one-dimensional and three-dimensional models were then linked implicitly through a connectivity stencil at matrix level in the solver (Kruger et al., 2008:2). The matrix solver then solves the integrated system of equations as a single system of unknowns.

An example of the temperature distribution through a MCHE solved with CFD techniques can be seen in Figure 4 below. The solution shows temperature profiles as a result of conduction heat transfer in the solid and convection heat transfer to internal micro channel flows.

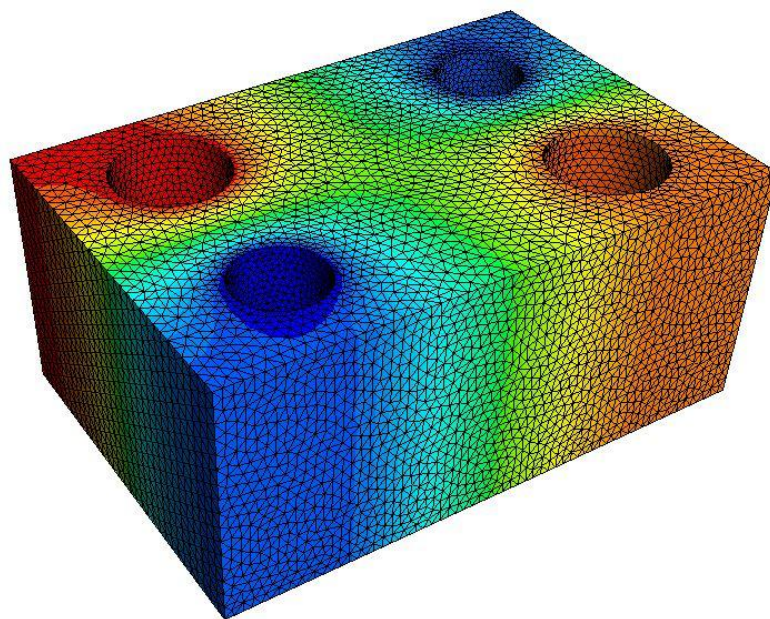


Figure 4: Solving the heat conduction through a MCHE using CFD techniques modified from Kruger et al. (2008:8)

1.2 Problem statement

The proper treatment of temperature gradients in the FVM is important in order to accurately model MCHEs used in the nuclear industry. This is also true for any other engineering application where heat transfer needs to be modelled in a complex geometry requiring non-orthogonal meshes.

Accuracy on orthogonal meshes regarding heat conduction is in general much easier obtained than on non-orthogonal meshes. First-order accuracy may be sufficient to compute a steady-state solution reliably and quickly on almost any mesh type and may be good enough to provide some guidance for product designs, but when the geometric accuracy of a solution is important, first-order methods are usually too crude (Wang, 2011).

Second-order accuracy is associated with the FVM on orthogonal meshes and also on some non-orthogonal meshes. The types of non-orthogonal meshes where second-order accuracy is not obtainable with the standard FVMs, are meshes with a high degree of mesh skewness (lines connecting the centres of neighbouring CVs do not intersect the centres of the dividing faces) and secondly those where mesh lines do not meet boundaries in an orthogonal manner. Non-orthogonal meshes therefore do in general introduce some form of a deterioration of accuracy.

The accurate modelling of heat transfer in a MCHE thus requires a thorough understanding of all the aspects that affect accuracy in a numerical modelling effort. This not only includes an understanding of the theory, but also discretisation practises and implementation details in computer codes. The challenge is therefore to obtain a clearer understanding by finding specific details through practical implementation of the numerical methods and by testing the resulting algorithms.

1.3 Objective of study

The objective of this study is to investigate the treatment of diffusion gradient calculations on non-orthogonal meshes using the FVM to gain insight into different implementation strategies that influences the accuracy of the solution. Both the literature (regarding the handling of non-orthogonal meshes) and existing software (such as OpenFOAM) will be considered to investigate and evaluate the

methodologies that are used for the treatment of diffusion gradient calculations on non-orthogonal meshes which are based on cell-centred CVs.

Two separate software packages were used during this study namely a self-developed FVM program and OpenFOAM. The intention of developing an FVM program specifically for this study is to ensure that the basic details of the theory and implementation methods are thoroughly understood.

OpenFOAM is a well-known open source FVM program that is used by analysts in the field of fluid and heat transfer modelling. The use of commercial CFD programs was not considered, because the in-code details of the exact implementation methods are proprietary.

The following sub-objectives were defined in order to properly consider the treatment of gradient calculations of non-orthogonal meshes during this study:

- Gain knowledge on the handling of non-orthogonal meshes based on a theoretical approach.
- Comparison of the differences in results obtained when performing calculations with a purpose developed code in Excel and OpenFOAM by solving a heat conduction problem on a non-orthogonal mesh.
- Understanding the different implementation methods of FVM gradient calculations based on the literature and as used in OpenFOAM; gaining insight regarding the accuracy of the different methods that are used.
- Demonstrate the effects of mesh skewness and non-orthogonality near the boundaries of a domain on solution accuracy.

1.4 Outline of study

The treatment of gradient calculations on non-orthogonal meshes for complex geometries is based on an investigation of the handling of non-orthogonal meshes in available literature and OpenFOAM. The outline of this study is divided according to the following chapter divisions

Chapter 2 presents the description or concept of a mesh and CVs. The difference between an orthogonal and a non-orthogonal mesh is described. The different methods for the handling of non-orthogonal meshes such as the minimum correction, orthogonal correction and over-relaxed approach based on published literature are described. The treatment of CVs adjacent to a boundary is also described.

Chapter 3 presents the partial differential equations that govern conduction heat transfer. The equations are integrated and discretised. The details of the discretisation process are given including the calculation of volumes, areas, interpolation factors, gradients and the treatment of boundary conditions.

Chapter 4 describes the specific case study that was used to compare the results of the Excel model and that of OpenFOAM. The methodologies followed to model the case study in Excel and OpenFOAM are described.

Chapter 5 presents the results from the Excel and OpenFOAM models. These results are compared with the analytical results and the differences, identified between the implementation methods, are discussed. The accuracy of solving a heat conduction problem, on a mesh with mesh skewness present and with non-orthogonal mesh lines at the boundary, is investigated. The three different methods for handling non-orthogonal meshes are also tested.

Chapter 6 presents a conclusion of the complete study by means of a summary of all the important findings. Finally, recommendations are made for further research.

Chapter 2 Literature study

The FVM is a popular and practical approach in the numerical solution of partial differential equations. It is relatively simple to understand as it uses physical concepts such as “fluxes of mass”, “momentum” and “energy” in the governing differential equations.

In the FVM the geometry that is considered is divided into a finite number of non-overlapping and contiguous CVs or cells (Ferziger & Perić, 1996:35). The transport equations are then integrated over each of these CVs, by approximating the variation of flow properties between the centres of the CVs with piecewise profiles. These piecewise profiles are constructed to support physical transport mechanisms, such as diffusion and convection. The discretised conservation equations for every CV in the mesh are assembled to form a system of linearly independent equations that can then be solved with a direct or iterative matrix solver.

The FVM can accommodate any type of mesh topology and is therefore suitable for the modelling of complex geometries (Loudyi et al., 2007:30). The complexity of the geometry plays an important role in the choice of mesh that would be used to model the geometry. Mesh generation and boundary specification play fundamental roles in the pre-processing stage of all CFD codes.

This section will include a brief description of the different CFD stages, the different types of meshes that can be used especially for the modelling of complex geometries, the different techniques used for the handling of non-orthogonal meshes and the handling of CVs adjacent to the boundaries in non-orthogonal meshes.

2.1 CFD analysis stages

A CFD analysis consists of three different stages (Apsley, 2012:1) namely:

- i. pre-processing;
- ii. solving and
- iii. post-processing.

2.1.1 Pre-processing

The pre-processing stage involves the transformation of the input variables of a flow problem into a form that is suitable for use in a solver. The necessary specifications at the pre-processing stage, involve the choice of the conservation equations to be solved as well as (Versteeg & Malalasekera, 2007:3):

- i. the definition of the geometry of the regions of interest;
- ii. the generation of a computational mesh;
- iii. the definition of fluid properties and the
- iv. specification of boundary conditions.

2.1.2 Solving

The solver is often operated as a “black box” in commercial CFD codes, however user intervention is necessary in order to adjust under-relaxation factors and error tolerances. The numerical algorithm involves the following steps:

- i. integration of the equations over all the CVs of the domain;
- ii. discretisation of the integral equations to obtain a system of algebraic equations;
- iii. solution of the algebraic equations by an iterative method and
- iv. updating the field variables for coupled problems.

CFD codes contain discretisation techniques for the treatment of the transport phenomena such as convection and diffusion as well as for the source terms that are used to indicate the creation or destruction of the general field variable ϕ . The conservation of ϕ in a CV is generally expressed as a balance between the various processes that tend to increase or decrease ϕ .

2.1.3 Post-processing

Post-processing involves the manipulation of the raw output values from the solver. Output values consist of a set of numbers corresponding to the values of each field variable, at each point of the mesh and are processed into a meaningful subset, to obtain the desired main output results. Commercial packages provide plotting tools for visualisation and analysis tools to extract and manipulate data for example:

- i. mesh and domain geometry display;
- ii. vector plots;

- iii. two-dimensional and three-dimensional surface plots and
- iv. view manipulation (rotation and scaling).

2.2 Control Volumes

The numerical mesh is a discrete representation of the geometric domain and it divides the solution domain into a finite number of non-overlapping sub-domains. These sub-domains are CVs that may consist of arbitrary shapes and sizes.

A domain divided into three CVs is illustrated in Figure 5. The CV is bounded by a set of flat faces indicated as f in the figure. CV faces are either classified as internal faces or as boundary faces. An internal face is the face between two CVs and a boundary face is a face that coincides with a boundary of the domain.

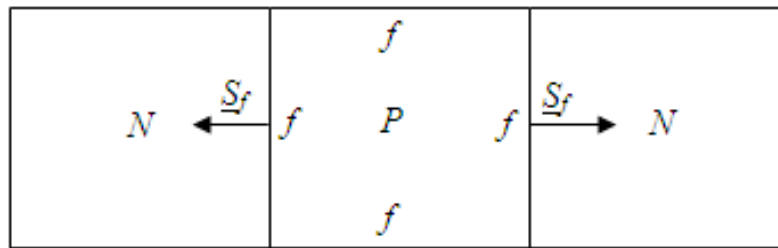


Figure 5: Illustration of the face, face area vector and the centre of a CV modified from Sezai (2011:22)

In the figure P indicates the centre of the “owned” CV and N the “neighbouring” CVs. A CV is either classified as “owned” by the face or as a “neighbour” of the face. If the face area vector $\underline{S}_f$ points outward from the CV, the CV is “owned” by the face or else the CV is a “neighbour” of the face. The face area vector is a vector with a magnitude that is equal to the area of the face and points outward from the “owned” CV in a direction that is normal to the face.

Meshes of various CV shapes may be constructed. Different CV shapes can be seen in Figure 6.

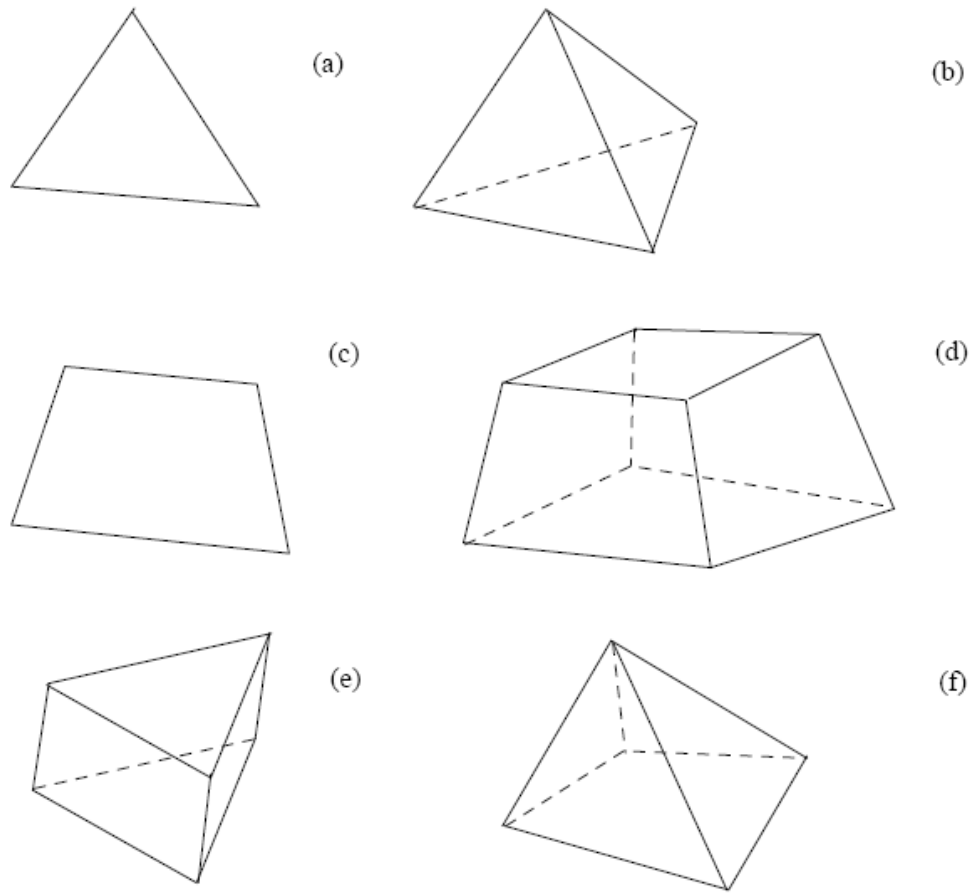


Figure 6: CV shapes where: (a) triangle, (b) tetrahedron, (c) quadrilateral, (d) hexahedron, (e) prism and (f) is a pyramid from Murthy & Mathur (2002:28)

Quadrilaterals can be used in two-dimensional structured meshes and hexahedra in the modelling of three-dimensional structured meshes (Apsley, 2012:2). Triangles and tetrahedra are commonly used for unstructured meshes that are necessary for the modelling of complex geometries (Murthy & Mathur, 2002:25). Arbitrary polyhedron CV shapes can also be used for the modelling of complex geometries.

2.3 Orthogonal versus non-orthogonal meshes

An orthogonal mesh is illustrated in Figure 7, where $\underline{S}_f$ indicates the face area vector and $\underline{d}_f$ is a line connecting the centres of the CVs.

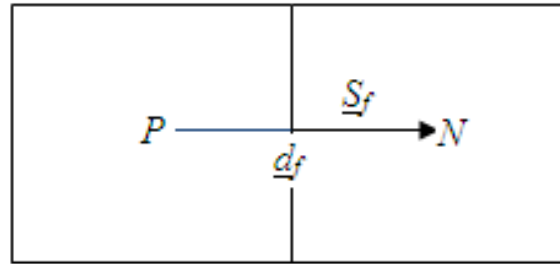


Figure 7: Illustration of an orthogonal mesh modified from Sezai (2011:20)

Non-orthogonal meshes are commonly used in the modelling of complex geometries. An example of a non-orthogonal mesh can be seen in Figure 8. A mesh is defined as non-orthogonal when the angle (α) between the face area vector, $\underline{S}_f$ and the line connecting the centres of the two neighbouring CVs $\underline{d}_f$, is not zero.

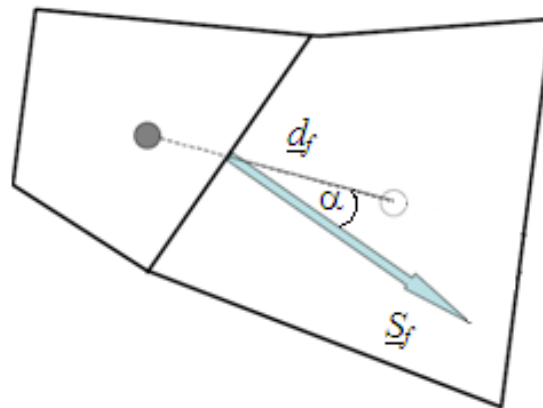


Figure 8: Illustration of a non-orthogonal mesh modified from Ubbink (2011:18)

The non-orthogonality of the mesh leads to the importance of a deeper investigation into the different types of meshes that can be used to model complex geometries and also into the handling of such non-orthogonal meshes.

2.4 Generation of meshes for complex geometries

When complex geometries are modelled with non-orthogonal meshes, it is important to generate the mesh as close to orthogonal as possible (Ferziger & Perić, 1996:211).

Mesh generation is very important, since the quality of the mesh plays an important role in the accuracy of the solution. There are many different commercial codes that can be used for the generation of meshes. The design of these codes is mainly aimed

at reducing the user interaction time and simultaneously at speeding up the mesh generation process. The generation of meshes that makes use of triangular and tetrahedral CV shapes is very popular due to easy automation.

For complex geometries, unstructured meshes are thus mainly used for the modelling of the geometry (Sezai, 2011:3).

2.4.1 Unstructured meshes

Unstructured meshes have the advantage that no implicit structure of co-ordinate lines are imposed on the mesh and thus mesh detail can be concentrated where necessary without wasting computer resources (Versteeg & Malalasekera, 2007:311). The CVs may have any shape and there is no restriction in the number of neighbouring CVs meeting at a point in two-dimensions or along a line in three-dimensions. Triangles or quadrilateral CVs in two-dimensions and tetrahedral or hexahedral CVs in three-dimensions are most often used in practice. Figure 9 is an illustration of an unstructured triangular mesh for the calculation of two-dimensional flow over an aerofoil.

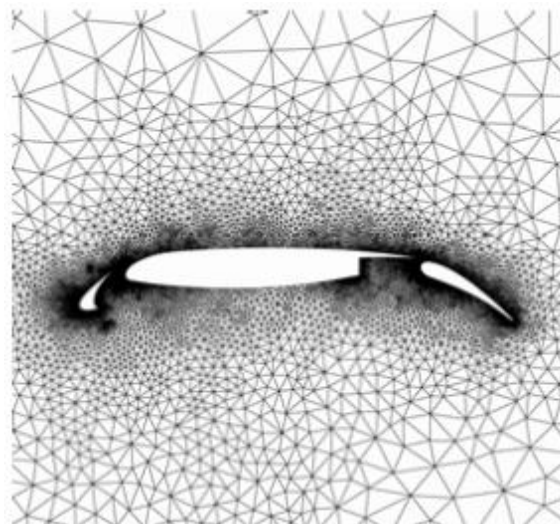


Figure 9: Illustration of a triangular unstructured mesh from Versteeg and Malalasekera (2007:311)

For unstructured mesh arrangements there are also no restrictions that determine the use of one particular CV type as it is possible to use a mixture of CV shapes. These meshes are called hybrid meshes. An illustration of such a hybrid mesh can be seen in Figure 10 that is used for the calculation of flow in a tube bank where the mesh is

made up of a combination of quadrilateral CVs that have been used near the solid walls and a triangular mesh structure elsewhere.

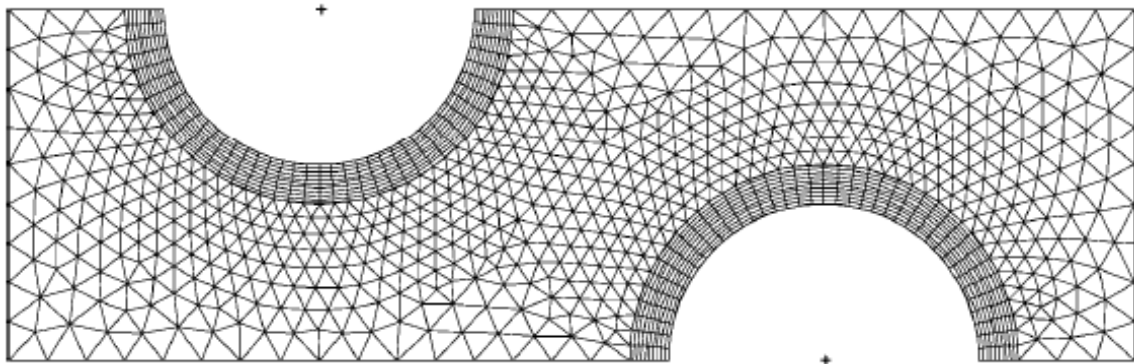


Figure 10: Illustration of a hybrid unstructured mesh for the calculation of flow in a tube bank from Versteeg & Malalasekera (2007:312)

2.5 Solution techniques for non-orthogonal meshes

There are three different ways of defining CVs in an unstructured mesh. These methods are vertex-centred, circum-centred and cell-centred CVs as seen in Figure 11.

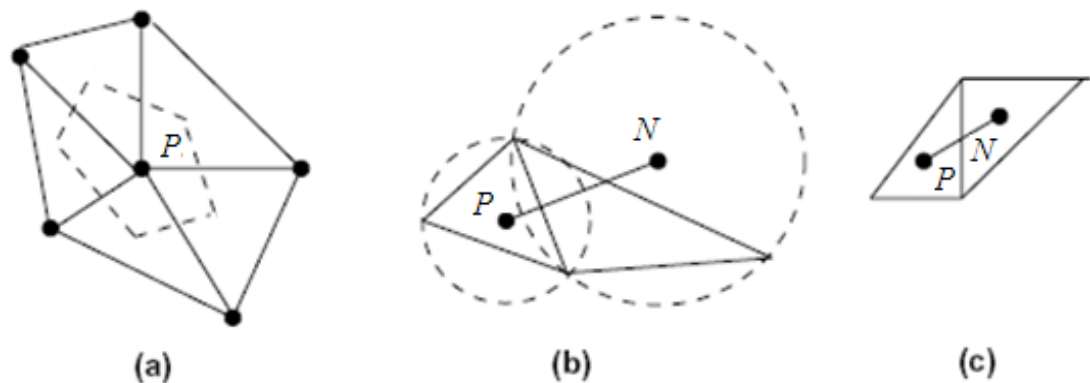


Figure 11: Vertex-centred (a), circum-centred (b) and cell-centred (c) CVs from Date (2005:1118)

In the vertex-centred method, the vertices are taken as nodes and the CV is constructed by the elements surrounding vertex P . The unknown solutions are stored on a per-vertex basis (Sezai, 2011:15). In general the node P will not be at the centroid of the CV, but the CV will always enclose the node. The line joining P to its neighbouring vertices will generally not be orthogonal to the CV faces (Date, 2005:1117).

The circum-centred approach defines the nodes at the circum-centre of each CV. This approach has the advantage that the CV face shared by two neighbouring CVs will

always be orthogonal to the line joining the circum-centres which are treated as the nodes, but a disadvantage is that the circum-centre of the CV may not lie within the CV as seen in Figure 11 (Date, 2005:1117).

The cell-centred approach will however be considered in this study. This approach is very popular, as it is a logical and more straight forward approach to use (Frink, 1991:10). In this approach the nodes are defined at the centroid of each CV. The cells serve directly as CVs that contain the unknown solutions stored on a per-cell basis.

A common approach for cell-centred calculations is where the non-orthogonality of the mesh is compensated for by defining a vector $\underline{k}_f$ as seen in Figure 12 and decomposing the face area vector into an orthogonal and a non-orthogonal corrector factor (Ubbink, 2011:18).

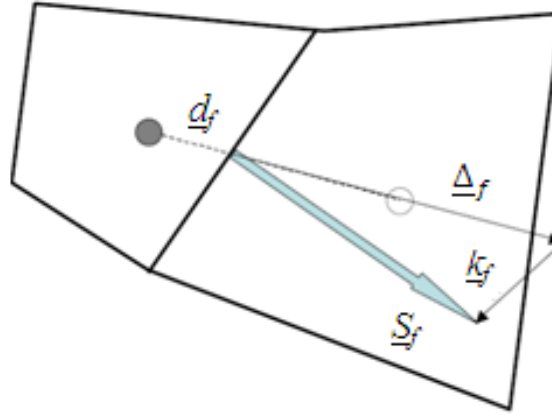


Figure 12: Illustration of the non-orthogonal corrector term on a non-orthogonal mesh modified from Ubbink (2011:18)

In the figure, $\underline{\Delta}_f$ is chosen as a vector parallel to $\underline{d}_f$ to express the face area vector as:

$$\underline{S}_f = \underline{\Delta}_f + \underline{k}_f \quad (2.1)$$

where $\underline{\Delta}_f$ is the orthogonal component and $\underline{k}_f$ is the non-orthogonal corrector factor. The orthogonal component and non-orthogonal corrector factor are then used in the calculation of the diffusion term that is represented by:

$$\sum_{f=1}^n \Gamma_f (\nabla \phi)_f \cdot \underline{S}_f = 0 \quad (2.2)$$

where:

Γ_f = diffusion coefficient at the face of the CV;

$(\nabla\phi)_f$ = the gradient of the field variable ϕ at the face of the CV;

The diffusion coefficient, the gradient of the field variable ϕ at the face of the CV as well as the face area vector will be described in more detail in Section 3.2, but the decomposition methods for the orthogonal component of the face area vector will be described in this section.

There are three main decomposition methods that exist, namely the minimum correction approach, orthogonal correction approach and the over-relaxed approach (Sezai, 2011-2012:24).

The decomposition of the face area vector into the orthogonal and non-orthogonal part determines the split between the implicit and explicit part of the diffusion term, where the implicit part is treated in the matrix and the explicit part is treated in the source term as described in Section 3.3. The split between the implicit and explicit part of the diffusion term will have an influence on the accuracy and convergence behaviour of the solution. The different non-orthogonality treatments will therefore be described in the following section.

2.5.1 Minimum correction approach

The minimum correction approach aims at keeping the non-orthogonal correction as small as possible by making $\underline{\Delta}_f$ and $\underline{k}_f$ orthogonal (Jasak, 1996:84) and calculating $\underline{k}_f$ with equation (2.1), where $\underline{\Delta}_f = |\underline{S}_f| \cos(\alpha) \underline{e}_{PN}$ and where $\underline{e}_{PN}$ is a unit vector in the direction of PN ($\underline{e}_{PN} = \frac{\underline{d}_f}{|\underline{d}_f|}$), which reduces to $\underline{\Delta}_f = (\underline{S}_f \cdot \underline{e}_{PN}) \underline{e}_{PN}$, resulting in equation (2.3) by considering that $|\underline{d}_f|^2 = \underline{d}_f \cdot \underline{d}_f$ (Stewart, 2003:807):

$$\underline{\Delta}_f = \frac{\underline{d}_f \cdot \underline{S}_f}{\underline{d}_f \cdot \underline{d}_f} \underline{d}_f \quad (2.3)$$

The minimum correction approach can be seen in Figure 13, where the vectors $\underline{\Delta}_f$ and $\underline{k}_f$ are orthogonal, keeping the non-orthogonal term to the minimum. The contribution of the orthogonal term decreases, as the non-orthogonality of the mesh increases.

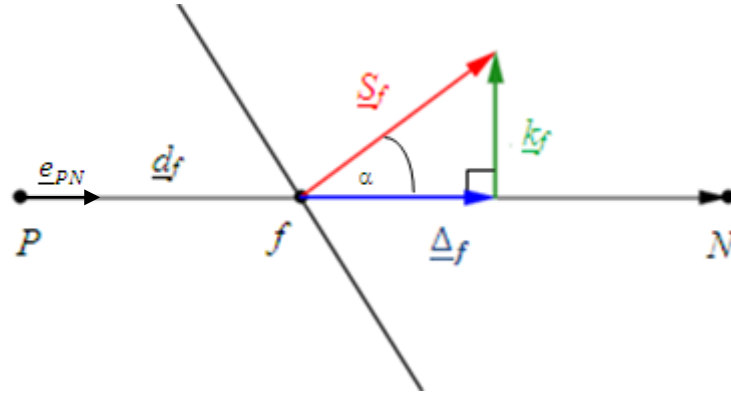


Figure 13: The "minimum correction approach" used for the treatment of non-orthogonality from Jasak (1996:85)

2.5.2 Orthogonal correction approach

The orthogonal correction approach (Jasak, 1996:85) is defined by equation (2.4).

$$\underline{\Delta}_f = \frac{\underline{d}_f}{|\underline{d}_f|} |\underline{S}_f| \quad (2.4)$$

Irrespective of the non-orthogonality, this approach aims at keeping $|\underline{S}_f|$ equal to $|\underline{\Delta}_f|$, as on an orthogonal mesh and attempts to maintain the condition of orthogonality irrespective of whether non-orthogonality exists (Menon, 2011:33). Figure 14 illustrates the decomposition of the face area vector for the orthogonal correction approach where $\underline{\Delta}_f = |\underline{S}_f| \underline{e}_{PN}$ and reduces to equation (2.4).

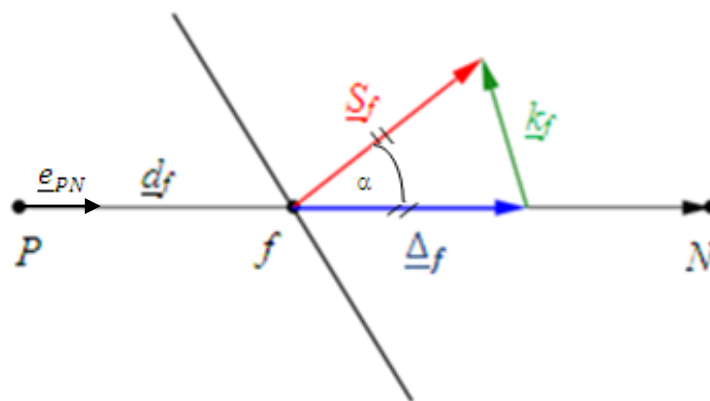


Figure 14: The "orthogonal correction approach" used for the treatment of non-orthogonality from Jasak (1996:85)

2.5.3 Over-relaxed approach

The over-relaxed approach (Jasak, 1996:85) is defined by equation (2.5).

$$\underline{\Delta}_f = \frac{\underline{d}_f}{\underline{d}_f \cdot \underline{S}_f} |\underline{S}_f|^2 \quad (2.5)$$

The importance of the orthogonal term is increased with the increase of non-orthogonality, as the vectors $\underline{S}_f$ and $\underline{k}_f$ are orthogonal. Figure 15 is used to illustrate the decomposition of the face area vector for this approach where $\underline{\Delta}_f = \frac{|\underline{S}_f|}{\cos(\alpha)} \underline{e}_{PN}$ and considering $\cos(\alpha) = \frac{\underline{S}_f \cdot \underline{e}_{PN}}{|\underline{S}_f| |\underline{e}_{PN}|}$ resulting in $\underline{\Delta}_f = \frac{\underline{S}_f \cdot \underline{S}_f}{\underline{S}_f \cdot \underline{e}_{PN}} \underline{e}_{PN}$ and reduces to equation (2.5).

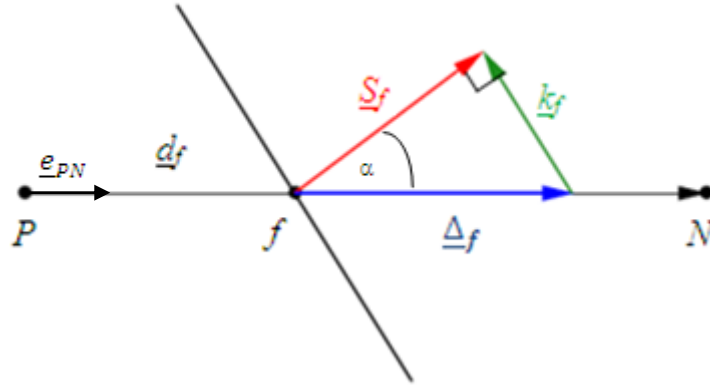


Figure 15: The "over-relaxed approach" for the treatment of non-orthogonality from Jasak (1996:85)

All of these approaches are valid, but they differ in accuracy and stability on non-orthogonal meshes. Accuracy refers to the correctness of the solution compared to the e.g. analytical solution, while stability again refers to the convergence of the iterative solution method (Murthy & Mathur, 2002:36). Unstability occurs when an iterative solution method fails to converge.

2.5.4 Comparison of the different non-orthogonality treatments

Jasak (1996:151) concluded that the over-relaxed approach has distinct advantages over the other two approaches in terms of stability. It also provides better convergence behaviour and allows higher angles of non-orthogonality. The reason why the over-relaxed approach performs better than the minimum and orthogonal correction approaches is due to the larger coefficient of the implicit term $\underline{\Delta}_f$ that is treated in the

matrix. As the importance of the orthogonal term is increased with the increase of non-orthogonality, the diagonal dominance of the coefficient matrix is enhanced. The enhanced diagonal dominance will result in better stability and convergence behaviour (Sezai, 2011:26). The condition for diagonal dominance will be given in Section 3.4.

The comparison of the different non-orthogonal treatments will also be performed as part of this study in order to gain further insight into their behaviour.

2.6 Boundaries

The treatment of gradients on the boundaries of a non-orthogonal mesh also requires special treatment.

In general the computational mesh includes a number of CV faces that are adjacent to a boundary. Boundary conditions are then used to describe the conditions at the boundaries.

2.6.1 Boundary conditions

Two basic types of numerical boundary conditions include the Dirichlet and Von Neumann boundary conditions. The Dirichlet boundary condition is used to prescribe a fixed value of ϕ at the boundary face, ϕ_b while the Von Neumann boundary condition prescribes a fixed gradient at the boundary, $(\nabla\phi)_b$ (Ubbink, 2011:12).

For a CV with a boundary face, the vector $\underline{d}_f$ that connects the centres of the CV with the neighbouring CV only extends to the centre of the boundary face. The boundary condition that is specified for the boundary face, indicated as b in Figure 16, is assumed to be valid along the whole area of the face (Jasak, 1996:93). This means that the value in the middle of the face is assumed to be valid along the whole face. The vector $\underline{d}_n$, between the centre of the CV and normal to the face, can then be defined and used in the calculations instead of the vector $\underline{d}_f$ that is normally used, which connects the centre of the CV with the centre of the face.

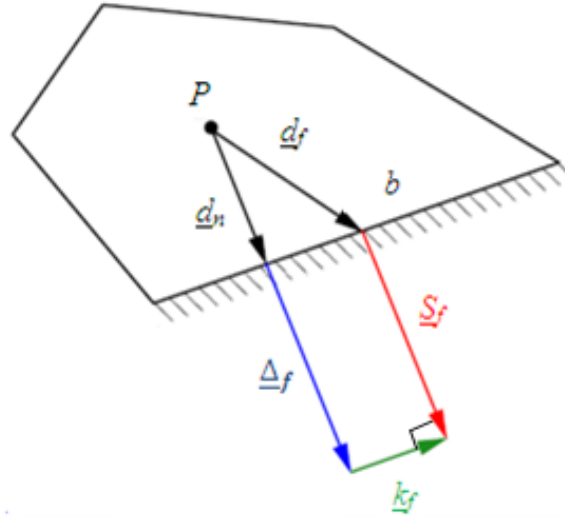


Figure 16: CV of a non-orthogonal mesh adjacent to a boundary from Jasak (1996:93)

As seen in the figure, the orthogonal component $\underline{\Delta}_f$, is equal to the face area vector $\underline{S}_f$, ($\underline{\Delta}_f = \underline{S}_f$) and is no longer located in the middle of the face. The vector $\underline{d}_n$ between the centre of the CV and normal to the boundary is then defined as:

$$\underline{d}_n = \frac{\underline{S}_f}{|\underline{S}_f|} \frac{\underline{d}_f \cdot \underline{S}_f}{|\underline{S}_f|} \quad (2.6)$$

It is used in the orthogonal term instead of $\underline{d}_f$ and the non-orthogonal corrector factor $\underline{k}_f$ is then not used. The Dirichlet boundary condition is expressed as:

$$\Gamma_b \underline{S}_f \cdot (\underline{\nabla} \phi)_b = \Gamma_b |\underline{S}_f| \frac{\phi_b - \phi_P}{|\underline{d}_n|} \quad (2.7)$$

and the Von Neumann boundary condition as:

$$\Gamma_b \underline{S}_f \cdot (\underline{\nabla} \phi)_b = \Gamma_b |\underline{S}_f| (\underline{\nabla} \phi)_b \quad (2.8)$$

where $(\underline{\nabla} \phi)_b$ is the gradient of ϕ at the boundary.

2.7 Alternative methods used for the calculation of gradients on non-orthogonal meshes

Alternative methods for the calculation of gradients on non-orthogonal meshes also exist. Ferziger and Perić (2001:232) describe how second-order accuracy can be

obtained on unstructured meshes if the values of the variable and the gradients are calculated at the CV centre by appropriate shape functions.

A control-volume-based finite element method also exist, which is a hybrid finite element/finite volume method that uses triangular elements and linear shape functions (Ferziger and Perić, 2001:245).

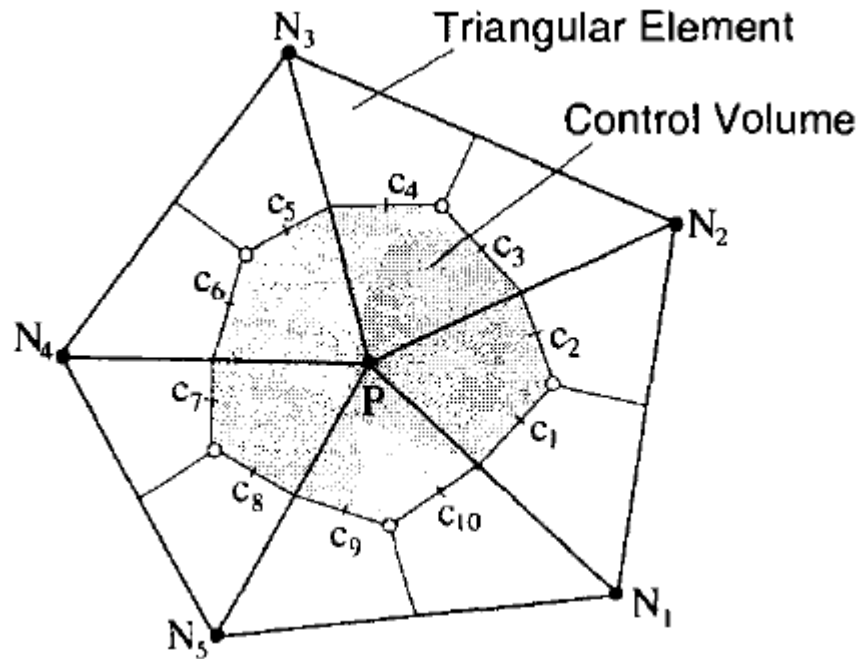


Figure 17: Control-volume-based finite element method from Ferziger and Perić (2001:245)

The solution domain is divided into triangular elements as shown in Figure 17. The variation of the variables is then described by the triangular elements. The computational nodes are located at the vertices of the triangles. The shape function for the linear variation of ϕ within an element is:

$$\phi = ax + by + c \quad (2.9)$$

where the coefficients a , b and c are determined by fitting the function to the nodal values at the vertices. As shown in Figure 17, around each node the CVs are formed by joining the centroids of the elements and midpoints on the edges of the elements. The conservation equations of the FVM that will be described in more detail in Chapter 3 are applied to these CVs. The surface and volume integrals are then easily calculated since an analytical function is used to describe the variation of variables over the triangular element. Node P (as shown in Figure 17) and the immediate

neighbours of the node (N_1 - N_5) are involved in the algebraic equation. The number of neighbours varies for the different CVs which may lead to an irregular matrix structure. The irregular matrix structure limits the range of solvers that can be used.

Versteeg and Malalasekera (2007:316) states that the most common form for the calculation of gradients to be corrected is by the introduction of a “cross diffusion” term ($\underline{S}_{D_{Cross}}$), similar to the non-orthogonal corrector term as described by Jasak (1996:84). The face area vector referred to as $\underline{A}$ in Versteeg and Malalasekera (2007: 319) for a non-orthogonal mesh is defined as:

$$\underline{A} = \underline{D} + \underline{S}_{D_{Cross}}, \quad (2.10)$$

where $\underline{D}$ is the orthogonal component:

$$\underline{D} = \frac{\underline{n} \cdot \underline{n}}{\underline{n} \cdot \underline{e}_\xi} \underline{e}_\xi \quad (2.11)$$

In equation (2.11), $\underline{n}$ is the unit normal vector and $\underline{e}_\xi$ is the unit vector in the direction of ξ , where $\Delta\xi$ is the distance between points P and A . Figure 18 is used to illustrate the different vectors.

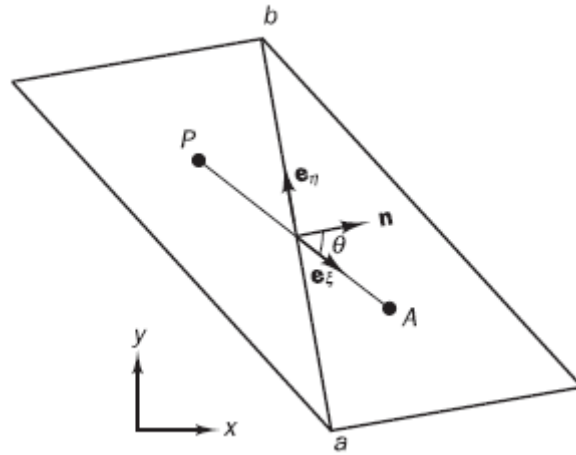


Figure 18: Illustration of different vectors from Versteeg and Malalasekera (2007:317)

Equation (2.11) is similar to the over-relaxed approach described by equation (2.5). However, the method described by Jasak (1996:84) provides an excellent description of the method used for the implementation of gradient calculations in OpenFOAM (Damián, 2013) and therefore the theory and definitions described by Jasak were studied in detail and implemented in Excel.

2.8 Error associated with non-orthogonal meshes

A typical error associated with non-orthogonal meshes is the skewness error. An illustration of the skewness error is seen in Figure 19. This error is a result of the calculation of face integrals that require the value of the field variable in the middle of the face, indicated as point f in the figure. The value of ϕ_f is obtained by linear interpolation between point P and N . The interpolation between these two points result in the value of ϕ at the point f_i which is not in the middle of the face and reduces the face integral to first-order accuracy, where $\underline{m}$ is the skewness correction vector (Jasak, 1996:124).

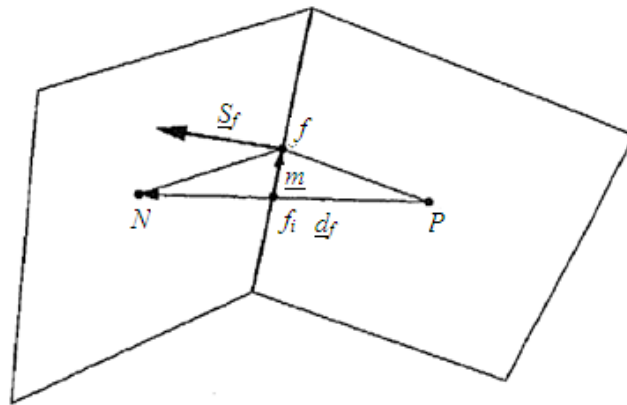


Figure 19: Skewness error on the face from Juretić & Gosman (2010:418)

Highly skewed meshes can decrease the accuracy of the solution. Remedies to improve the accuracy are the use of higher-order interpolation schemes and mesh refinement. The solution to the problem of decreased accuracy on highly skewed meshes in the FVM lies in the combined use of these two methods.

The advantage of using higher-order interpolation methods is offset by a few factors. Firstly, the use of higher-order interpolation (higher than second-order) usually results in a larger computational molecule to be used needing extra memory and more complex data structures. Furthermore the use of higher-order interpolation methods is often accompanied with the likelihood of stability problems. The stability problems originate from the fact that it is not always possible to implement the higher-order schemes in a complete implicit fashion as some terms end up as explicit source terms of the matrix solution which leads to less stable solutions.

Lilek & Perić (1995:240) reported that the use of higher-order interpolation schemes in the FVM is more often associated with the discretisation of convective terms. This implies that for a solution that contains convection and diffusion terms, the increase in overall solution accuracy is much more pronounced when higher-order schemes are applied to convection terms than to diffusion terms. Volume weighted interpolation for modelling convective transport is also a method that was reported (Coetzee, 2005:28) for successfully increasing solution accuracy.

Refinement of the mesh is an obvious way to increase the accuracy on skew meshes (Ferziger & Perić, 1996:77). Mesh refinement is however accompanied with longer solution times and increased memory requirements. The use of mesh refinement in the absence of higher than second-order interpolation schemes retains the simplicity and elegance of the implementation of the FVM.

2.9 Summary

In the literature study the definition of a non-orthogonal mesh was described. Non-orthogonal meshes are very popular for the modelling of complex geometries as the mesh can be concentrated where necessary and CVs can be of any shape. Different methods for the handling of non-orthogonal meshes on internal CV faces as well as a method for boundary faces were investigated.

Three main methods for the handling of non-orthogonal meshes were investigated. These three methods are the minimum correction approach, the orthogonal correction approach and the over-relaxed approach. Jasak (1996:143) concluded that these approaches differ in their convergence rate and solution stability.

The over-relaxed approach is preferred for the handling of non-orthogonal meshes as the approach reaches converged results even on meshes of which the non-orthogonality is so severe, that the solutions of the minimum correction and non-orthogonal correction approaches diverge.

The boundary CV faces of a non-orthogonal mesh also require special treatment. A vector $\underline{d}_n$ is defined which is a vector from the centre of the CV to the boundary face in a direction that is normal to the boundary. As the specified boundary condition is

assumed to be valid along the entire boundary face, $\underline{d}_n$ can be used in the calculation of the Dirichlet boundary condition instead of $\underline{d}_f$.

Alternative methods for the calculation of gradients on non-orthogonal meshes are available, but the method described by Jasak (1996:83) is the method best describing the method implemented in OpenFOAM.

The user should be aware that the skewness error is a typical error associated with non-orthogonal meshes.

In the following chapter, the discretisation of the general transport equation for a steady-state problem, in the absence of convection, will be discussed. The diffusion term will be integrated and by applying Gauss' divergence theorem, the volume integrals will be rewritten as integrals over a bounded surface. The flow domain will then be discretised into a finite number of non-overlapping CVs and by applying the Midpoint rule, a system of linear algebraic equations will be written in matrix form of $A\underline{\phi} = \underline{b}$, in order to solve these equations numerically.

Chapter 3 Governing equations and discretisation

The conservation laws are derived by the consideration of a given quantity of matter in a CV and the related extensive properties of the matter (Ferziger & Perić, 1996:3). The following conservation laws of physics are included in the governing equations (Bakker, 2012:2):

- i. conservation of mass;
- ii. conservation of momentum (equilibrium of forces) and the
- iii. conservation of energy.

These quantities are conserved over the whole solution domain for any number of mesh points.

3.1 General transport equation

The conservative form of all fluid flow equations can be written in the following form as the general transport equation for the conservation of the property ϕ in a CV:

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla \cdot (\rho\phi\mathbf{u}) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi \quad (3.1)$$

where:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\phi) &= \text{rate of increase of } \phi \text{ in fluid element;} \\ \nabla \cdot (\rho\phi\mathbf{u}) &= \text{net rate of flow of } \phi \text{ out of fluid element (convection);} \\ \nabla \cdot (\Gamma\nabla\phi) &= \text{net rate of increase of } \phi \text{ due to diffusion;} \\ S_\phi &= \text{net rate of increase of } \phi \text{ due to sources.} \end{aligned}$$

and ρ , $\mathbf{u}$ and Γ represents the density, velocity vector and the diffusion coefficient respectively.

In the absence of convection and for a steady-state problem, equation (3.1) reduces to:

$$\nabla \cdot (\Gamma\nabla\phi) + S_\phi = 0 \quad (3.2)$$

For a domain that is discretised into a finite number of non-overlapping contiguous CVs, with flat faces, the conservation equation of the property ϕ integrated over one of the CVs yields:

$$\int_{CV} \underline{\nabla} \cdot (\Gamma \underline{\nabla} \phi) dV + \int_{CV} S_{\phi} dV = 0 \quad (3.3)$$

The first volume integral can be rewritten as an integral over the bounding surface of the CV by applying Gauss' divergence theorem which states that for a vector $\underline{F}$:

$$\int_{CV} \underline{\nabla} \cdot \underline{F} dV = \int_{\partial V} \underline{F} \cdot d\underline{S} \quad (3.4)$$

where $d\underline{S}$ is a normal vector to the surface ∂V (Ubbink, 2011:7).

By considering equation (3.4), equation (3.3) can be written as:

$$\int_{\partial V} (\Gamma \underline{\nabla} \phi) \cdot d\underline{S} + \int_{CV} S_{\phi} dV = 0 \quad (3.5)$$

where:

$\int_{\partial V} (\Gamma \underline{\nabla} \phi) \cdot d\underline{S}$ = the net flux of fluid property ϕ across the boundaries due to diffusion;

$\int_{CV} S_{\phi} dV$ = the net rate of creation of the fluid property ϕ as a result of sources.

As the flow domain is discretised into a finite number of non-overlapping CVs with flat faces, the CV consists of a volume that is bounded by n flat faces. The surface integral can be decomposed as a sum of the integrals over the different faces that surround the CV and can be expressed as (Ubbink, 2011:8):

$$\sum_{f=1}^n \int_f (\Gamma \underline{\nabla} \phi) \cdot d\underline{S} + \int_{CV} S_{\phi} dV = 0 \quad (3.6)$$

where f denotes the CV face number. The integration midpoint rule can then be applied for continuous functions under the integrals, where $\underline{S}_f$ is the outward pointing face area vector, the subscript f indicates the face of the CV and P represents the CV number.

$$\sum_{f=1}^n (\Gamma \nabla \phi)_f \cdot \underline{S}_f + S_{\phi P} V_P = 0 \quad (3.7)$$

The values of the individual flow properties can be interpolated to the faces of the CV for continuous flow properties:

$$\sum_{f=1}^n \Gamma_f (\nabla \phi)_f \cdot \underline{S}_f + S_{\phi P} V_P = 0 \quad (3.8)$$

The diffusion term, source term and the numerical representation of the general transport equation are discussed in the following paragraphs.

3.2 Diffusion Term

The diffusion term is represented by equation (2.2) of which the diffusion coefficient, the face area vector and the gradient of the field variable ϕ at the face of the CV, are described in the following paragraphs.

3.2.1 Diffusion coefficient

The diffusion coefficient represents the thermal conductivity of the material and depends on the specific material used in the calculation region. For isotropic material properties, the diffusion coefficient at the centre of the CV is equal to the value at the face of the CV.

3.2.2 Face area vector

The face area vector for a two-dimensional face is illustrated in Figure 20:

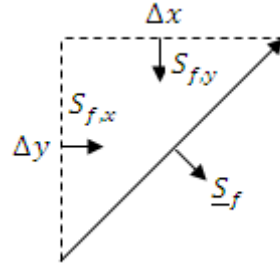


Figure 20: Illustration of the face area vector modified from Apsley (2012:5)

where $S_{f,x}$ and $S_{f,y}$ for a 2-dimensional mesh, are the projections of Δy and Δx respectively. This means that $S_{f,x}$ is equal to Δy and $S_{f,y}$ is equal to Δx , as seen in Figure 20, which represents the x and y component of the face area vector.

$\underline{S}_f$ is defined in equation (3.9).

$$\underline{S}_f = S_{f,x}\underline{i} + S_{f,y}\underline{j} \quad (3.9)$$

The magnitude of the face area vector is then calculated in equation (3.10).

$$|\underline{S}_f| = \sqrt{S_{f,x}^2 + S_{f,y}^2} \quad (3.10)$$

For non-orthogonal meshes, the face area vector is decomposed into an orthogonal component and a non-orthogonal corrector factor which is discussed in the following paragraph.

Non-orthogonal meshes

For non-orthogonal meshes the face area vector can be decomposed into an orthogonal component, $\underline{\Delta}_f$ and a non-orthogonal corrector factor as described in Section 2.5 (for an orthogonal mesh, $\underline{S}_f$ is equal to $\underline{\Delta}_f$).

By referring to equation (2.1), the orthogonal component $\underline{\Delta}_f$ is based on the approach followed for the handling of non-orthogonal meshes as described in Section 2.5 where: equation (2.3) represents the minimum correction approach, equation (2.4) the orthogonal correction approach and equation (2.5) represents the over-relaxed approach.

3.2.3 Gradient of ϕ at the face of the Control Volume

The gradient of ϕ at the face of a CV, $(\nabla\phi)_f$ is determined differently for orthogonal and non-orthogonal meshes as the computational molecule, resulting from the method used for the non-orthogonal component approach, is not a compact molecule existing only of the CV and its nearest neighbours. The skew and far neighbours are used to calculate the gradient over the CV. Thus it is difficult to treat this large molecule implicitly when considering the contributions of all these extra neighbours. Therefore the gradient of ϕ at the face of a CV is handled differently for the orthogonal and non-orthogonal component where equation (3.12) is used for the orthogonal component and equation (3.13) is used for the non-orthogonal component.

For non-orthogonal meshes considering the decomposition of the face area vector $\underline{S}_f \cdot (\nabla\phi)_f$ is:

$$\underline{S}_f \cdot (\nabla\phi)_f = \underline{\Delta}_f \cdot (\nabla\phi)_f + \underline{k}_f \cdot (\nabla\phi)_f \quad (3.11)$$

The gradient at the face for the orthogonal component can be solved implicitly, but the gradient for the non-orthogonal corrector term needs to be solved explicitly as it is needed to compensate for the non-orthogonal mesh's contribution.

Orthogonal component

The gradient of ϕ at the face of a CV, $(\nabla\phi)_f$ for the orthogonal component is calculated similar to the gradient of ϕ at the face for an orthogonal mesh.

The gradient at the face for the orthogonal component is calculated by considering the difference between the field variable ϕ at the centre of the owner and the neighbouring CV of the face as in the following equation:

$$\underline{\Delta}_f \cdot (\nabla\phi)_f = |\underline{\Delta}_f| \frac{\phi_N - \phi_P}{|\underline{d}_f|} \quad (3.12)$$

where $|\underline{\Delta}_f| = \frac{\Delta_f}{n_f}$, $|\underline{d}_f| = \frac{d_f}{n_f}$, and the subscripts N and P indicate the centres of the neighbour and owner CVs .

Non-orthogonal corrector term

The gradient at the face for the non-orthogonal corrector term is calculated by considering interpolation factors and interpolating the individual gradients of the field variable ϕ over the owner and neighbouring CVs to the face connecting these CVs:

$$(\underline{\nabla}\phi)_f = L_P(\underline{\nabla}\phi)_P + L_N(\underline{\nabla}\phi)_N \quad (3.13)$$

where L_P and L_N are the interpolation factors and $(\underline{\nabla}\phi)$ is the gradient over the CV calculated for the owner and neighbouring CVs (Jasak, 1996:84).

Interpolation factors

The interpolation factor L_P is the ratio of the distance between the centre of the face and the centre of the neighbouring CV (indicated by $\underline{fN}$), to the distance between the centre of the owner and the centre of the neighbouring CV (indicated by $\underline{PN}$).

$$L_P = \frac{|\underline{fN}|}{|\underline{PN}|} \quad (3.14)$$

$$L_N = (1 - L_P) \quad (3.15)$$

Figure 21 is used for the illustration of $\underline{fN}$ and $\underline{PN}$. In order to determine $\underline{fN}$ and $\underline{PN}$, it is necessary to determine the face centre of each of the faces of a CV as well as the centres of the CVs.

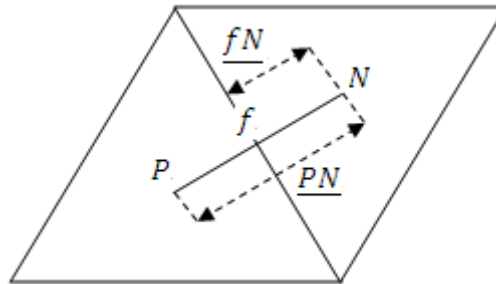


Figure 21: Illustration of $\underline{fN}$ and $\underline{PN}$ used in the calculation of the interpolation factors

Face centres

Consider a typical triangular (two-dimensional) CV as shown in Figure 22 below where f_1 indicates face 1 of the CV.

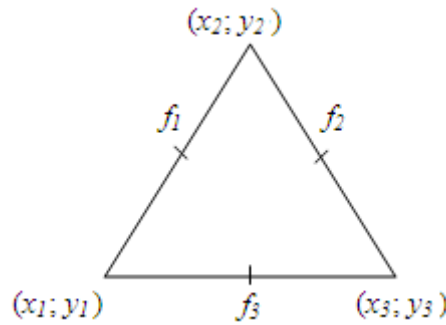


Figure 22: A typical triangular control volume illustrating face centres

The centres of the faces are calculated by averaging the coordinates of the vertices of the face where $(x_c)_{f_1}$ indicates the centre x-coordinate of face 1, N_v the number of vertices (where $N_v = 2$ for a 2-dimensional mesh) and $(y_c)_{f_1}$ the centre y-coordinate of face 1.

$$(x_c)_{f_1} = \frac{1}{N_v} \sum_{i=1}^{N_v} x_i \quad (3.16)$$

$$(y_c)_{f_1} = \frac{1}{N_v} \sum_{i=1}^{N_v} y_i \quad (3.17)$$

A similar approach is followed for the calculation of the centres of faces f_2 and f_3 .

Control Volume centres

Consider the triangular CV in Figure 23 below. The CV is similar to the one shown in the previous figure with the addition of its centre point.

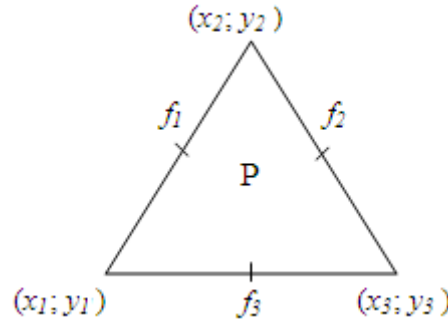


Figure 23: A triangular control volume with the centre point added

The geometric centre is calculated by summing the face centre points and then dividing by the number of faces (n).

$$P_x = \frac{1}{n} \sum_{i=1}^n (x_c)_{f_i} \quad (3.18)$$

$$P_y = \frac{1}{n} \sum_{i=1}^n (y_c)_{f_i} \quad (3.19)$$

The centre of the triangle is indicated by point P (see Figure 23), where P_x and P_y are the x - and y -coordinates of point P .

Gradient of ϕ over a Control Volume

The gradient over a CV, $(\nabla \phi)$ is calculated by applying Gauss' divergence theorem as stated by equation (3.4) on $(\nabla \phi)_P$ (Sezai, 2011:27):

$$(\nabla \phi)_P = \frac{1}{V_P} \sum_f \underline{S}_f \phi_f \quad (3.20)$$

where V_P is the volume of the CV and $\sum_f \underline{S}_f \phi_f$ is the sum of the products of the face area vector and the field variable ϕ_f at each of the faces surrounding the CV.

The volume of a CV is calculated using the face area vectors and the centres of the faces. This will be discussed in the following paragraph.

Volume of a Control Volume

Consider the CV illustrated in Figure 24 where the face centres and face area vectors are indicated for each of the faces.

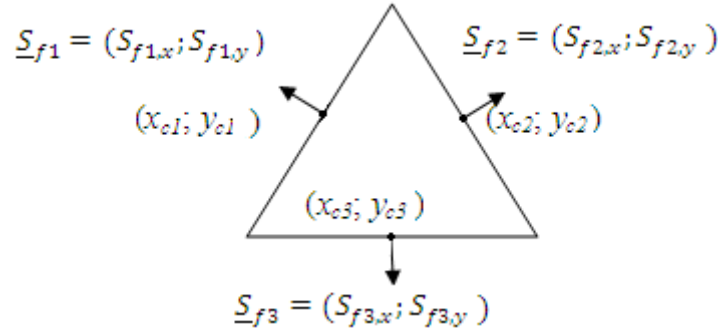


Figure 24: CV for the calculation of the volume of a CV

The volume of a CV is calculated by applying Gauss' theorem on the identity $1 = \text{div}(\underline{x}_i)$ as $\text{div}(\underline{x}_i) = \frac{\partial x}{\partial x} = 1$ (Sezai, 2011:18):

$$V = \int_V 1 dV = \int_V \text{div}(\underline{x}_i) dV = \int_S \underline{x}_i \cdot \underline{n} dS \approx \sum_f x_f S_{f,x} \quad (3.21)$$

where the volume of the CV is calculated by the summation of the product of the x coordinate of the face centre, x_f and the x component of the face area vector $S_{f,x}$ of each of the CV faces:

$$V = \sum_{i=1}^n x_{ci} S_{fi,x} \approx \sum_{i=1}^n y_{ci} S_{fi,y} \quad (3.22)$$

After calculating the volume, the sum of the products of the face area vector and the field variable ϕ_f still needs to be calculated to determine the gradient of ϕ over the CV.

Sum of the products of the face area vector and the field variable ϕ_f

The following equation should be considered in the sum of the products of the face area vector and the field variable ϕ_f , as there can be distinguished between “owned” and “neighbouring” faces (Jasak, 1996:80). A face is classified as “owned” by the CV

if the face area vector points outward from the CV otherwise it is considered a “neighbouring” face:

$$\sum_f \underline{S}_f \phi_f = \sum_{owner} \underline{S}_f \phi_f - \sum_{neighbour} \underline{S}_f \phi_f \quad (3.23)$$

where ϕ_f is determined by interpolating the field variable ϕ at the centre of the owner and neighbouring CVs to the centre of the face.

$$\phi_f = L_P \phi_P + L_N \phi_N \quad (3.24)$$

The non-orthogonal corrector term is handled in the source term. This will be described in the following section.

3.3 Source term

The non-orthogonal contributions and the conditions at the boundary are discussed in general in relation to the source term. The source term may be a function of the dependant variable ϕ and therefore a linear approximation of the term $S_{\phi P} V_P$ is expressed as:

$$S_{\phi P} V_P = S_u + S_P \phi_P \quad (3.25)$$

S_u is the term added to the matrix source and S_P is the source added to the central coefficient a_P . The central coefficient will be described in the following section.

3.4 Numerical representation of the transport equation

The numerical form of the general transport equation is written as:

$$a_P \phi_P = \sum_{nb=1}^n a_{nb} \phi_{nb} + S_u \quad (3.26)$$

Here nb indicates the neighbours of the CV, a_P is the central coefficient and a_{nb} indicates the neighbouring coefficients, where a_P and a_{nb} are:

$$a_p = \sum_{nb=1}^n a_{nb} - S_p \quad (3.27)$$

$$a_{nb} = \frac{\Gamma_f |\underline{\Delta}_f|}{|\underline{d}_f|} \quad (3.28)$$

Equation (3.26) should be written for each of the CVs of the solution domain to form a system of linear algebraic equations. The system of linear algebraic equations can be solved numerically in matrix form:

$$A\underline{\phi} = \underline{b} \quad (3.29)$$

where A is the $N \times N$ coefficient matrix and $\underline{\phi}$ and $\underline{b}$ are the column vectors of the unknown field variable ϕ and the source respectively. Diagonal dominance in a matrix is achieved if $|a_p| > \sum_{nb=1}^n |a_{nb}|$ is satisfied.

3.4.1 Boundary Conditions

The discretised equations are modified at the boundaries of a domain in order to introduce the boundary conditions. There are two main approaches found in boundary conditions namely the Dirichlet and Von Neumann boundary conditions.

Dirichlet boundary condition

The Dirichlet boundary condition indicates a fixed value at the boundary face as described in Section 2.6:

$$\phi_f = \phi_b \quad (3.30)$$

where b indicates the boundary. The source terms are modified as follows:

$$S_p = -\frac{\Gamma_b |\underline{S}_f|}{|\underline{d}_n|} \quad (3.31)$$

$$S_u = \frac{\Gamma_b |\underline{S}_f|}{|\underline{d}_n|} \phi_b \quad (3.32)$$

Von Neumann boundary condition

The Von Neumann boundary condition represents a fixed gradient at the boundary and is expressed as:

$$(\nabla \phi)_b = \phi_{d_n} \quad (3.33)$$

the source term S_u is then influenced as follows:

$$S_u = \Gamma_b \phi_{d_n} |\underline{S}_f| \quad (3.34)$$

3.5 Summary

In this chapter the general theory related to the FVM for non-orthogonal meshes was described. In the absence of convection and for steady state problems the general transport equation consists only of a diffusion term and a source term.

The diffusion term consists of the diffusion coefficient, a face area vector and the gradient of ϕ at the face of a CV. The diffusion coefficient represents the thermal conductivity of the domain, while the face area vector is a vector in a direction that is perpendicular to the face with a magnitude equal to the area of the face. For non-orthogonal meshes the face area vector is decomposed into an orthogonal component and non-orthogonal corrector factor. The orthogonal component is calculated based on the minimum correction, orthogonal correction or over-relaxed approach discussed in the literature study.

The gradient of ϕ at the face of a CV is treated differently for the orthogonal component and for the non-orthogonal corrector term as the orthogonal component is solved implicitly and the non-orthogonal corrector term is solved explicitly. The necessary calculations such as the interpolation factors, CV centres, face centres and the volumes of CVs for the determination of the gradients were discussed.

The non-orthogonal corrector term and the conditions at the boundaries of the domain are handled in the source term. The Dirichlet and Von Neumann boundary conditions were described as well as the manner in which the source terms are modified by the boundary conditions.

The theory, as described, was implemented in an Excel model to determine the temperature distribution through a domain discretised with a non-orthogonal mesh. Results from this model will then be compared with results from the CFD code OpenFOAM to determine any differences resulting from variances in the implementation methods of non-orthogonal meshes.

Chapter 4 Implementation in Excel and OpenFOAM

4.1 Introduction

This chapter starts with the definition of a heat conduction case study, which was used for both the Excel program and OpenFOAM. Then it looks at how the solution of the heat conduction equation, as described in Chapter 3, was implemented in Excel. The same case study was then also solved in OpenFOAM. Finally, the operational philosophy of the two codes is described to conclude the chapter.

4.2 Heat conduction case study

Consider a simple domain with dimensions of 2.1 x 1.5 m that also includes the following boundary conditions as seen in Figure 25:

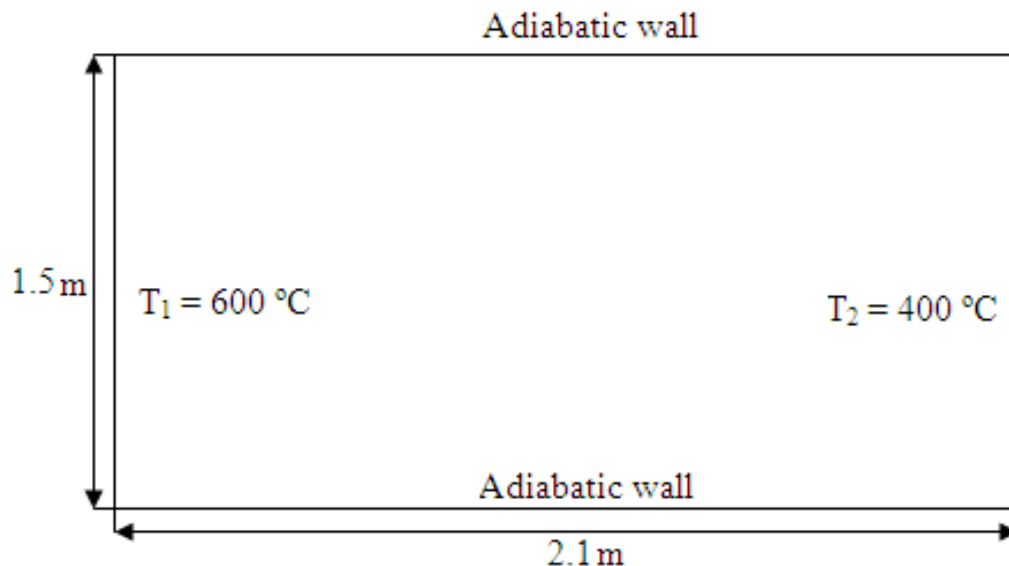


Figure 25: Domain with specified boundary conditions

Assumptions

The following assumptions were made:

- the initial temperatures of the CVs were 500°C;
- constant thermal conductivity of 1 W/m·K;
- no time dependant effects were modelled.

4.3 Implementation of the FVM in Excel

The reason for implementing the FVM in Excel instead of using a classic programming language was to present the entire calculation details on a single view. With a conventional program, one has to compile, run and use debug statements in order to see the details of the calculations. With the Excel model the effect of changing a single parameter in the implementation can be immediately observed and such a setup comes in handy in a research environment. A drawback of having the program in Excel is that the mesh configuration and resolution cannot be changed easily.

4.3.1 Mesh generation

The domain was divided into a number of non-orthogonal and non-overlapping contiguous CVs as seen in Figure 26 which is an illustration of the domain. **Note that for the Excel calculations**, a two-dimensional representation of the domain was modelled, while for OpenFOAM a full three-dimensional mesh was created (but quasi-two-dimensional).

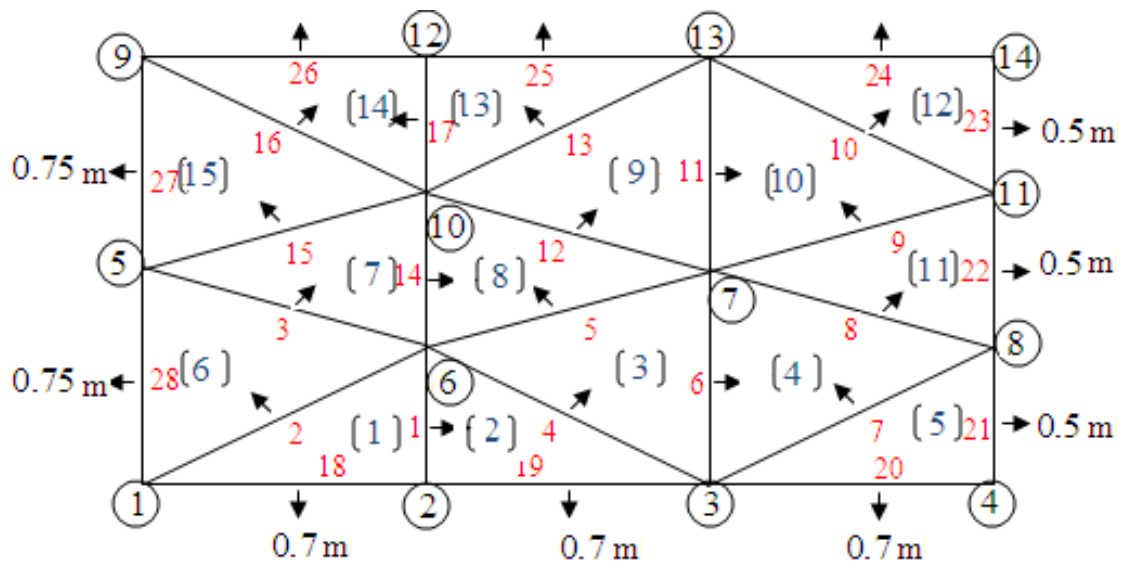


Figure 26: Domain divided into non-orthogonal CVs

The nodes, internal faces, boundary faces and CVs were numbered and the face area vectors for each of the internal and boundary faces are indicated in the figure. The owner and neighbour of a face were determined by whether the face area vector pointed outwards or inwards into a CV as described in Section 2.2. In the Excel

calculations, it is ensured that the distance between the centres of the CVs is based on the calculation where the line connecting the CV centres, intersects through the centre of the face, between the neighbouring CVs, ensuring that the skewness error is not present in the current calculations.

4.3.2 Implementation

The theory described in Chapter 3 was implemented in an Excel spreadsheet to determine the temperatures at the centres of each of the CVs. The detail of the implementations can be found inside the spreadsheet, which is included on a physical compact disc in Appendix C. A summarised description of the implementation is given below and only the first few entries of each table in Excel is given in Appendix A, as the complete Excel spreadsheet can be viewed on the physical compact disk.

A number of calculations were performed to determine the central and neighbouring coefficients in order to form the system of algebraic equations that were implemented in matrix form. This was done to solve the temperatures at the centres of each of the CVs based on the minimum correction, orthogonal correction and over-relaxed approaches.

In the Excel model, the node coordinates of each of the nodes indicated in Figure 26 were specified as seen in Table 3 in Appendix A. These coordinates were used to both determine the face area vectors described in Section 3.2.2 and the centres of the faces of a CV. After the centres of the faces were calculated, the CV centres were calculated as described in Section 3.2.3. The distances between CV centres and the face centres were then calculated subsequently.

The orthogonal component was then calculated based on the different approaches as described in Section 2.5. As soon as the orthogonal component was known, the non-orthogonal corrector factor could be calculated as seen in Table 15 in Appendix A.

The interpolation factors were calculated hereafter. Some of the values can be seen in Table 8 in Appendix A, while the complete list of values is available in the Excel spreadsheet in the compact disk. The interpolation factors were used in the calculation of the gradient of ϕ at the face of a CV for the non-orthogonal corrector term.

The non-orthogonal correction factors were calculated based on the assumed initial temperatures of 500°C for each of the CVs and interpolated to the faces. This led to the calculation of the non-orthogonal corrector terms as the dot-product of the non-orthogonal corrector factors and the gradients at the faces of the CVs.

The non-orthogonal corrector term also involved the calculation of the gradient of ϕ over a CV. This was calculated by using the volume of the CV and the sum of the products of the face area vectors as well as the value of ϕ at the faces as described in Section 3.2.3 and seen in Table 16.

The central and neighbouring coefficients were then determined as seen in Table 17 and Table 18 in Appendix A. This resulted into a system of algebraic equations before it was rearranged in matrix form.

The non-orthogonal correction factors were treated in the source term together with the fixed temperature boundary conditions of 600°C and 400°C as seen in Table 19 in Appendix A.

4.3.3 Solution

After calculating the central and neighbouring coefficients, the system of linear algebraic equations was solved numerically in matrix form by referring to equation (3.29):

$$\underline{\phi} = \underline{b} A^{-1} \quad (4.1)$$

The solution was then obtained through an iteration process until the solution converged. Convergence was measured as the difference between consecutive solutions of the temperature field from iteration to iteration with a maximum residual of 1×10^{-13} . These calculations can be found in the Excel spreadsheet on the attached CD in Appendix C.

In the following section, the steps needed to solve the standard case in OpenFOAM are described.

4.4 Implementation in OpenFOAM version 2.1

OpenFOAM is an open source software program that can be used to model various kinds of physical phenomena. The program is based on the FVM and implemented in an object-orientated way using C++. After more than a decade, the program (that was originally developed in the nineties) is now used by a very wide user base that includes students, analysts and engineers.

Simulation of the same case in OpenFOAM involved: the data for the mesh, fields, properties and control parameters. This data were stored in OpenFOAM in a set of files within the case directory. The following descriptions are based on an inspection of actual OpenFOAM code and the user manuals (The OpenFOAM Foundation, 2011- 2012).

4.4.1 Data Structure

The directory structure in OpenFOAM to run a typical case can be seen in Figure 27:

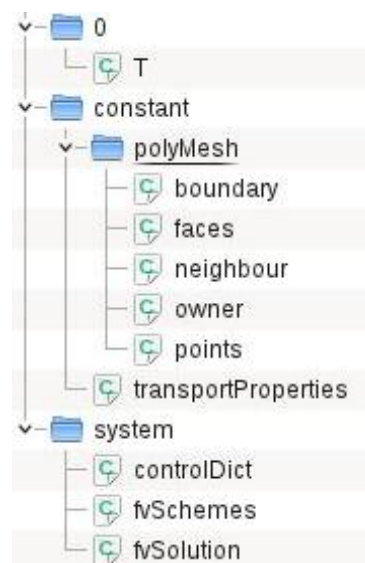


Figure 27: Case directory structure in OpenFOAM of a typical OpenFOAM case

the three main directories that can be found in the root are:

- i. “0” (Time directory);
- ii. “constant” and
- iii. “system”.

Time directory

The time directory contains individual files of data that are either the initial values or boundary conditions or the results written to file by OpenFOAM.

Constant directory

The constant directory consists of a polyMesh subdirectory and a transportProperties dictionary file. In the polyMesh subdirectory the domain and mesh is defined while the diffusion coefficient, k , is specified in the transportProperties dictionary.

System directory

The system directory contains three dictionary files namely the controlDict (where the control parameters such as the start or end time are set), the fvSchemes (where the discretisation schemes used may be selected) and the fvSolution dictionary (where tolerances and other algorithm controls are set).

4.4.2 Mesh definition

A domain and mesh are defined in OpenFOAM in the polyMesh subdirectory of the Constant directory, which contains the following files:

- i. Points;
- ii. Faces;
- iii. Owner;
- iv. Neighbour and
- v. Boundaries.

Each of these files is described in Appendix A and can be seen in Table 20, Table 21, Table 22 and Table 23 respectively. A description of the Boundaries file is also given in Appendix A.

By defining the CVs in this structure, the CVs can have an unlimited number of faces and for each face there is no limit to the number of edges or any limit to the alignment of the face (The OpenFOAM Foundation, 2011- 2012). The use of such a polymesh offers great freedom in the generation of meshes.

4.4.3 Physical properties

The conductivity, k of the domain was specified as 1 W/m·K.

4.4.4 Boundary and initial conditions

The boundary conditions for the domain were specified in the 0 (zero) sub-directory in the Time directory and can be seen in Appendix A.

4.4.5 Specification of the files in the System directory

controlDict

The controlDict file was used to set input parameters that were essential for the creation of the database. In this file the application was specified as laplacianFoam that indicated that it was a pure heat conduction problem. The start time for the simulation was specified as 0 with an end time specified at 1 second. The time step deltaT was also specified in the file as 1 second for a steady-state scenario.

fvSchemes

The fvSchemes dictionary was used to set the numerical schemes for the terms such as the derivatives in equations. The only changes made in this dictionary were the specification of the conductivity k and the temperature T in the laplacian scheme.

fvSolution

The fvSolution dictionary controls the equation solvers, tolerances and algorithms. For the laplacianFOAM solver, the solvers selected was for the temperature, T , with a Preconditioned Conjugate Gradient (PCG) solver for symmetric matrices and a preconditioner specified as Diagonal Incomplete-Cholesky (DIC), for symmetric matrices.

A tolerance of 1e-06 was specified which represented the level at which the residual was small enough for an accurate solution. A relative tolerance of 0 was indicated, which was the ratio of current to initial tolerances. In this dictionary the under-relaxation value can also be specified. The current under-relaxation value is specified as 1.0.

Under-relaxation values were introduced as they are parameters that affect the convergence behaviour of a numerical scheme (Barron & Salehi Neyshabouri, 2005:923). Under-relaxation limits the amount to which a variable changes from one iteration to another. The under-relaxation factor is used to specify the amount of under-relaxation and can be between 0.0 and 1.0. A value of 1.0 indicates no under-relaxation and a value of 0.0 represents a solution which does not change with successive iterations. The optimum choice for specifying an under-relaxation factor is one that ensures a stable computation, but also one that ensures that the iterative process moves forward quickly (The OpenFOAM Foundation, 2013). This means that under-relaxation factors should be specified as high as possible without resulting in the divergence of the solution.

4.4.6 Viewing the mesh

The mesh was viewed in OpenFOAM by running the paraFoam command in the specific case directory. Figure 28 shows the domain and the mesh generated in OpenFOAM with the face nodes numbered.

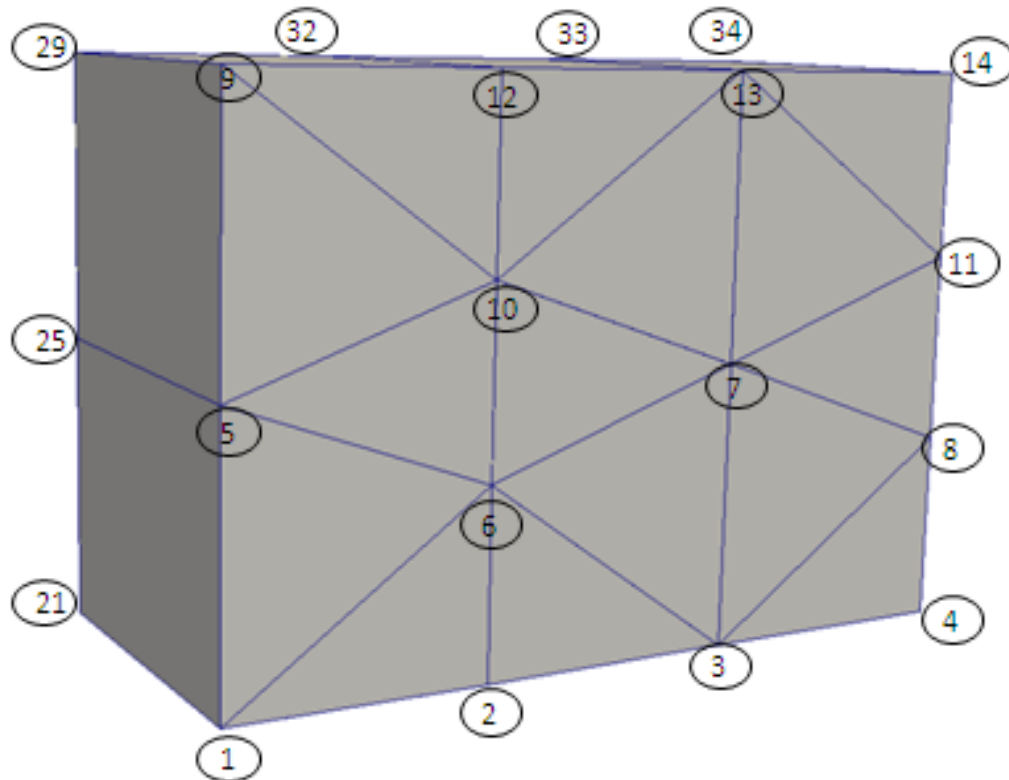


Figure 28: Mesh generated in OpenFOAM

4.4.7 Solution

The case was run in OpenFOAM by executing the laplacianFoam solver. The laplacianFoam solver is a pre-developed solver for the solution of the Laplace equation for diffusion. The results are discussed in Chapter 5.

4.5 Summary

In this chapter the implementation of Excel and OpenFOAM models were described which were used to determine the temperature distribution through a domain on a non-orthogonal mesh. These models were developed in order to compare the results calculated in Excel, with the results from OpenFOAM in order to aid in the investigation of the treatment of gradient calculations for non-orthogonal meshes.

The case study is based on the conduction through a domain with fixed boundary conditions at the sides of the domain. The domain was meshed quasi-two-dimensional with a non-orthogonal triangular mesh and the numbering of the different CVs, faces and face area vectors were indicated in a figure.

The theory described in Chapter 3 was implemented in the Excel model to calculate the temperature at each of the CV centres. The procedure that was followed to determine the temperature distribution was discussed. It is described in more detail in Appendix A.

The same heat conduction case was implemented in OpenFOAM. Details of the implementation were discussed. The case was run using the pre-compiled laplacianFoam solver for diffusion equations.

In the following chapter the results from the Excel and OpenFOAM models will be compared as a basis to investigate the different treatments of gradient calculations.

Chapter 5 Results and discussion

5.1 Excel and OpenFOAM results

The temperature distribution through the domain as calculated by OpenFOAM is shown in Figure 29 varying linearly between 600°C and 400°C.

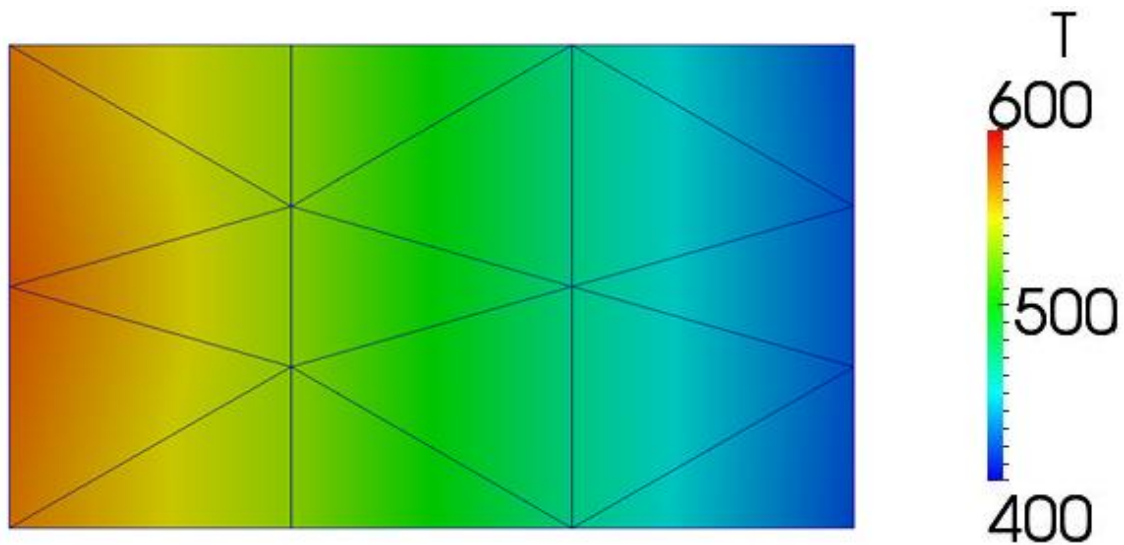


Figure 29: Temperature distribution through domain

The results from the Excel and the OpenFOAM models are compared with the analytical results in Figure 30 below.

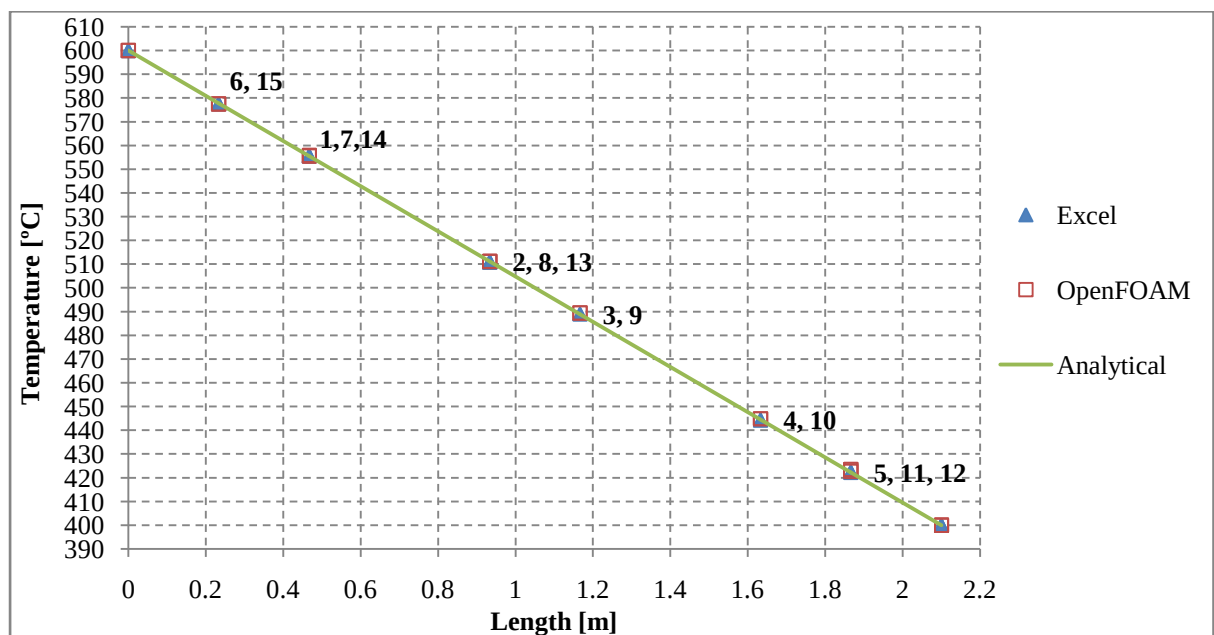


Figure 30: Comparison between the Excel, OpenFOAM and analytical results

Figure 30 illustrates the calculated temperatures at the centres of the CVs through the domain. The numbered points refer to the numbered CVs indicated in Figure 26. Table 1 is used to support the results illustrated in Figure 30.

Table 1: Comparison between the Excel, OpenFOAM and analytical results

Control Volume	Excel	OpenFOAM	Analytical Results
	Temperature [°C]	Temperature [°C]	Temperature [°C]
1	555.790	555.711	555.556
2	510.877	511.095	511.111
3	489.035	489.401	488.889
4	444.298	444.946	444.444
5	422.457	423.401	422.222
6	577.631	577.421	577.778
7	555.526	555.455	555.556
8	511.141	511.322	511.111
9	489.035	489.401	488.889
10	444.298	444.946	444.444
11	422.193	422.606	422.222
12	422.457	423.401	422.222
13	510.877	511.095	511.111
14	555.790	555.711	555.556
15	577.631	577.421	577.778

It is clear that there are small deviations between the Excel and OpenFOAM solution sets. The reasons for these differences were found by comparing the results from the individual steps that contributed to the temperatures calculated in the Excel and OpenFOAM models. The comparison of the results will be described in more detail in the following sections.

The different implementation methods used in OpenFOAM, that will be described in the following sections, were thus implemented in another Excel model and verified so that the same results from both OpenFOAM and the new Excel model could be obtained. This was done in order to ensure that all of the differences were found

between the fundamental theory approach and OpenFOAM implementation. It was then dealt with in the investigation.

5.2 Comparison between the results of the calculations in Excel and OpenFOAM

In order to compare the handling of non-orthogonal meshes between the Excel model and OpenFOAM, it was necessary to compare the factors used in the calculation of the temperatures at the centres of the CVs. Thus, it was first necessary to determine the non-orthogonal correction approach that was used in OpenFOAM.

After determining the non-orthogonal correction approach used in OpenFOAM and by referring to equations (3.11) and (3.12), it was necessary to compare the results of the coefficients of the orthogonal component of the face area vectors, $\frac{|\underline{\Delta}_f|}{|\underline{d}_f|}$ for the internal and boundary faces of the domain as well as the factors that contributed to the calculation of the gradients of ϕ at the faces of the CVs.

5.2.1 Comparison of the coefficients for the internal faces based on the different non-orthogonality treatments in Excel with OpenFOAM

In order to determine the approach followed for the handling of non-orthogonal meshes in OpenFOAM, the coefficients calculated for the internal faces of the domain for the minimum correction, orthogonal correction and over-relaxed approach, were compared with the coefficients that were calculated in OpenFOAM.

Figure 31 illustrates the results of the coefficients that were calculated using the different approaches in the Excel model in comparison to the results of the coefficients calculated in OpenFOAM.

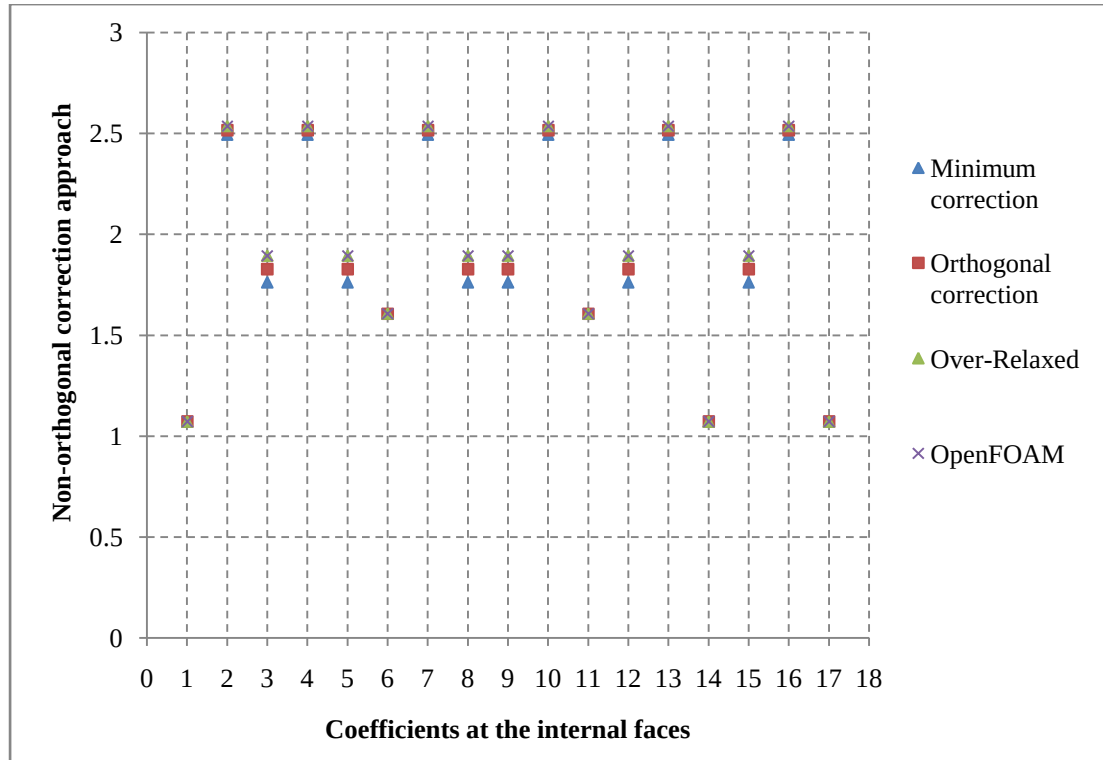


Figure 31: Comparison of the calculated coefficients based on the different non-orthogonal treatments with the method used in OpenFOAM

It can be seen in Figure 31 that the coefficients calculated based on the over-relaxed approach, were identical to the coefficients calculated in OpenFOAM. The method used in OpenFOAM (for the handling of non-orthogonal meshes) was therefore confirmed as being based on the over-relaxed approach.

The figure also shows that the coefficients calculated with the over-relaxed approach were larger than the coefficients calculated based on the minimum and orthogonal correction approaches. These larger coefficients enhanced the diagonal dominance of the coefficient matrix. It is desirable to improve the diagonal dominance as it improves the solver convergence.

5.2.2 Coefficients of the internal and boundary faces used in the matrix for the calculation of the temperature in each CV

In Figure 32 the coefficients of the orthogonal component of the face area vectors as used in the matrix, for each of the faces of the CVs in the domain, are compared.

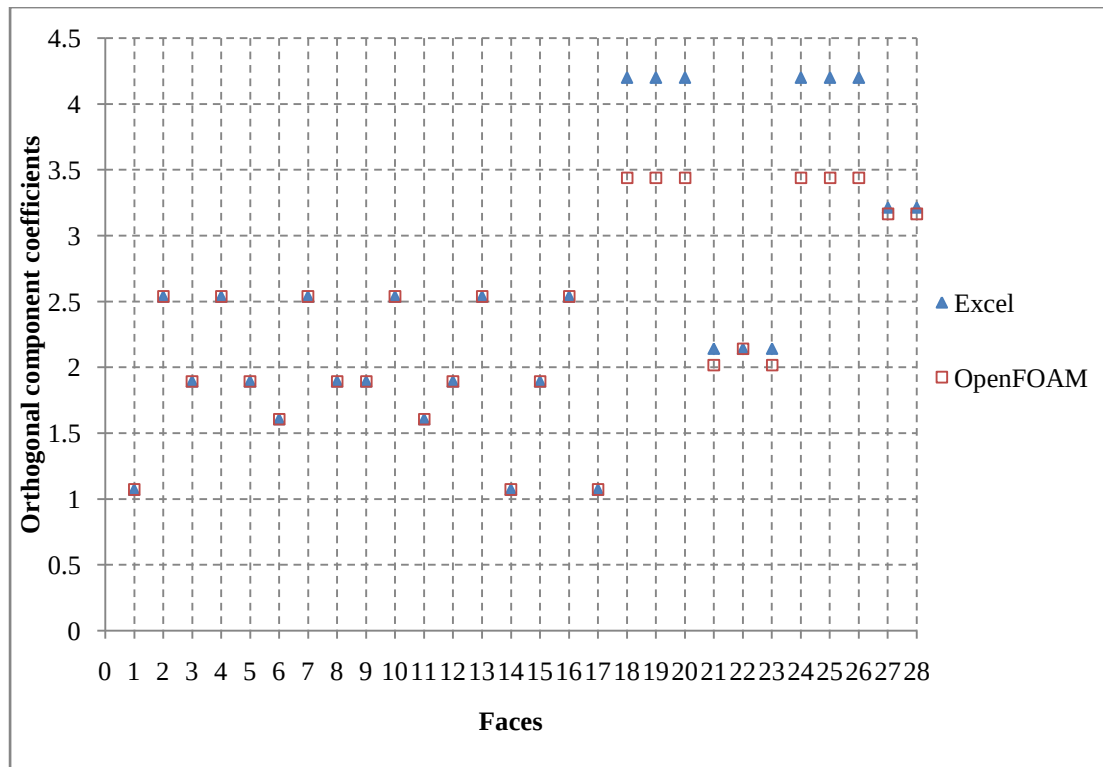


Figure 32: Comparison between the coefficients of the orthogonal components of the face area vectors between the Excel and OpenFOAM models

The figure shows that the orthogonal component coefficients of the face area vectors were the same for all of the internal faces of the domain (faces 1-17), but different for the boundary faces of the domain (faces 18 – 28).

The coefficients at the boundary faces of the original Excel model were calculated as described in Section 2.6.1. OpenFOAM however, calculated these orthogonal component coefficients of the boundary faces, as the magnitude of the face area vector, divided by the distance between the centre of the boundary face to the centre of the CV, adjacent to the boundary. The method OpenFOAM used is similar to the manner with which orthogonal CVs, adjacent to the boundaries of a domain, should be handled.

The reason for the differences found between the approaches, was that OpenFOAM set the non-orthogonal correction to zero at the boundaries by default, which would only be accurate for orthogonal meshes. This means that custom boundary conditions, with non-orthogonal corrections, must be implemented if more accurate solutions at the boundaries are required.

The gradients of ϕ at the faces of the CVs were calculated as defined by equation (3.13). It was necessary to compare the results of the gradients of ϕ over the CVs from the Excel and OpenFOAM models as well as the interpolation factors. The interpolation factors were also used in the calculation of the gradients of ϕ over the CVs, together with the magnitudes of the face area vectors and the volumes of the CVs.

5.2.3 Gradient of ϕ over a Control Volume

The results of the gradients of ϕ over the CVs are shown in Figure 33 and Figure 34 as they are compared with regard to the Excel and OpenFOAM models that were used. In Figure 33, the gradients of ϕ over the CVs for the x components are compared and in Figure 34 the y components are compared.

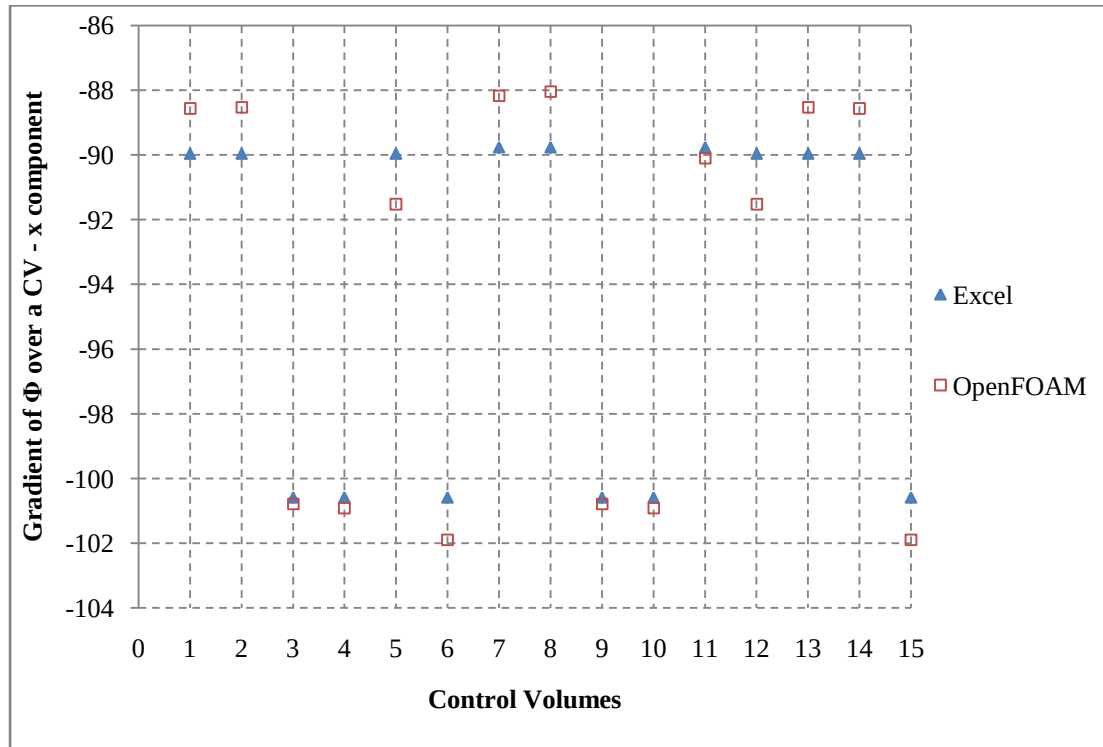


Figure 33: Comparison of the x-components of the gradients of ϕ over the CVs with regard to the Excel and OpenFOAM models that were used

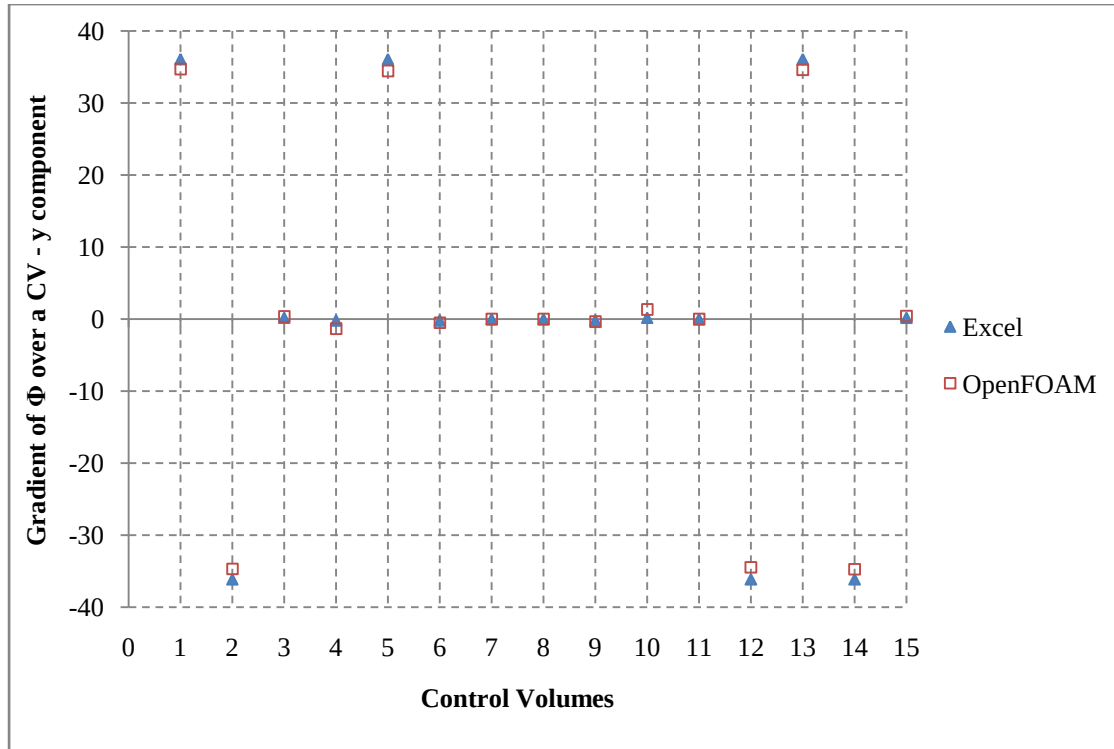


Figure 34: Comparison of the y-components of the gradients of ϕ over the CVs between the Excel and OpenFOAM models

It is clear from these figures that the Excel and OpenFOAM results for the gradients of ϕ over the CVs were different. As the gradient over a CV is calculated by equation (3.20), the results of the volumes of the CVs, the face area vectors and the interpolation factors as in equation (3.13) needed to be compared.

5.2.4 Face area vectors

The magnitude of the face area vectors for each of the faces in the domain from Excel and OpenFOAM can be seen in Figure 35 and in Table 25 of Appendix B.

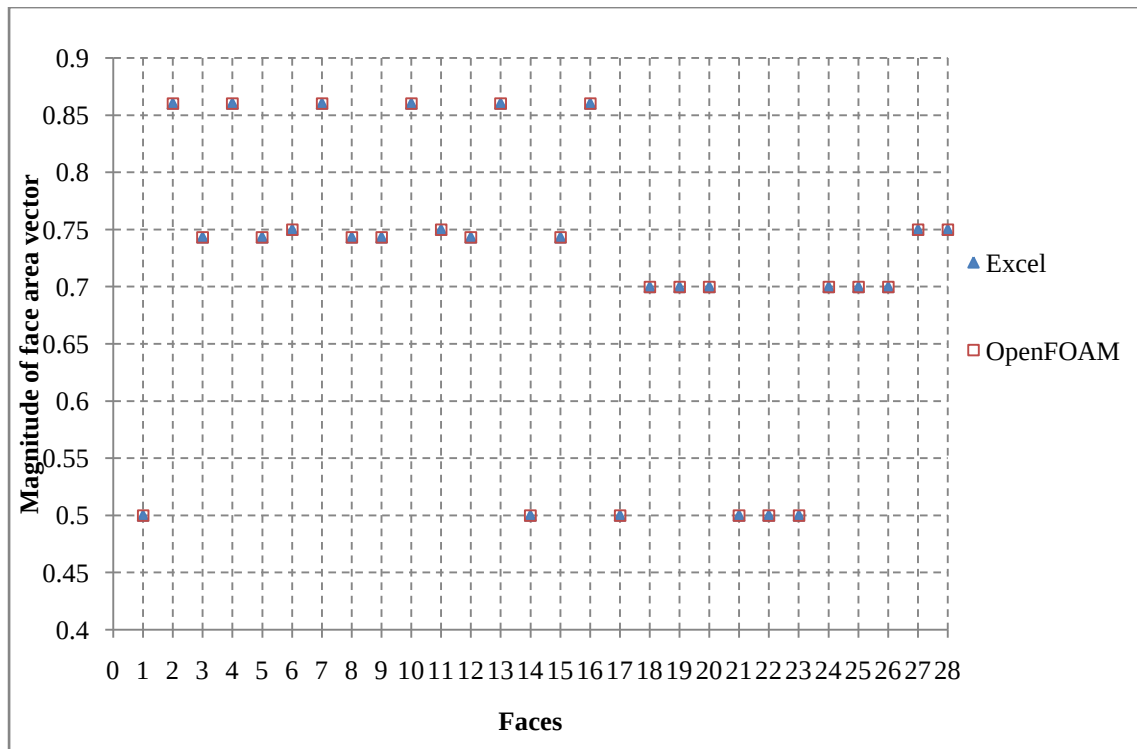


Figure 35: Comparison of the calculated magnitudes of the face area vectors between Excel and OpenFOAM

It is clear from investigating the above figure that there were no differences between the results of the magnitudes of the face area vectors between the Excel and OpenFOAM models.

5.2.5 Volumes of Control Volumes

The volumes of the CVs calculated in Excel, as described in Section 3.2.3 by using equation (3.22) and in OpenFOAM are compared in Figure 36. A comparison of the values can be seen in Table 27 of Appendix B.

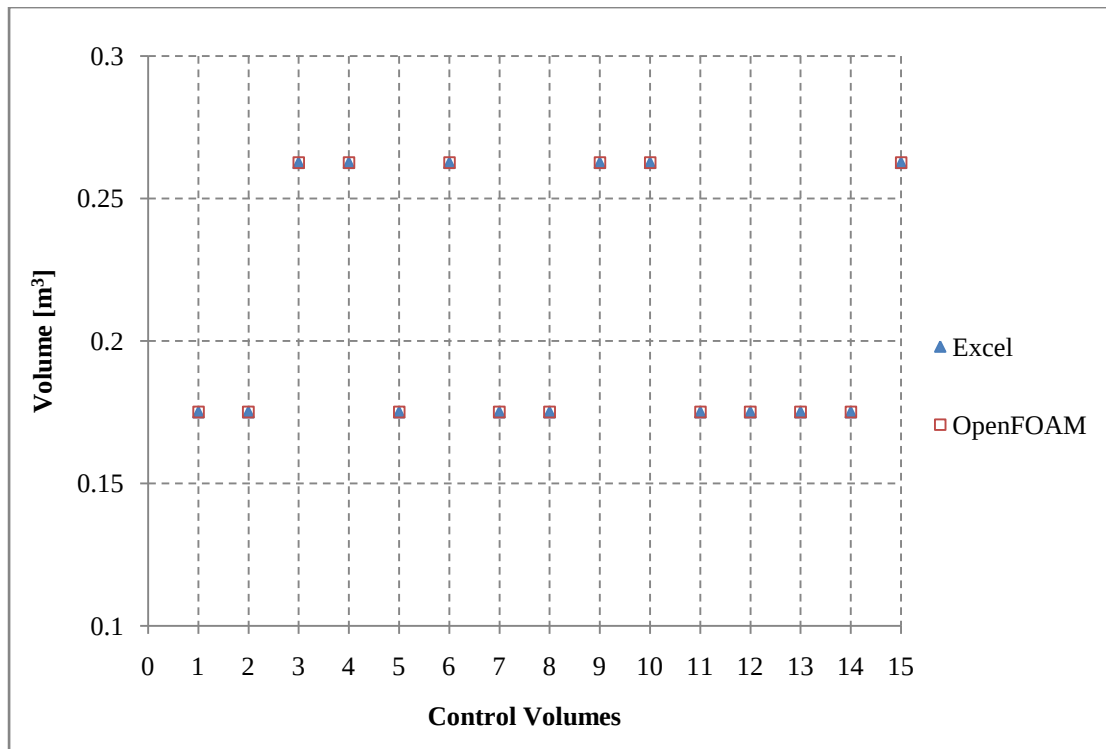


Figure 36: Comparison between the volumes of the CVs between the Excel and OpenFOAM models

As shown by the figure, there were no differences between the results of the calculated volumes in the Excel and OpenFOAM models.

5.2.6 Interpolation factors

The interpolation factors of the Excel and OpenFOAM models are compared in Figure 37 and Figure 38. A comparison of the values can be seen in Table 28 of Appendix B.

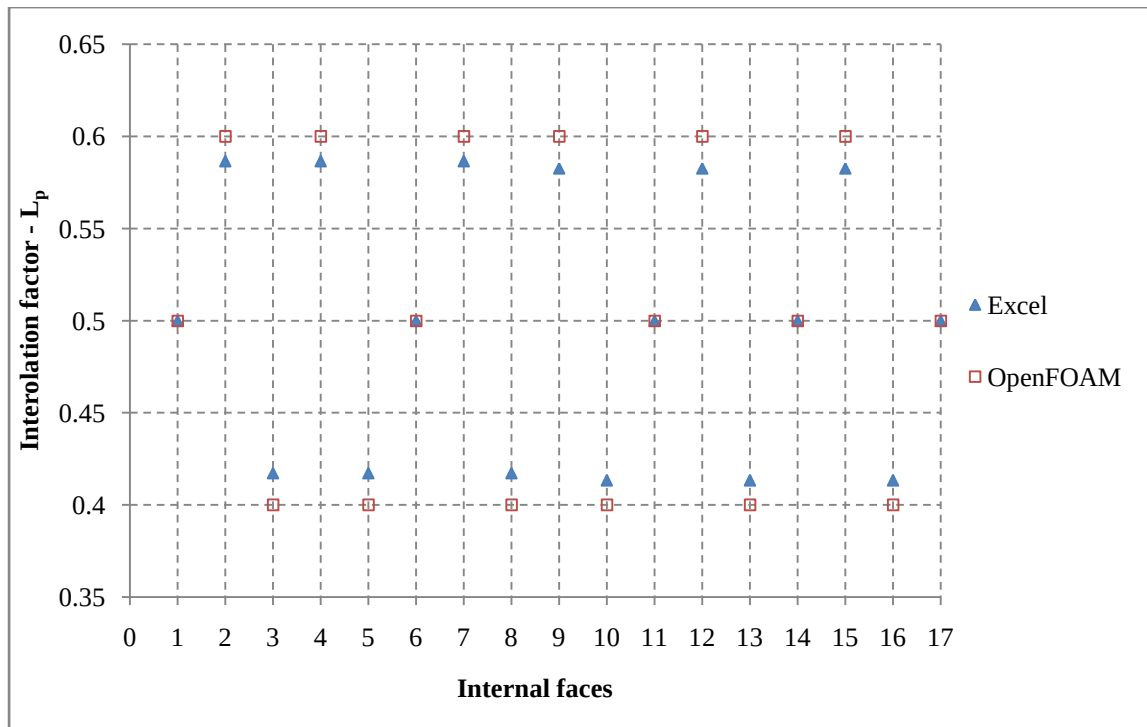


Figure 37: Comparison of the interpolation factor, L_p between the Excel and OpenFOAM models for the internal faces of the domain

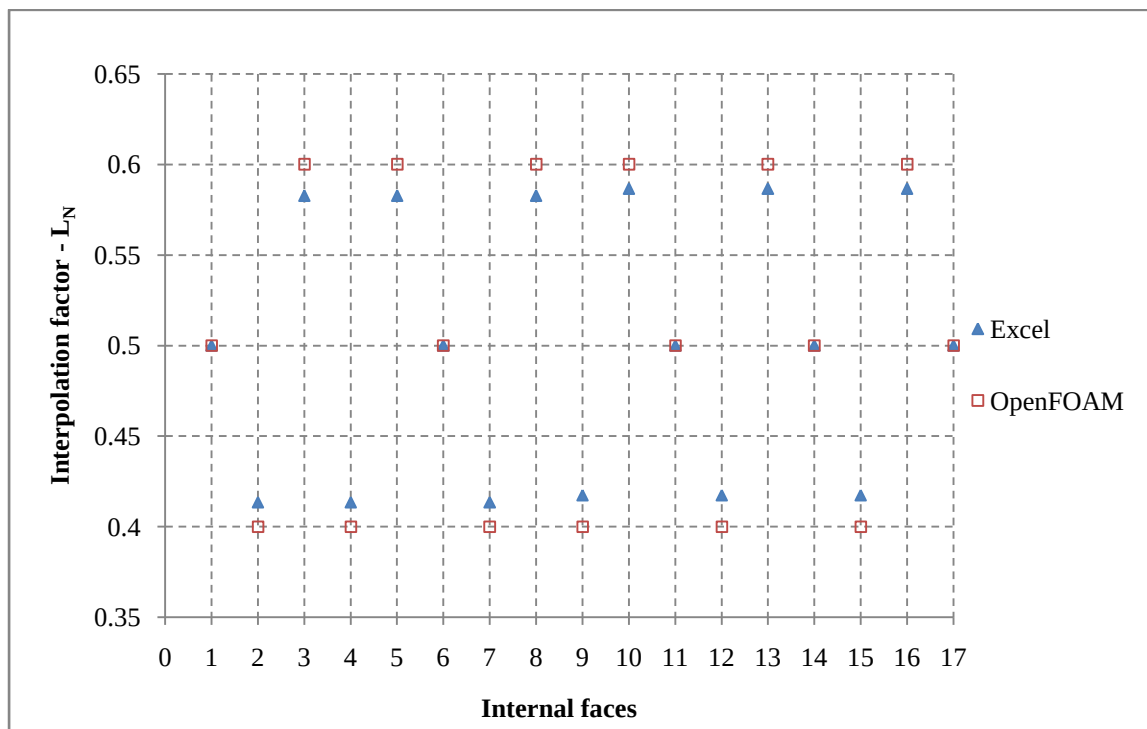


Figure 38: Comparison of the interpolation factor, L_N between the Excel and OpenFOAM models for the internal faces of the domain

The interpolation factors calculated in the Excel and OpenFOAM models were different as shown in Figure 37 and in Figure 38. In the Excel model the interpolation factors were calculated as described in Section 3.2.3 by using equation (3.14) and equation (3.15), but in OpenFOAM the interpolation factor L_P was calculated as seen in the surfaceInterpolation dictionary as the weight factor w :

forAll(owner, facei)

{

scalar SfdOwn = mag(Sf[facei] & (Cf[facei] - C[owner[facei]]));

scalar SfdNei = mag(Sf[facei] & (C[neighbour[facei]] - Cf[facei]));

w[facei] = SfdNei/(SfdOwn + SfdNei);

}

where & is used to indicate the dot product between the face area vector and the distance between the face and CV centres, i indicates the face index, C_f is the location of the centre of the face and C is the location of the centre of the CV. Mathematically it is expressed as:

$$w_{f,i} = \frac{\underline{S}_{f,i} \cdot (\underline{C}_N - \underline{C}_{f,i})}{\underline{S}_{f,i} \cdot (\underline{C}_{f,i} - \underline{C}_P) + \underline{S}_{f,i} \cdot (\underline{C}_N - \underline{C}_{f,i})} \quad (5.1)$$

Similar to the Excel model, the distance between the face and CV centres are calculated as seen in Table 8 in Appendix A, as well as described in Section 3.2.3. However, the calculations differ where the distance is dotted with the face area vector. This means that the distance between f and N is equal to $\underline{fN}$ and the distance between P and N is equal to $\underline{PN}$ as illustrated below, in Figure 39.

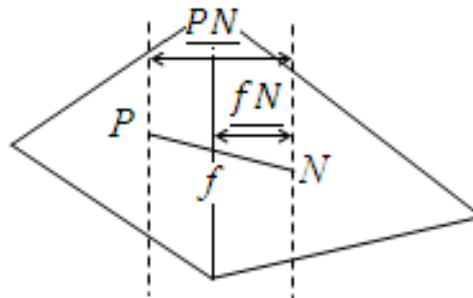


Figure 39: Calculation of the interpolation factors in OpenFOAM

As seen in Figure 39, $\underline{PN}$ and $\underline{fN}$ are not the same as $\underline{PN}$ and $\underline{fN}$ illustrated in Figure 21. The dot product stabilises the calculation, but has an influence on the accuracy of the solution. For meshes with a non-orthogonality less than 90° ($\alpha < 90^\circ$), the result of the dot product will be positive, but for meshes with a non-orthogonality higher than or equal to 90° , the dot product will be less than or equal to 0. This will result in a more stabilised solution, but one with poorer results (The Open Source CFD Toolbox, 2013).

5.3 Implementing the different implementation methods in the new Excel model

Differences in results between the Excel model and OpenFOAM were observed in the previous paragraphs. In order to ensure that the OpenFOAM implementation was understood correctly the different implementations were transferred to the new Excel model so that the results could be compared once again. The comparison of the new set of Excel results with those of OpenFOAM is shown in Figure 40 below and compared in tabulated form in Table 2.

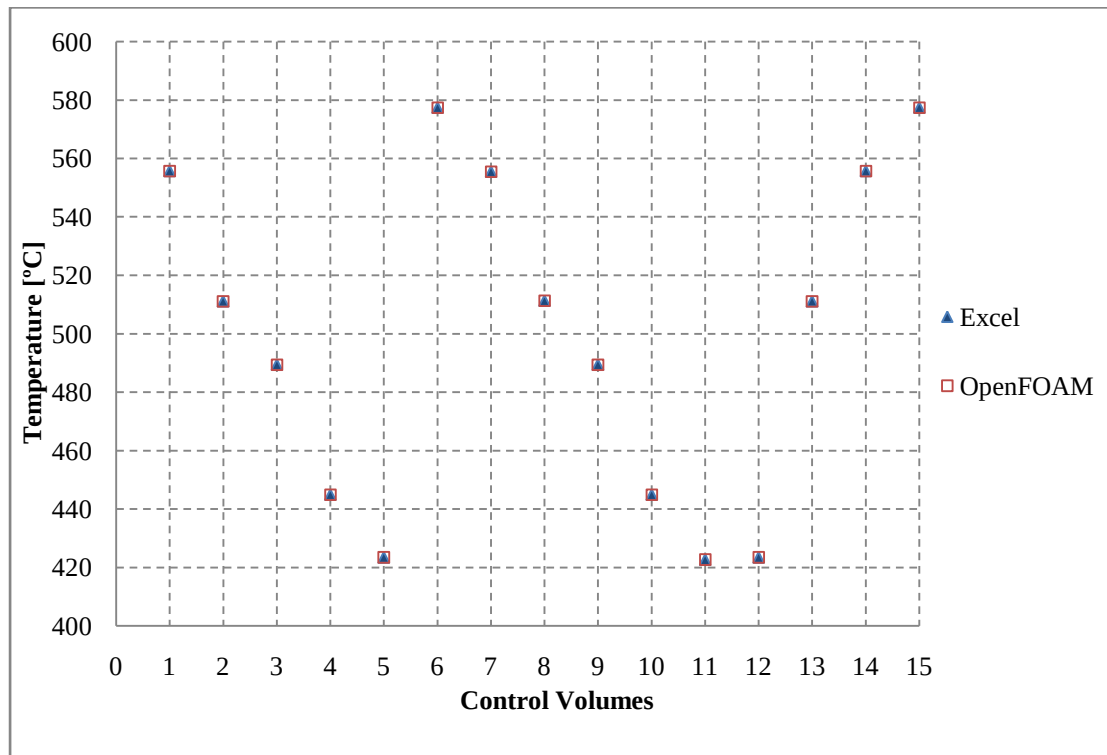


Figure 40: Comparing the temperature results for the new Excel model with the OpenFOAM results

The comparison shows that the results are identical as seen in Table 2.

Table 2: Comparison between the results of the new Excel model and the results from OpenFOAM

Control Volume	Excel	OpenFOAM
	Temperature [°C]	Temperature [°C]
1	555.711	555.711
2	511.095	511.095
3	489.401	489.401
4	444.946	444.946
5	423.401	423.401
6	577.421	577.421
7	555.455	555.455
8	511.322	511.322
9	489.401	489.401
10	444.946	444.946
11	422.606	422.606
12	423.401	423.401
13	511.095	511.095
14	555.711	555.711

After the differences were investigated in the calculation methods, between the theoretical Excel model and OpenFOAM, additional test cases were run to further investigate the treatment of non-orthogonality and compare the different decomposition methods for non-orthogonal meshes.

5.4 Treatment of non-orthogonality and mesh skewness

In this section the accuracy of the treatment of non-orthogonality in a mesh was investigated using OpenFOAM. This was done in the absence of mesh skewness. In a second case mesh skewness was introduced to demonstrate its effects on accuracy. Finally the three decomposition methods, for treating non-orthogonality (as described in Section 2.5) were tested. The test case was similar to the test case defined in Section 4.2 with the exception that it had a length of 1m instead of 2.1m. This was done to obtain CVs with increased non-orthogonality and/or skewness in order to

better illustrate findings. It also had a height of 0.3m and specified temperatures of 400°C and 600°C as seen in Figure 42.

5.4.1 Test for validity of non-orthogonal treatment

The first test that was performed was one where the basic concept of the treatment of non-orthogonality was evaluated. The decomposition method used was the over-relaxed approach as described in Section 2.5.3. The mesh used was a hexahedral mesh instead of a tetrahedral mesh. The reason for the use of a hexahedral mesh was to have more control over mesh non-orthogonality and skewness, while simultaneously demonstrating the different techniques. The meshed geometry is shown in Figure 41 below. All of the applicable files for these cases, such as the files described in Section 4.4, can be found on the compact disk attached in Appendix C.

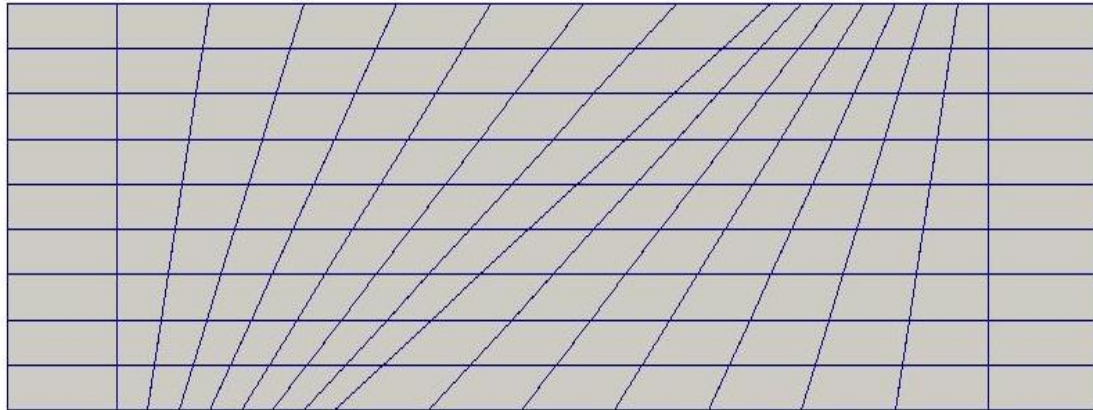


Figure 41: Non-orthogonal hexahedral mesh

The mesh in Figure 41 possessed non-orthogonality, but without mesh skewness, which eliminated a possible source of inaccuracy. The results for the heat conduction problem are shown in Figure 42 and Figure 43 below.

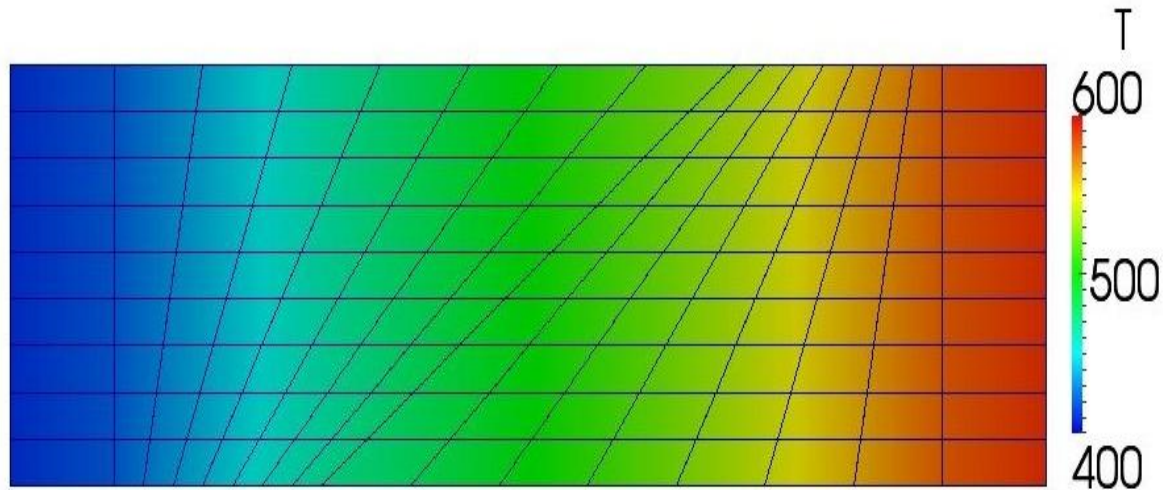


Figure 42: Temperature profiles on a non-orthogonal hexahedral mesh

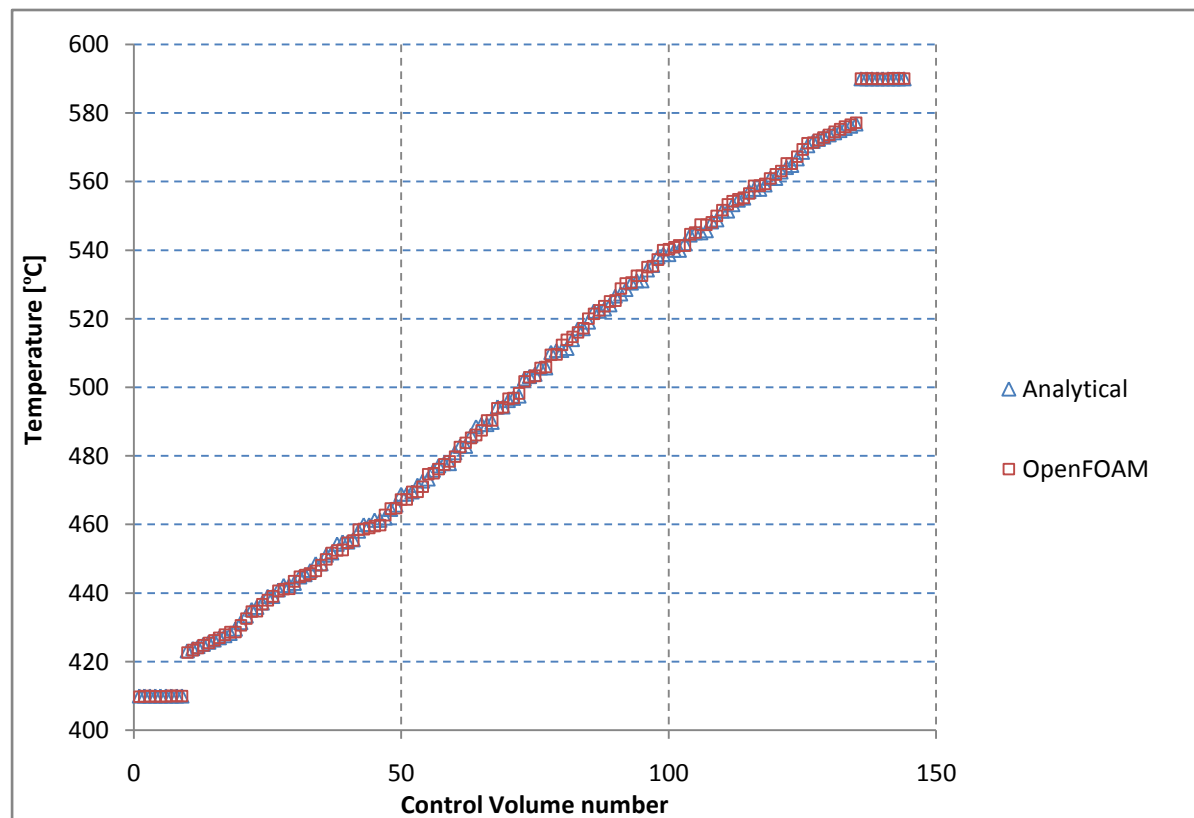


Figure 43: Comparison between the analytical and OpenFOAM results of the temperatures for the CVs

The CVs in Figure 42 are numbered in a top-down structured fashion and the temperature is plotted versus the CV number in Figure 43. By comparing the analytical solution with the solution in OpenFOAM in the graph, it is clear that the non-orthogonal treatment was functioning properly. When the temperature gradient

over the mesh was examined, it showed that there were flaws in the solution. This phenomenon is shown in Figure 44 below.

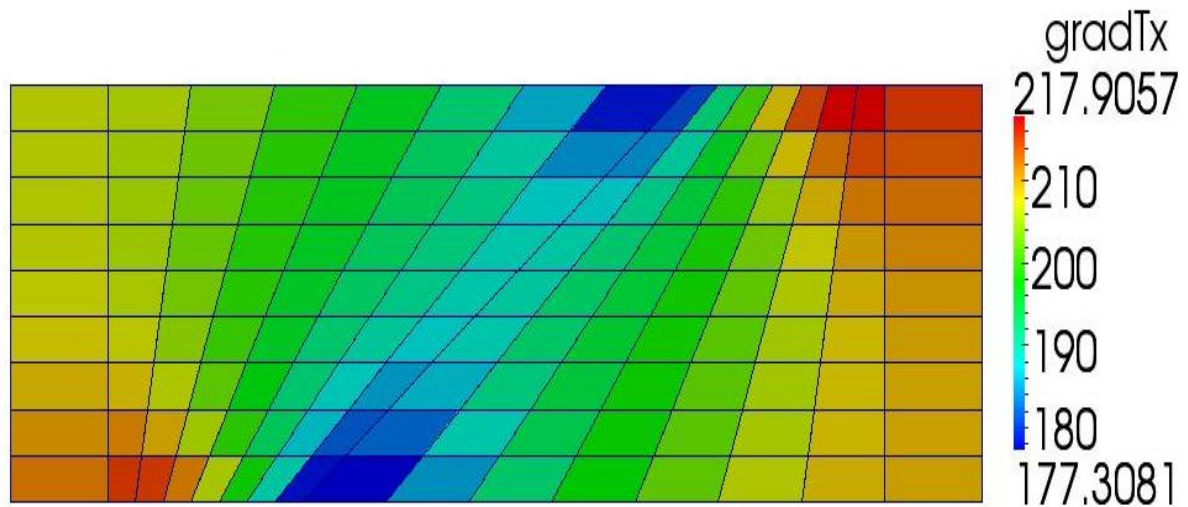


Figure 44: Error in temperature gradient calculation near boundaries

The plot of the temperature gradient distribution over the domain in Figure 44 illustrates an uneven gradient distribution instead of the physically correct uniform gradient distribution. The temperature gradient should have been constant at $200 \text{ W/m}\cdot\text{K}$ throughout the whole domain. The errors were the worst near the boundaries of the domain and at the points where the mesh was highly non-orthogonal as seen in the figure.

The reason for the error was a combination of the application of boundary conditions and mesh non-orthogonality at the boundaries of the domain. The implementation of the adiabatic boundary condition (the zero gradient condition), that was applied to the top and bottom of the geometry similar to the handling of boundaries as described in Section 5.2.2, was responsible for the error. During the iteration process, updating of boundary values occurs according to the zero gradient condition. The updating of boundary temperatures does not take into account the non-orthogonality near the boundary. The implementation method was valid, but the accuracy deteriorates to first-order in the presence of non-orthogonal mesh lines at the boundaries. The method can potentially be second-order accurate, but only in the case with an orthogonal mesh at the boundaries of the domain.

The inadequacy of the implementation of the adiabatic boundary condition was demonstrated by replacing the adiabatic condition with a fixed temperature boundary condition. In the following test case the temperature along the top and bottom boundaries were considered as known and obtained by linear interpolation between the left and right boundaries. In this case it was not necessary to update the temperatures at these boundaries as the temperatures were fixed. The correct constant temperature gradient distribution of $200 \text{ W/m}\cdot\text{K}$ is shown in Figure 45 below.

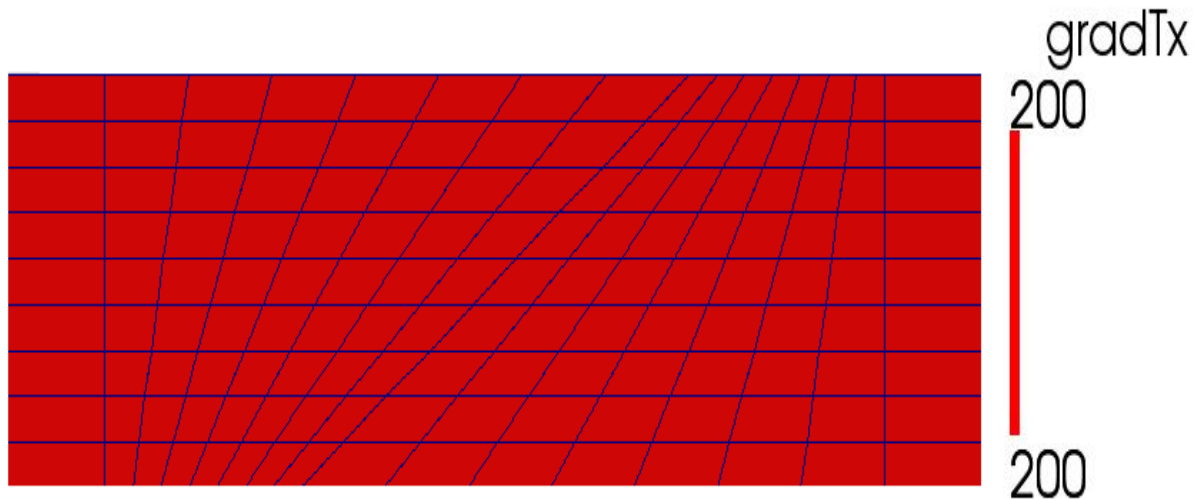


Figure 45: Constant temperature gradient over the domain

It can therefore be concluded that the implementation of the treatment of non-orthogonality was valid, provided that there is no mesh skewness present. However, care should be taken when updating boundary values, according to applied boundary conditions, when the mesh is not orthogonal at the boundaries of the domain.

5.4.2 Mesh skewness

Mesh skewness was defined in section 2.8 as when the lines connecting neighbouring CVs, do not pass through the centres of the dividing faces. The effect of mesh skewness is investigated in a new test case. A temperature solution is shown on a non-orthogonal mesh, which contains mesh skewness, as seen in Figure 46 below.

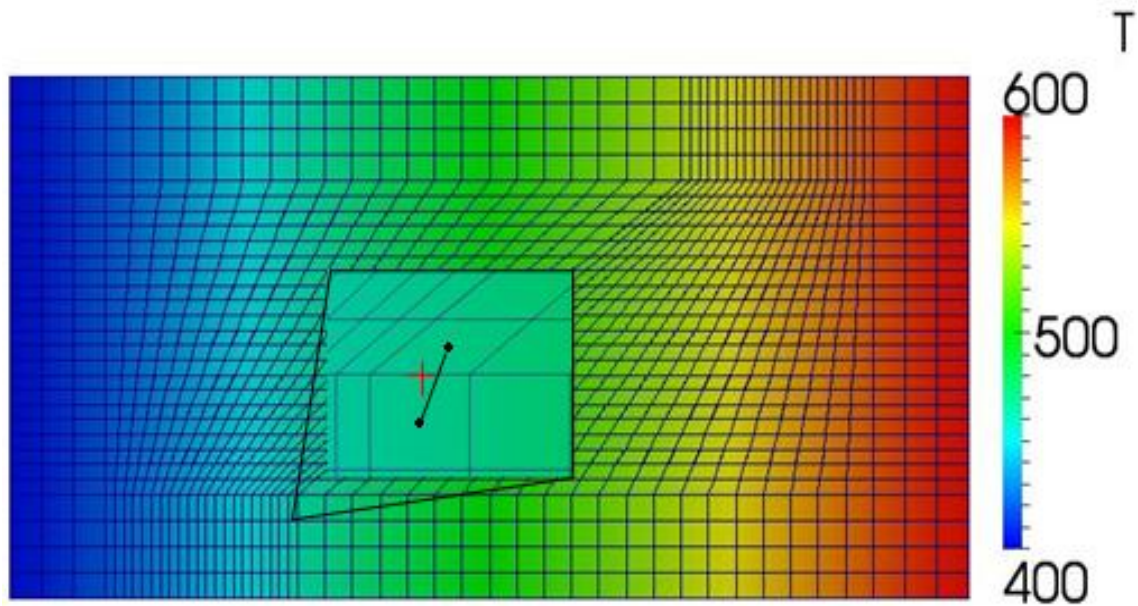


Figure 46: Non-orthogonal mesh with mesh skewness

The inaccuracy in the solution is difficult to spot on the temperature plot, but a plot of the gradient of the temperature shows the problem areas clearly. The distribution of the gradient of the temperature on the mesh is shown in Figure 47.

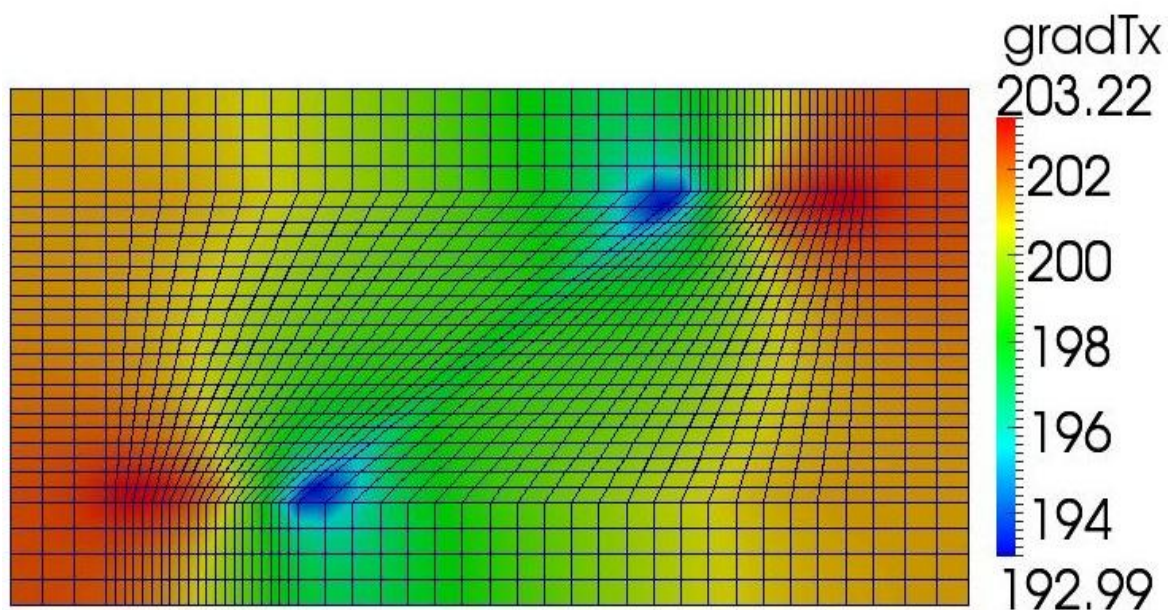


Figure 47: Distribution of the temperature gradient on a non-orthogonal mesh with mesh skewness

The current method of treating non-orthogonality, described in Section 2.5, is not meant to resolve mesh skewness. Figure 47 illustrates that the skewness error does

have an influence on the accuracy of the results. The improvement of mesh skewness has been discussed in Section 2.8.

5.4.3 Comparison between the decomposition methods for non-orthogonal meshes

In the following test case the three decomposition methods namely: the minimum correction, the orthogonal correction and the over-relaxed approaches for mesh non-orthogonality, were compared. These methods were implemented in OpenFOAM in the SurfaceInterpolation.C file which is available in the SurfaceInterpolation dictionary from the Source directory.

The test case was based on the standard test case, with fixed temperatures at the right and left boundaries of the domain and with adiabatic conditions, similar to the case described in Section 4.2. The temperature distribution through the geometry is shown in Figure 48 below.

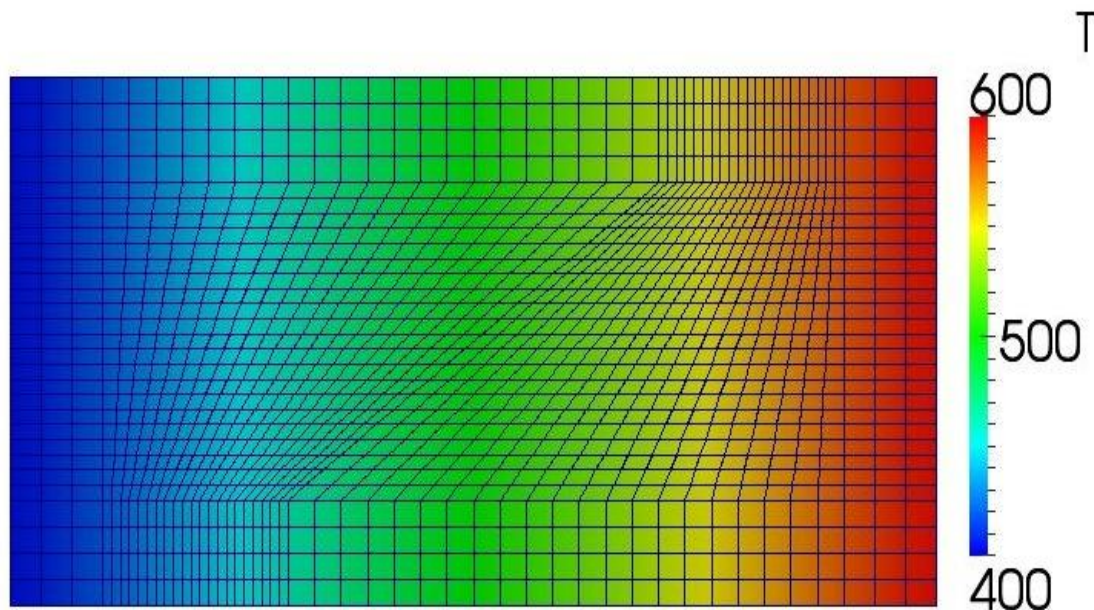


Figure 48: Linear temperature distribution in a rectangular block

It was found during the test case that there was a marked difference in the convergence behaviour and stability of the three different methods.

A plot is shown of the convergence history of the three different methods for the mesh (demonstrated in Figure 48) for different under-relaxation values of 1.0, 0.9 and 0.8, in Figure 49. The use of under-relaxation values have been described in Section 4.4.5.

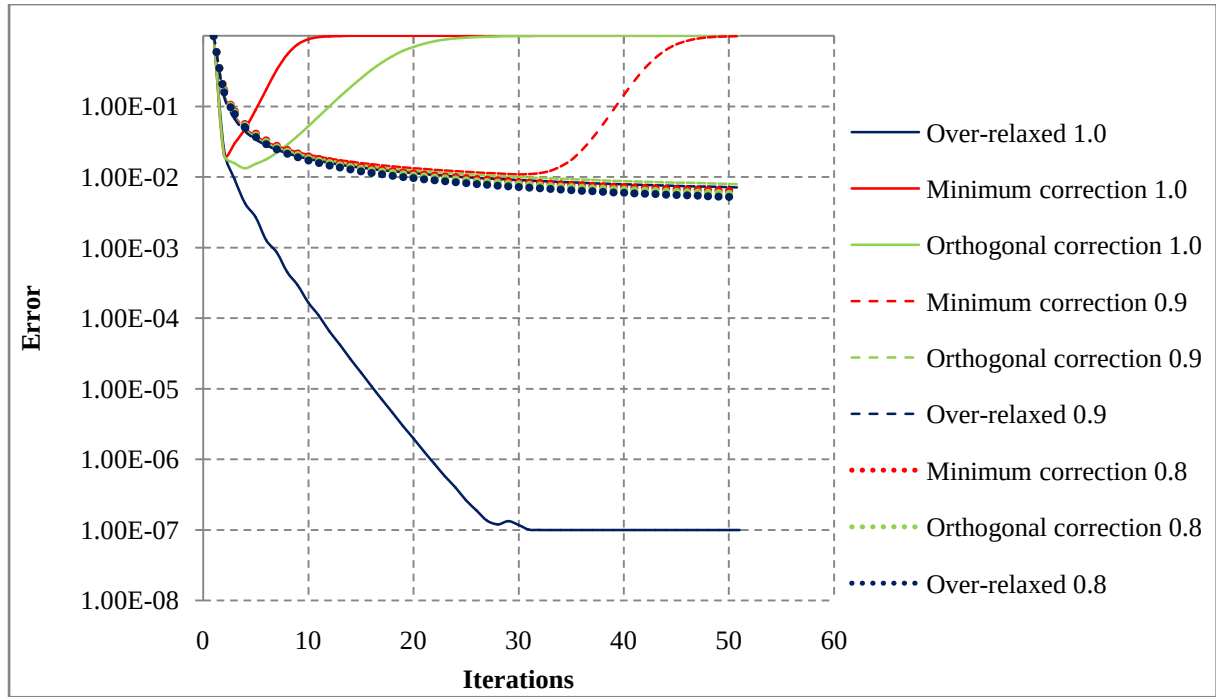


Figure 49: Convergence behaviour of the different decomposition methods on a skew mesh

In order to interpret the graph more clearly, another graph (see Figure 50) is used where the minimum correction, orthogonal correction and over-relaxed approach (with under-relaxation values of 0.9 and 0.8) are illustrated.

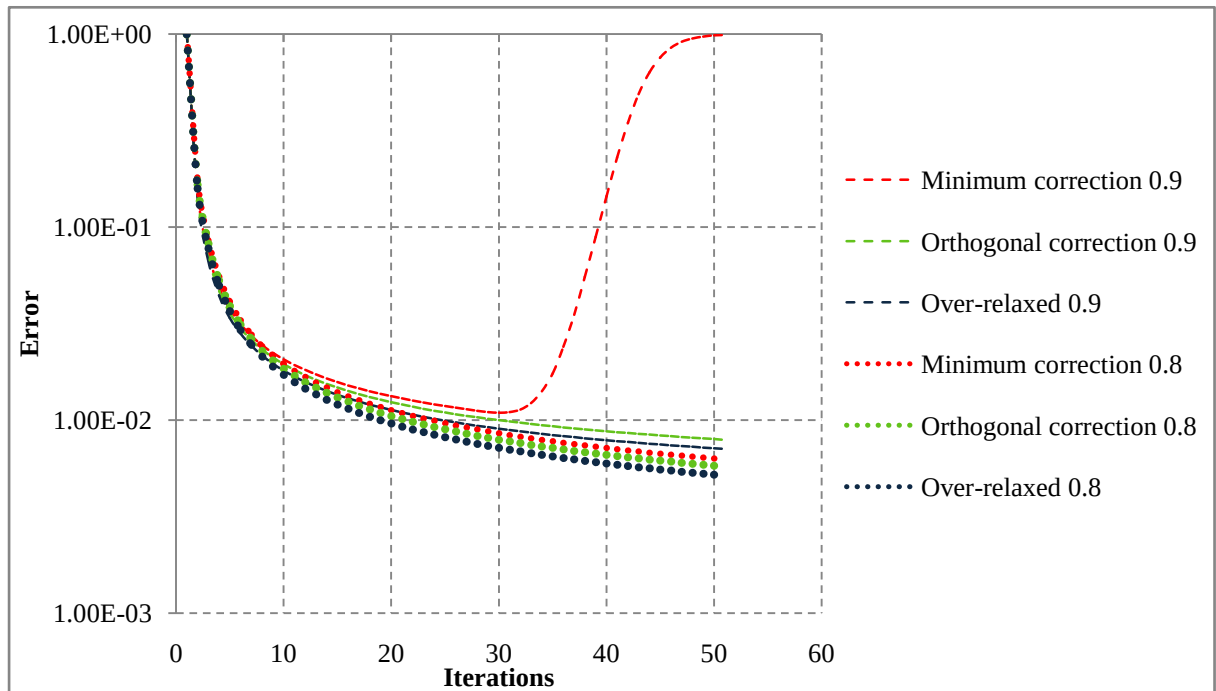


Figure 50: Convergence behaviour of the different decompositions on a skew mesh specifically for under-relaxation values of 0.9 and 0.8

The figure shows that the minimum correction approach was the least stable of the three approaches. In the first test, the minimum correction approach, with under-relaxation values of 1.0 and 0.9, struggled to converge. When the under-relaxation value was specified as 0.8, this approach also had the largest error compared to both the orthogonal correction and over-relaxed method.

The orthogonal correction approach struggled to converge when an under-relaxation value of 1.0 was specified, but was more stable when under-relaxation values of 0.9 and 0.8 were specified. In Figure 49 it is clear that the over-relaxed approach was the fastest converging approach, with the smallest error while having an under-relaxation value specified as 1.0.

The use of a lower relaxation value resulted in better stability for all of the approaches that were tested. With an under-relaxation factor of 0.8, all of the approaches seemed stable, but the convergence rate was very slow.

In Figure 51 the convergence history was plotted for the three different decomposition methods on a non-orthogonal mesh with no skewness. This was done again for the different under-relaxation values of 1.0, 0.9 and 0.8. This test case was based on the test case similar to Figure 42 and described in Section 5.4.1.

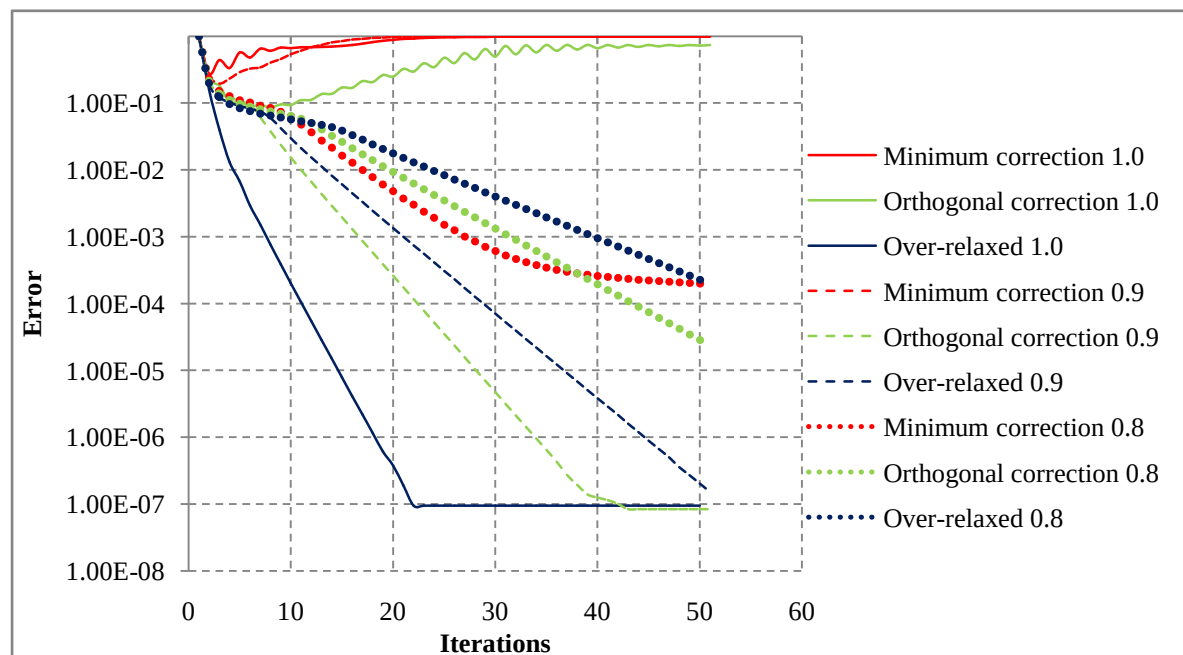


Figure 51: Convergence behaviour of the different decomposition methods on a mesh with no skewness

Figure 51 shows that the minimum correction approach struggled to converge with under-relaxation values that was specified as 1.0 and 0.9, but seemed more stable when an under-relaxation value of 0.8 was specified.

The orthogonal correction approach struggled to converge when the under-relaxation value was specified as 1.0, but for under-relaxation values of 0.9 and 0.8, this approach seemed stable similar to the behaviour of the approach for a non-orthogonal mesh with mesh skewness.

The over-relaxed approach converged easily with an under-relaxation value of 1.0 and it is clear that this approach is more stable than the orthogonal correction and minimum correction approaches.

The under-relaxation values control the stability and the convergence rate of the iterative process. As seen in the figures, the under-relaxation values increase the stability of solution, but decrease the rate of convergence. Figure 51 shows that the convergence rates of the different approaches were better than the convergence rates of the approaches in Figure 50 as the approaches were more stable on a mesh with no mesh skewness.

The over-relaxed approach has distinct advantages with regards to stability and convergence behaviour over the orthogonal correction and minimum correction approaches. These results are similar to the findings by Jasak (1996:151).

The advantages with regards to stability and convergence of the over-relaxed approach are due to the diagonal dominance of the matrix when the over-relaxed approach is used. The diagonal dominance for the over-relaxed approach is enhanced as the coefficient of the implicit term (refer to equation (2.5)) is greater than the coefficients of the implicit terms of the orthogonal correction and minimum correction approaches. This can also be seen in Figure 31 where the coefficients for the implicit terms for the internal faces were compared for the different approaches.

The use of under-relaxation values enhanced the diagonal dominance of the matrix (Jasak, 1995:115) and therefore improved the stability of the orthogonal and minimum correction approaches. As described in Section 4.4.5, under-relaxation

limits the amount to which a variable changes from one iteration to another iteration and therefore the convergence rate decreased for lower under-relaxation values.

5.5 Summary

In this chapter the Excel, OpenFOAM and analytical results were compared for a selected case. It was found that the coefficients, of the orthogonal components of the face area vectors, were similar for the internal faces of the domain, but different for the boundary faces due to the non-orthogonal correction at the boundaries of the domain, which was set to zero by default in OpenFOAM.

It was also found that the results of the gradients of ϕ over the CVs were different. The reason for this was that the interpolation factors were calculated differently in OpenFOAM. The method used by OpenFOAM resulted in a more stabilised solution, but with poorer results since it was less accurate in comparison to the analytical solution.

After implementing the different implementation methods used in OpenFOAM in the new Excel model, it was found that the results of the models were exactly the same. It was concluded that the implementation methods used in the Excel model were more accurate, but that the method used in OpenFOAM resulted in a more stable solution which would be preferred for complex meshes with severe non-orthogonality.

Further tests that made use of OpenFOAM and a hexahedral mesh were conducted to show the effect of a non-orthogonal mesh on a boundary. When the updating of boundary values are needed during a solution, compensation for the non-orthogonality will be needed in order to retain accuracy.

The effect of mesh skewness on the accuracy of results was also investigated in Section 5.4.2. The user should be aware that it was found that mesh skewness will contribute to less accurate results.

Finally the three decomposition methods used in the non-orthogonal compensation were evaluated. It was demonstrated that the over-relaxed approach performed better than both the minimum correction and orthogonal correction approaches. This is in accordance with the findings of Jasak (1996:143).

Chapter 6 Conclusions

The treatment of gradient calculations on non-orthogonal meshes using the FVM was investigated in this study. The engineering application is one of heat conduction in a MCHE. A finite volume numerical solution method was developed and implemented in Excel to handle heat conduction on a non-orthogonal mesh. Various test cases were considered to investigate the accuracy with which a FVM can handle non-orthogonal meshes. The test cases were solved by a second FVM-based software code namely OpenFOAM. The differences in the results were used to track the different implementation of discretisation methods used in the two independent codes. Knowledge of the different implementations led to a better understanding of how accuracy can be improved when handling non-orthogonal meshes.

In the following section a list of the important findings are given.

6.1 Summary of the findings

During the investigation of the ability of the FVM to handle heat conduction on a non-orthogonal mesh, the following objectives were achieved and findings were made:

- Knowledge was gained on the handling of non-orthogonal meshes based on a theoretical approach as the differential equations governing conduction heat transfer were integrated and discretised on a general non-orthogonal mesh using the FVM.
- An independent numerical program was developed to calculate and organise the above discretised equations into a system of linear algebraic equations. The program was developed in Excel with the added feature that all computations were visible simultaneously on one calculation sheet. This feature made the program especially suitable for research purposes.
- A relative simple test case of heat conduction in a rectangular block was considered. A non-orthogonal mesh that consisted of tetrahedrons was generated. The case was solved using the Excel program.

- The test case was also solved with OpenFOAM. The results of the Excel program and of OpenFOAM were compared. Small differences occurred and their origin was traced to slightly different implementation methods. Knowledge of the differences in implementation between the two codes brought about a better understanding of the aspects that influence accuracy and stability.
- The three decomposition methods, of splitting a face area vector in two components for handling non-orthogonality in a mesh, were implemented and evaluated. The results showed that the over-relaxed approach performed the best in terms of stability and convergence rate. This result confirmed the findings of previous studies of Jasak (1996:143).
- Computations on meshes, which contain mesh skewness and non-orthogonal mesh lines at boundaries, were performed and an accompanying decrease in accuracy was observed. The results showed that the standard FVM as implemented in the Excel code and in OpenFOAM will require special methods to compensate for the decrease in accuracy.
- During the study a deeper knowledge and understanding was gained of the challenge of obtaining accurate solutions of heat conduction on non-orthogonal meshes. This knowledge may lead to the overall improvement of heat transfer models for the MCHE.

6.2 Recommendations for further work

During the study a few aspects were identified as possible topics for further research namely:

- Firstly the FVM implemented in Excel proved to be a valid approach that enabled effective research in numerical solutions. Efforts may be spent to further develop the potential of this type of implementation.
- The problem of mesh skewness and the handling of boundary conditions, in the presence of non-orthogonal mesh lines at the boundary, present possible topics which could require further attention. Practical solutions to these

problems will make a useful contribution to the success of the application of the FVM on non-orthogonal meshes.

References

APSLEY, D. 2012. Lecture notes Spring 2013. (Part of the course in Computational Fluid Dynamics at Manchester University.) United Kingdom.

BAKKER, A. 2012. Class lecture notes (Part of the course in Computational Fluid Dynamics at Dartmouth college, 2002-2006) available at:
<http://www.bakker.org/dartmouth06/engs150/> Date of access: 22 July 2012.

BARRON, R.M. & SALEHI NEYSHABOURI, A.A. 2003. Effects of under-relaxation factors on turbulent flow simulations. *International Journal for Numerical Methods in Fluids*, 42:923-928.

COETZEE, R.V. 2005. Volume weighted interpolation for unstructured meshes in the finite volume method. Potchefstroom: North-West University (Thesis – DPhil) 99p.

DAMIÁN, S.M. Description and utilization of interFOAM multi phase solver. 2013. Available at:
<http://infofich.unl.edu.ar/.../5e6dfd7ff282e2deabe7447979a16d49b0b1b675.pdf> . Date of access: 20 April 2013.

DATE, A.W. 2005. Solution of transport equations on unstructured meshes with cell-centred collocated variables. Part :1 Discretization. *International Journal of Heat and Mass Transfer*, 48:1117-1127.

FERZIGER, J.H. & PERIĆ, M. 1996. Computational Methods for Fluid Dynamics. Springer–Verlag. 364p. 1st Edition.

FERZIGER, J.H. & PERIĆ, M. 2001. Computational Methods for Fluid Dynamics. Springer–Verlag. 431p. 3rd Edition.

FRINK, N.T. 1991. Three dimensional upwind scheme for solving the Euler equations on unstructured tetrahedral grids. Virginia: Virginia Polytechnic Institute and State University (Thesis – DPhil) 131p.

HEATRIC. Compact Heat Exchanger Solutions. 2012. Available at:
<http://www.heatric.cn/download/Brochure.pdf> Date of access: 23 March 2012.

JASAK, H. 1996. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. London: University of London (Thesis – DPhil) 396 p.

JURETIĆ, F. & GOSMAN, A.D. 2010. Error analysis of the Finite-Volume Method with respect to mesh type. *Numerical Heat Transfer, Part B*, 57:414-439.

KHAN, M.G. & FARTAJ, A. 2011. A review on microchannel heat exchangers and potential applications. *International Journal of Energy Research*, 35:553-582.

KRUGER, J.H., LE GRANGE, L.A. & GREYVENSTEIN, G.P. 2008. Integrated systems CFD modelling applied to diffusion-bonded compact heat exchangers. (Proceedings of 4th International Topical Meeting on High-Temperature Reactor Technology, HTR 2008, September 28-October 1, 2008, Washington, DC, USA.)

LILEK, Ž. & PERIĆ, M. 1995. A fourth-order finite volume method with colocated variable arrangement. *Computers and Fluids*, 24(3):239-252.

LOUDYI, D., FALCONER, R.A. & LIN, B. 2007. Mathematical development and verification of a non-orthogonal finite volume model for groundwater flow applications. *Advances in Water Resources*, 30: 29-42.

MENON, S. 2011. A numerical study of droplet formation and behaviour using interface tracking methods. Massachusetts: University of Massachusetts. (Thesis – DPhil) 211 p.

MURTHY, J.Y. & MATHUR, S.R. 2002. Lecture notes for Spring 2002. Numerical methods in heat, mass and momentum transfer. (Lecture notes part of the course ME 608 at Purdue University).

NEBENFÜHR, B. 2010. OpenFOAM: A tool for predicting automotive relevant flow fields. Göteborg: Chalmers University of Technology. (Thesis-Master's) 114 p.

PRA, F., TOCHON, P., MAUGET, C., FOKKENS, J. & WILLEMSSEN, S. 2008. Promising designs of compact heat exchangers for modular HTRs using the Brayton cycle. *Nuclear Engineering and Design*, 238:3160-3173.

SEZAI, I. 2011. Lecture notes Fall 2011-2012. Methods for dealing with complex geometries (Part of the course in Computational Fluid Dynamics at Eastern Mediterranean University).

STEWART, J. 2003. Multivariable Calculus early transcendentals. Thomson Brooks/Cole. 1168 p. 5th edition.

THE OPENFOAM FOUNDATION. 2011-2012. Mesh description.
<http://www.openfoam.org/docs/user/mesh-description.php#x23-1280005.1> Date of access: 16 August 2012.

THE OPENFOAM FOUNDATION. 2013. Solution and algorithm control. <http://www.openfoam.org/docs/user/fvSolution.php>. Date of access: 10 March 2013.

THE OPEN SOURCE CFD TOOLBOX. 2013.
<ftp://ftp.ux.uis.no/uis/x/OpenFOAM/OpenFOAM-1.5-dev/src/finiteVolume/lnInclude/surfaceInterpolation.C> Date of access: 2 February 2013.

THONON, B. & BREUIL, E. 2000. Compact heat exchangers technologies for the HTRs recuperator application. (*IAEA Technical Committee Meeting on “Gas Turbine Power Conversion Systems for Modular HTGRs”, held from 14-16 November 2000 in Palo Alto, California.* International Atomic Energy Agency, Vienna (Austria). Technical Working Group on Gas-Cooled Reactors IAEA-TECDOC-1238:149-160.)

UBBINK, O. 2011. Lecture notes July 2011. (Part of the course in Computational Fluid Dynamics at the North-West University.) Potchefstroom.

VERSTEEG, H.K. & MALALASEKERA, W. 2007. An introduction of Computational Fluid Dynamics: The Finite Volume Method. *Pearson Education Limited*, Second Edition, 503 p.

WANG, Z.J. 2011. CFD and Others: Why is the order of accuracy important?
<http://cfd-zjw.blogspot.com/2011/10/most-of-commercial-cfd-codes-are-1st-or.html>
Date of access: 27 April 2012.

Appendix A - Excel and OpenFOAM calculations

Excel Calculations

Node coordinates

In the theoretical Excel model, the three-dimensional coordinates of each of the nodes were determined and indicated in a table similar to Table 3. Only the first few entries are given as all of the data can be found in the attached CD in Appendix C.

Table 3: Node coordinates

Nodes	Coordinates		
	X	Y	Z
1	0	0	0
2	0.7	0	0
3	1.4	0	0
4	2.1	0	0
5	0	0.75	0

The node coordinates were used to determine the face centres and the CV centres. The centre of a face was calculated by determining the midpoint between the nodes of the face as was discussed in Section 3.2.3.

For each of the faces the owner and neighbouring CVs are indicated as seen in Table 4. There were a total of 17 internal faces and 11 boundary faces. The complete table is available in the attached CD with the complete set of all the Excel calculations.

Table 4: Owners and neighbours of a face

Face	Owner	Neighbour
1	1	2
2	1	6
3	6	7
4	2	3
5	3	8

Face area vector

The face area vectors for the first five faces can be seen in Table 5 and were determined as described in Section 3.2.2 by equation (3.10).

Table 5: Face area vectors

Faces	S_{fy}	S_{fx}	$ \underline{S}_f $
1	0	0.5	0.5
2	0.7	-0.5	0.860232527
3	0.7	0.25	0.743303437
4	0.7	0.5	0.860232527
5	0.7	-0.25	0.743303437

Face centres

The face centres for the first five faces can be seen in Table 6 calculated with equations (3.16) and (3.17).

Table 6: Face centres

Faces	Face Centres [X]	Face Centres [Y]
	(Distance in X between Nodes)/2	(Distance in Y between Nodes)/2
1	0.7	0.25
2	0.35	0.25
3	0.35	0.625
4	1.05	0.25
5	1.05	0.625

Control Volume centres

The CV centres were then calculated by determining the sum of the face centres of a CV divided by the number of faces as calculated by using equations (3.18) and (3.19) as described in section 3.2.3 and can be seen in Table 7 for the first five CVs:

Table 7: CV centres

CV	CV Centre [X]	CV Centre [Y]
	[Sum of face centres x/3]	[Sum of face centres y/3]
1	0.466666667	0.166666667
2	0.933333333	0.166666667
3	1.166666667	0.416666667
4	1.633333333	0.416666667
5	1.866666667	0.166666667

The CV centres were then used to determine the distance between the neighbouring CVs as well as the distance between a face and the centre of the CV. The distance between the CV centres and face centres were used in the calculation of the interpolation factors.

Interpolation factors

The interpolation factors were then calculated as described in Section 3.2.3 by using equations (3.14) and (3.15). This can be seen for the first five faces in Table 8. These interpolation factors were used for the calculation of the gradient of ϕ , at the face of a CV for the non-orthogonal corrector term, of the face area vector. It was also used to determine the value of ϕ at the face of a CV.

Table 8: Interpolation factors

Face	L_P	L_N
1	0.5	0.5
2	0.586603176	0.413396824
3	0.417281021	0.582718979
4	0.586603176	0.413396824
5	0.417281021	0.582718979

In order to determine the gradient of ϕ at the face of a CV after calculating the interpolation factors, the gradient of ϕ over the CV could be calculated. This involved the calculation of the volume of the CV as well as the sum of the products of the face area vector and the field variable ϕ at each of the faces of the CV.

Volume of a Control Volume

The volume calculated for the first five CVs can be seen in Table 9 and was calculated as described in Section 3.2.3 by using equation (3.22).

Table 9: Volumes of CVs

CV	Volume
1	0.175
2	0.175
3	0.2625
4	0.2625
5	0.175

Sum of the products of the face area vector and the field variable ϕ_f

In order to determine the sum of the products of the face area vector and the field variable ϕ_f at each of the CV faces as in equation (3.23), the face area vector of each face had to be determined and the value of ϕ at the CV centres had to be interpolated to the face of the CV as in equation (3.24).

Value of ϕ at the face of a Control Volume

The value of ϕ at the centre of a CV was interpolated to the face of a CV as described in Section 3.2.3 by using equation (3.24). The initial values of ϕ_f for the first five faces can be seen in the following table:

Table 10: The value of ϕ at the CV face

Faces	ϕ_f
	$\phi_f = L_P \phi_P + L_N \phi_N$
1	533.3333333
2	564.8191443
3	564.7501266
4	501.8475223
5	501.9165401

The gradient over the CVs can be seen in the following tables where the x- and y-coordinates of the owner and neighbouring CVs were calculated by using equation (3.20):

Table 11: Gradient of ϕ over the CV - Owner

CV	$(\nabla\phi)_P$	
	$(\nabla\phi)_{P,x} = \frac{1}{V_P} \sum_{f=1}^{nf} S_x \phi_f$	$(\nabla\phi)_{P,y} = \frac{1}{V_P} \sum_{f=1}^{nf} S_y \phi_f$
1	-89.95946002	36.11651379
2	-89.95946002	36.11651379
3	-100.5824616	-0.18404731
4	-89.95946002	-36.11651379
5	-100.5824616	0.18404731

Table 12: Gradient of ϕ over the CV - Neighbour

CV	$(\nabla\phi)_N$	
	$(\nabla\phi)_{N,x} = \frac{1}{V_N} \sum_{f=1}^{nf} S_x \phi_f$	$(\nabla\phi)_{N,y} = \frac{1}{V_N} \sum_{f=1}^{nf} S_y \phi_f$
1	-89.95946002	-36.11651379
2	-100.5824616	-0.18404731
3	-89.76226647	0
4	-100.5824616	0.18404731
5	-89.76226647	0

The complete set of results can be seen in the Excel spreadsheet on the attached CD. The gradient of ϕ at the face of the CV can then be calculated as seen in the table below by using equation (3.13).

Table 13: Gradient of ϕ at the face of a CV

Face	$(\nabla\phi)_f$	
	$(\nabla\phi)_{f,x} = L_P(\nabla\phi)_{P,x} + L_N(\nabla\phi)_{N,x}$	$(\nabla\phi)_{f,y} = L_P(\nabla\phi)_{P,y} + L_N(\nabla\phi)_{N,y}$
1	-89.95946002	1.49214E-13
2	-94.35097515	21.10997713
3	-94.27732856	-0.076799449
4	-94.35097515	-21.10997713
5	-94.27732856	0.076799449

Note that Face 2 and 3 do not belong to CVs 2 and 3 as given in Table 11 and Table 12 as seen in Figure 26.

Orthogonal and non-orthogonal components of gradient

In order to calculate the non-orthogonal corrector term, the non-orthogonal corrector factor had to be calculated as described in Section 3.2.3. This, first of all, required the calculation of the orthogonal component of the face area vector.

Orthogonal component

The orthogonal component of the face area vector, based on the over-relaxed approach calculated for the first five faces, as calculated by using equation (2.5), can be seen in Table 14.

Table 14: Orthogonal component of the face area vector for each of the faces

Face	Δ_x	Δ_y	$ \underline{\Delta} $
	$\frac{d_x}{(d_x S_x + d_y S_y)} \underline{S} ^2$	$\frac{d_y}{(d_x S_x + d_y S_y)} \underline{S} ^2$	$\sqrt{\Delta_x^2 + \Delta_y^2}$
1	0.5	0	0.5
2	-0.592	0.634285714	0.867630317
3	0.442	0.631428571	0.770756797
4	0.592	0.634285714	0.867630317
5	-0.442	0.631428571	0.770756797

Non-orthogonal corrector factor

After calculating the orthogonal component, the non-orthogonal corrector factor $\underline{k}_f$, was calculated for each of the faces as seen in the following table by using equation (2.1):

Table 15: Non-orthogonal corrector factor for each of the CV faces

Faces	k_x	k_y
	$k_x = S_x - \Delta_x$	$k_y = S_y - \Delta_y$
1	0	0
2	0.092	0.065714286
3	-0.192	0.068571429
4	-0.092	0.065714286
5	0.192	0.068571429

Non-orthogonal corrector term

The non-orthogonal corrector term $\underline{k}_f \cdot (\underline{\nabla}\phi)_f$ for each of the faces was calculated by considering the dot product between the non-orthogonal corrector factor and the gradient of ϕ at the faces of the CV. The non-orthogonal corrector term in equation (3.11), for each of the faces, was then calculated as seen in Table 16 and was handled in the source term:

Table 16: Non-orthogonal corrector term for each of the CV faces

Faces	$\underline{k} \cdot (\underline{\nabla}\phi)_f$
	$k_x(\nabla\phi)_{f,x} + k_y(\nabla\phi)_{f,y}$
1	0
2	-7.293062646
3	18.09598084
4	7.293062646
5	-18.09598084

Central and neighbouring coefficients

The numerical representation of the general transport equation for pure diffusion is given by equation (3.26), where the neighbouring coefficients a_{nb} is calculated for each of the faces of the CV, as seen in the following table by using equation (3.28):

Table 17: Neighbouring coefficient for each of the faces of a CV

Face	a_{nb}
	$a_{nb} = \frac{ \underline{\Delta}_f }{ \underline{d}_f }$
1	1.071428571
2	2.537142857
3	1.894285714
4	2.537142857
5	1.894285714

The central coefficient a_p , for the first five CVs were then calculated as seen in Table 18 by using equation (3.27):

Table 18: Calculation of the central coefficient

CV	a_p	$\sum_{nb=1}^n a_{nb}$	S_p
	$a_p = \sum_{nb=1}^n a_{nb} - S_p$	$a_{nb} = \frac{ \underline{\Delta}_f }{ \underline{d}_f }$	$S_p = -\frac{ \underline{\Delta}_f }{ \underline{d}_n }$
1	3.608571429	$a_1 + a_2$	-
2	3.608571429	$a_1 + a_4$	-
3	6.038571429	$a_4 + a_5 + a_6$	-
4	6.038571429	$a_6 + a_7 + a_8$	-
5	4.68	a_7	$-a_{21}$

The boundary conditions and the non-orthogonal corrector term were handled in the matrix source term S_u . The source term was added to the matrix source term for the first five CVs as can be seen in the following table:

Table 19: Source term added to the matrix source

CV	S_u
	$S_u = S_u + \frac{ \underline{\Delta}_f }{ \underline{d}_n } \phi_b$
1	-7.29306265
2	7.29306265
3	-25.3890435
4	25.3890435
5	-7.29306265

OpenFOAM

This section refers to section 4.4 and is used to describe the different dictionary files and the related entries, in more detail.

Points

A point in OpenFOAM is a location in three-dimensional space and is defined by a coordinate vector in metres. Each point is referred to by a label that indicates the position of the point in a numerated list and that starts at 0.

For consistency the first entry in the list was a zero entry so that the second entry “Label 1” resembles node 1 from the Excel calculations.

Table 20: Node coordinates specified in OpenFOAM

Nodes	Node Coordinates
Node 0	(0 0 0)
Node 1	(0 0 0)
Node 2	(0.7 0 0)
Node 3	(1.4 0 0)
Node 4	(2.1 0 0)
Node 5	(0 0.75 0)
Node 6	(0.7 0.5 0)
Node 7	(1.4 0.75 0)
Node 8	(2.1 0.5 0)

Node 9	(0 1.5 0)
Node 10	(0.7 1.0 0)
Node 11	(2.1 1.0 0)
Node 12	(0.7 1.5 0)
Node 13	(1.4 1.5 0)
Node 14	(2.1 1.5 0)
Node 15	(0 0 0)
Node 16	(0 0 0)
Node 17	(0 0 0)
Node 18	(0 0 0)
Node 19	(0 0 0)
Node 20	(0 0 0)
Node 21	(0 0 -1)
Node 22	(0.7 0 -1)
Node 23	(1.4 0 -1)
Node 24	(2.1 0 -1)
Node 25	(0 0.75 -1)
Node 26	(0.7 0.5 -1)
Node 27	(1.4 0.75 -1)
Node 28	(2.1 0.5 -1)
Node 29	(0 1.5 -1)
Node 30	(0.7 1.0 -1)
Node 31	(2.1 1.0 -1)
Node 32	(0.7 1.5 -1)
Node 33	(1.4 1.5 -1)
Node 34	(2.1 1.5 -1)

Faces

In the Faces file, each face is specified by listing the nodes of the face. Each node refers to the label in the Points file for the applicable coordinates of the node.

Each entry states the number of nodes for the specific face (e.g. 4) and then the nodes surrounding the face in brackets. The order of the nodes specified is determined by

the right hand rule (Face 0 refers to face 1 in the Excel calculations). Due to the full three-dimensional mesh used in OpenFOAM, node numbers are shown in Table 21 that do not appear in Figure 26.

Table 21: Faces specified in the Faces file in the polyMesh subdirectory of the Constant directory in OpenFOAM

Faces	Face Nodes
Face 0	4(6 2 22 26)
Face 1	4(1 6 26 21)
Face 2	4(5 6 26 25)
Face 3	4(6 3 23 26)
Face 4	4(6 7 27 26)
Face 5	4(7 3 23 27)
Face 6	4(3 8 28 23)
Face 7	4(7 8 28 27)
Face 8	4(7 11 31 27)
Face 9	4(13 11 31 33)
Face 10	4(13 7 27 33)
Face 11	4(10 7 27 30)
Face 12	4(10 13 33 30)
Face 13	4(10 6 26 30)
Face 14	4(5 10 30 25)
Face 15	4(9 10 30 29)
Face 16	4(10 12 32 30)
Face 17	4(2 1 21 22)
Face 18	4(3 2 22 23)
Face 19	4(4 3 23 24)
Face 20	4(8 4 24 28)
Face 21	4(11 8 28 31)
Face 22	4(14 11 31 34)
Face 23	4(13 14 34 33)
Face 24	4(12 13 33 32)
Face 25	4(9 12 32 29)

Face 26	4(5 9 29 25)
Face 27	4(1 5 25 21)
Face 28	3(1 2 6)
Face 29	3(2 3 6)
Face 30	3(6 3 7)
Face 31	3(3 8 7)
Face 32	3(3 4 8)
Face 33	3(1 6 5)
Face 34	3(5 6 10)
Face 35	3(6 7 10)
Face 36	3(10 7 13)
Face 37	3(7 11 13)
Face 38	3(7 8 11)
Face 39	3(13 11 14)
Face 40	3(10 13 12)
Face 41	3(9 10 12)
Face 42	3(5 10 9)
Face 43	3(21 26 22)
Face 44	3(22 26 23)
Face 45	3(26 27 23)
Face 46	3(23 27 28)
Face 47	3(23 28 24)
Face 48	3(21 25 26)
Face 49	3(25 30 26)
Face 50	3(26 30 27)
Face 51	3(30 33 27)
Face 52	3(27 33 31)
Face 53	3(27 31 28)
Face 54	3(33 34 31)
Face 55	3(30 32 33)
Face 56	3(29 32 30)
Face 57	3(25 29 30)

Owners of faces

The Owner file is a list where the owner of each face, in the Faces file, is specified.

Table 22: Owners of faces specified in the Owners file in the polyMesh subdirectory of the Constant directory

Owner of face	Control Volume
Face 0	0
Face 1	0
Face 2	5
Face 3	1
Face 4	2
Face 5	2
Face 6	4
Face 7	3
Face 8	10
Face 9	9
Face 10	8
Face 11	7
Face 12	8
Face 13	6
Face 14	6
Face 15	14
Face 16	12
Face 17	0
Face 18	1
Face 19	4
Face 20	4
Face 21	10
Face 22	11
Face 23	11
Face 24	12
Face 25	13
Face 26	14

Face 27	5
Face 28	0
Face 29	1
Face 30	2
Face 31	3
Face 32	4
Face 33	5
Face 34	6
Face 35	7
Face 36	8
Face 37	9
Face 38	10
Face 39	11
Face 40	12
Face 41	13
Face 42	14
Face 43	0
Face 44	1
Face 45	2
Face 46	3
Face 47	4
Face 48	5
Face 49	6
Face 50	7
Face 51	8
Face 52	9
Face 53	10
Face 54	11
Face 55	12
Face 56	13
Face 57	14

Neighbours of faces

The Neighbour file is a list of the neighbour CV labels that specify the neighbour of each of the faces specified in the Faces file.

Table 23: Neighbours of faces specified in the Neighbour file in the polyMesh subdirectory of the Constant directory

Neighbour of face	Control Volume
Face 0	1
Face 1	5
Face 2	6
Face 3	2
Face 4	7
Face 5	3
Face 6	3
Face 7	10
Face 8	9
Face 9	11
Face 10	9
Face 11	8
Face 12	12
Face 13	7
Face 14	14
Face 15	13
Face 16	13

Boundaries

The Boundaries dictionary file in OpenFOAM is a list of patches – where a patch is a list of face labels containing only boundary faces. Each patch is declared by defining a patch name.

There were six boundaries that had to be specified. These boundaries were: the adiabatic walls, the fixed temperatures at the front and back of the domain and the front and back sides of the domain. The boundaries were specified by declaring a name for the boundary patch that indicated the number of faces (nFaces) in the patch

and also the face, which refers to the first specified face in the faces list (startFace) for that specific boundary:

```
5
(
  adia_1
  {
    type      patch;
    nFaces    3;
    startFace 17;
  }
  temp_2
  {
    type      patch;
    nFaces    3;
    startFace 20;
  }
  adia_3
  {
    type      patch;
    nFaces    3;
    startFace 23;
  }
  temp_4
  {
    type      patch;
    nFaces    2;
    startFace 26;
  }
  frontBack
  {
    type      patch;
    nFaces    30;
```

```
        startFace    28;  
    }  
)
```

Boundary and initial conditions

The boundary conditions for the domain were specified in the 0 (zero) sub-directory in the Time directory. An initial temperature of 500°C was specified for the internal faces and for the boundary field, the boundary conditions were specified, by referring to the patch name of the boundary in the Boundary file for example:

```
InternalField uniform 500;  
BoundaryField  
Adia_1  
{  
Type: zeroGradient;  
}  
temp_2  
{  
Type: FixedValue  
Value: uniform 400;  
}  
Adia_2  
{  
Type: zeroGradient  
temp_3  
{  
Type: FixedValue  
Value: uniform 600;
```

Appendix B - Excel and OpenFOAM results

The calculated face area vectors are compared in the table below that distinguishes between the Excel and OpenFOAM models.

Table 24: Comparison of the calculated coordinates of the face area vectors between Excel and OpenFOAM

Face	Excel	OpenFOAM
1	(0.5; 0; 0)	(0.5; 0; 0)
2	(-0.5; 0.7; 0)	(-0.5; 0.7; 0)
3	(0.25; 0.7; 0)	(0.25; 0.7; 0)
4	(0.5; 0.7; 0)	(0.5; 0.7; 0)
5	(-0.25; 0.7; 0)	(-0.25; 0.7; 0)
6	(0.75; 0; 0)	(0.75; 0; 0)
7	(-0.5; 0.7; 0)	(-0.5; 0.7; 0)
8	(0.25; 0.7; 0)	(0.25; 0.7; 0)
9	(-0.25; 0.7; 0)	(-0.25; 0.7; 0)
10	(0.5; 0.7; 0)	(0.5; 0.7; 0)
11	(0.75; 0; 0)	(0.75; 0; 0)
12	(0.25; 0.7; 0)	(0.25; 0.7; 0)
13	(-0.5; 0.7; 0)	(-0.5; 0.7; 0)
14	(0.5; 0; 0)	(0.5; 0; 0)
15	(-0.25; 0.7; 0)	(-0.25; 0.7; 0)
16	(0.5; 0.7; 0)	(0.5; 0.7; 0)
17	(-0.5; 0; 0)	(-0.5; 0; 0)
18	(0; -0.7; 0)	(0; -0.7; 0)
19	(0; -0.7; 0)	(0; -0.7; 0)
20	(0; -0.7; 0)	(0; -0.7; 0)
21	(0.5; 0; 0)	(0.5; 0; 0)
22	(0.5; 0; 0)	(0.5; 0; 0)
23	(0; 0.7; 0)	(0; 0.7; 0)
24	(0; 0.7; 0)	(0; 0.7; 0)
25	(0; 0.7; 0)	(0; 0.7; 0)
26	(-0.75; 0; 0)	(-0.75; 0; 0)

27	(-0.75; 0; 0)	(-0.75; 0; 0)
28	(-0.75; 0; 0)	(-0.75; 0; 0)

Table 25 contains the list of values given by Figure 35.

Table 25: Comparison of the magnitudes of the faces area vectors between Excel and OpenFOAM

Face	Excel	OpenFOAM
1	0.5	0.5
2	0.860232527	0.860233
3	0.743303437	0.743303
4	0.860232527	0.860233
5	0.743303437	0.743303
6	0.75	0.75
7	0.860232527	0.860233
8	0.743303437	0.743303
9	0.743303437	0.743303
10	0.860232527	0.860233
11	0.75	0.75
12	0.743303437	0.743303
13	0.860232527	0.860233
14	0.5	0.5
15	0.743303437	0.743303
16	0.860232527	0.860233
17	0.5	0.5
18	0.7	0.7
19	0.7	0.7
20	0.7	0.7
21	0.5	0.5
22	0.5	0.5
23	0.5	0.5
24	0.7	0.7
25	0.7	0.7
26	0.7	0.7

27	0.75	0.75
28	0.75	0.75

Table 26: Comparison of the coordinates of the CV centres between Excel and OpenFOAM

Control Volume	Excel	OpenFOAM
1	(0.466667; 0.166667; 0)	(0.466667; 0.166667; -0.5)
2	(0.933333; 0.166667; 0)	(0.933333; 0.166667; -0.5)
3	(1.166667; 0.416667; 0)	(1.166667; 0.416667; -0.5)
4	(1.633333; 0.416667; 0)	(1.633333; 0.416667; -0.5)
5	(1.866667; 0.166667; 0)	(1.866667; 0.166667; -0.5)
6	(0.233333; 0.416667; 0)	(0.233333; 0.416667; -0.5)
7	(0.466667; 0.75; 0)	(0.466667; 0.75; -0.5)
8	(0.933333; 0.75; 0)	(0.933333; 0.75; -0.5)
9	(1.166667; 1.083333; 0)	(1.166667; 1.083333; -0.5)
10	(1.633333; 1.083333; 0)	(1.633333; 1.083333; -0.5)
11	(1.866667; 0.75; 0)	(1.866667; 0.75; -0.5)
12	(1.866667; 1.333333; 0)	(1.866667; 1.333333; -0.5)
13	(0.933333; 1.333333; 0)	(0.933333; 1.333333; -0.5)
14	(0.466667; 1.333333; 0)	(0.466667; 1.333333; -0.5)
15	(0.233333; 1.333333; 0)	(0.233333; 1.333333; -0.5)
16	(0.35; 0; 0)	(0.35; 0; -0.5)
17	(1.05; 0; 0)	(1.05; 0; -0.5)
18	(1.75; 0; 0)	(1.75; 0; -0.5)
19	(2.1; 0.25; 0)	(2.1; 0.25; -0.5)
20	(2.1; 0.75; 0)	(2.1; 0.75; -0.5)
21	(2.1; 1.25; 0)	(2.1; 1.25; -0.5)
22	(1.75; 1.5; 0)	(1.75; 1.5; -0.5)
23	(1.05; 1.5; 0)	(1.05; 1.5; -0.5)
24	(0.35; 1.5; 0)	(0.35; 1.5; -0.5)
25	(0; 1.125; 0)	(0; 1.125; -0.5)
26	(0; 0.375; 0)	(0; 0.375; -0.5)

Table 26 shows that the only difference between the CV and boundary centres, between the Excel and OpenFOAM models, are the z-coordinate that is 0 in Excel and -0.5 in OpenFOAM. This difference however has no implication on the difference between the calculated temperatures at the centres of the CVs, as an empty condition is specified for the patch which requires no solution.

Volumes of Control Volumes

The calculated volumes of the CVs are compared in Table 27 between the Excel and OpenFOAM models. Table 27 contains the values that were given in Figure 36.

Table 27: Comparison of the calculated volumes between Excel and OpenFOAM

Control Volume	Excel	OpenFOAM
1	0.175	0.175
2	0.175	0.175
3	0.2625	0.2625
4	0.2625	0.2625
5	0.175	0.175
6	0.2625	0.2625
7	0.175	0.175
8	0.175	0.175
9	0.2625	0.2625
10	0.2625	0.2625
11	0.175	0.175
12	0.175	0.175
13	0.175	0.175
14	0.175	0.175
15	0.2625	0.2625

As seen in the table there were no differences between the calculated volumes for the Excel and OpenFOAM models.

Interpolation factors

The interpolation factors for the Excel and OpenFOAM models are shown in Table 28. The differences in the results are due to the different calculation methods that were used in OpenFOAM as were described in Section 5.2.6. Table 28 contains the values that are illustrated in Figure 37 and Figure 38.

Table 28: Comparison of the calculated interpolation factors between Excel and OpenFOAM

Face	L_P		L_N	
	Excel	OpenFOAM	Excel	OpenFOAM
1	0.5	0.5	0.5	0.5
2	0.586603176	0.6	0.413396824	0.4
3	0.417281021	0.4	0.582718979	0.6
4	0.586603176	0.6	0.413396824	0.4
5	0.417281021	0.4	0.582718979	0.6
6	0.5	0.5	0.5	0.5
7	0.586603176	0.6	0.413396824	0.4
8	0.417281021	0.4	0.582718979	0.6
9	0.582718979	0.6	0.417281021	0.4
10	0.413396824	0.4	0.586603176	0.6
11	0.5	0.5	0.5	0.5
12	0.582718979	0.6	0.417281021	0.4
13	0.413396824	0.4	0.586603176	0.6
14	0.5	0.5	0.5	0.5
15	0.582718979	0.6	0.417281021	0.4
16	0.413396824	0.4	0.586603176	0.6
17	0.5	0.5	0.5	0.5

Matrix coefficients

The coefficients for the Excel and OpenFOAM models are shown in Table 29:

Table 29: Comparison between the coefficients of the Excel and OpenFOAM models

Face	Excel	OpenFOAM
1	1.071428571	1.071428571
2	2.537142857	2.537142857
3	1.894285714	1.894285714
4	2.537142857	2.537142857
5	1.894285714	1.894285714
6	1.607142857	1.607142857
7	2.537142857	2.537142857
8	1.894285714	1.894285714
9	1.894285714	1.894285714
10	2.537142857	2.537142857
11	1.607142857	1.607142857
12	1.894285714	1.894285714
13	2.537142857	2.537142857
14	1.071428571	1.071428571
15	1.894285714	1.894285714
16	2.537142857	2.537142857
17	1.071428571	1.071428571
18	4.2	3.440774066
19	4.2	3.440774066
20	4.2	3.440774066
21	2.142857143	2.018018382
22	2.142857143	2.142857143
23	2.142857143	2.018018382
24	4.2	3.440774066
25	4.2	3.440774066
26	4.2	3.440774066
27	3.214285714	3.164231493
28	3.214285714	3.164231493

The reason for the difference found between the matrix coefficients at the boundary faces was discussed in Section 5.2.2.

Appendix C

The Excel model can be viewed on the attached CD found on the back page.

