



UNIVERSITY OF TECHNOLOGY

DIPLOMA THESIS

**Validation and Evaluation of
Numerical Methods for the
Dimensioning of Smoke Control and
Ventilation Systems**

Author:
Daniel STIX B.Sc.

Supervisor:
Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang SANZ
Dipl.-Ing. Reinhard GERTL

October 11, 2013

Acknowledgment

I would like to express my deep gratitude to my diploma thesis supervisor Professor Wolfgang Sanz. He not only guided and helped me through my thesis, but also taught me many things while I have been his student.

I want to thank all employees at ILF Consulting Engineers, who directly or indirectly helped me to finish my diploma thesis. Special thanks are given to Dipl.-Ing Reinhard Gertl. He spent many hours on helping me with my work and reading through this documentation.

I owe more than thanks to my family members, which includes my parents and my younger sister, for their financial support and encouragement throughout my life. Last but not least important, I want to thank my girlfriend for her permanent support and the sometimes necessary motivation.

Abstract

Many different CFD codes enable simulations of technical processes in the field of fluid mechanics. The department for mechanical and electrical engineering at ILF Consulting Engineers ZT GmbH uses three CFD programs to design and dimension smoke control and ventilation systems. Following programs are used: NUMSTA3 (NUMerical Simulation of Tunnel Aerodynamics, developed further by ILF), FDS 5 (Fire Dynamics Simulator) and OpenFOAM (Open Source Field Operation and Manipulation). The objective of this diploma thesis is to determine which of these three programs is suited best for which applications and what errors occur at which program for standard flow problems. Several simple simulations were performed and compared with empirical calculations. After that, a fire test of the Memorial Tunnel Fire Ventilation Test Program is simulated and the results are compared with the measurements. OpenFOAM approved to be the most flexible and versatile software, but simple simulations can be simulated much faster with a one-dimensional CFD program like NUMSTA3. The simulations showed that FDS is not suited best for processes which are dominated by the effects of the boundary layer. However, it allows a simple and fast visualization of spreading smoke and fires. NUMSTA3 is the fastest and most efficient software for simple simulations, where three-dimensional effects are so small that they can be neglected. The heat transfer in NUMSTA3 is not calculated correctly, because a reduction of the heat release rate is used instead of a heat radiation model. Further investigations about radiation heat transfer processes could be useful to determine if the use of a reduced heat release rate instead of a heat radiation model is reasonable. In addition, these investigations could help to develop NUMSTA3 further.

Kurzfassung

Eine Vielzahl an CFD Codes ermöglichen die Simulation von Prozessen in der Strömungsmechanik. Die Abteilung für Maschinenbau und Elektrotechnik der Firma ILF Beratende Ingenieure ZT GmbH verwendet für die Planung und Dimensionierung von Lüftungs- und Entrauchungsanlagen verschiedene CFD Programme. Folgende Programme kommen zum Einsatz: FDS 5 (Fire Dynamics Simulator 5), OpenFOAM (Open Source Field Operation and Manipulation) und das von ILF weiterentwickelte NUMSTA3 (NUMerical Simulation of Tunnel Aerodynamics 3). Die hier vorliegende Diplomarbeit beschäftigt sich damit, die idealen, für ILF relevanten, Anwendungsbereiche für jedes der Programme zu bestimmen. Außerdem werden Fehler bei der Simulation von Standardströmungsprozessen abgeschätzt. Einfache Strömungsprozesse werden simuliert und mit empirisch berechneten Lösungen verglichen. Anschließend wird ein Brandversuch des Memorial Tunnel Fire Ventilation Test Programs simuliert und mit Messergebnissen verglichen. Die Simulationsergebnisse zeigen, dass jedes der Programme seine Stärken und Schwächen hat. OpenFOAM erweist sich als das flexibelste und vielseitigste Programm. Allerdings können sehr einfache Strömungsvoränge viel schneller und einfacher mit NUMSTA3 simuliert werden. FDS 5 ist am schlechtesten für Prozesse geeignet, die stark von der Grenzschicht beeinflusst werden. Jedoch ermöglicht FDS eine einfache und übersichtliche Visualisierung von Rauchausbreitungen und Flammen. NUMSTA3 ist das schnellste und effizienteste Programm zu Simulation von einfachen Geometrien, unter der Annahme eines eindimensionalen Strömungsverhaltens. Wärmeübertragungsprozesse, die von Wärmestrahlung dominiert werden, können mit NUMSTA3 zurzeit nicht sehr genau simuliert werden. Grund dafür ist die Verwendung einer reduzierten Brandlast anstelle eines Strahlungsmodells. Weitere Untersuchungen bezüglich der Wärmeübertragung durch Strahlung könnten die Methode der reduzierten Brandlast hinterfragen und die Weiterentwicklung von NUMSTA3 vorantreiben.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, at October 11, 2013

Daniel STIX B.Sc.



Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Purpose of this Diploma Thesis | 4 |
| 2 | Principles of Fluid Mechanics and Heat Transfer | 5 |
| 2.1 | Basic Equations of Fluid Mechanics | 5 |
| 2.1.1 | Equation of Continuity [5] [4] | 5 |
| 2.1.2 | Conservation of Momentum [5] [4] | 5 |
| 2.1.3 | Conservation of Total Energy [5] | 7 |
| 2.1.4 | Transport Equation of Temperature [5] | 8 |
| 2.1.5 | Basic Equations for Incompressible Fluids [5] | 8 |
| 2.2 | Dimensionless Parameters | 9 |
| 2.2.1 | Reynolds Number [5] | 9 |
| 2.2.2 | More Dimensional Parameters [5] | 10 |
| 2.3 | Turbulent and Laminar Flow [5] [4] | 10 |
| 2.3.1 | DNS - Direct Numerical Solution | 11 |
| 2.3.2 | RANS-Method - Reynolds Averaged Navier-Stokes Method [5] | 11 |
| 2.4 | Boundary Layer [5] [4] | 13 |
| 2.5 | Pressure Loss | 15 |
| 2.5.1 | Static Pressure vs. Total Pressure | 15 |
| 2.5.2 | Pressure Loss of a Pipe Flow [4] | 16 |
| 2.5.3 | Pressure Loss for a Sudden Change in Velocity [9] | 18 |
| 2.5.4 | Pressure Loss for a Change of Flow Direction [9] | 19 |
| 2.6 | Heat Transfer | 21 |
| 2.6.1 | Heat Conduction [6] | 21 |
| 2.6.2 | Heat Convection [5] | 21 |
| 2.6.3 | Heat Radiation [5] | 22 |
| 3 | Basics of Computational Fluid Dynamics | 24 |
| 3.1 | Discretization of the Examined Domain [10] | 24 |
| 3.2 | Spatial and Temporal Discretization [10] [11] | 26 |
| 3.2.1 | Stability of a Numerical Scheme [13] | 28 |
| 3.2.2 | Explicit Euler Scheme [10] | 28 |
| 3.2.3 | Stability of a Unilateral Explicit Euler Scheme [10] [11] | 29 |
| 3.2.4 | Implicit Euler Scheme [10] | 30 |
| 3.2.5 | Stability of an Unilateral Implicit Euler Scheme | 30 |
| 3.2.6 | Combination of Explicit and Implicit Schemes [10] [16] | 31 |



Contents

| | | |
|----------|---|-----------|
| 3.3 | Upwind Schemes [11] | 32 |
| 3.3.1 | Flux Vector Splitting | 32 |
| 3.3.2 | Riemann Solver (Godunov Scheme) | 33 |
| 3.4 | Techniques for Faster Convergence [11] | 34 |
| 3.5 | Turbulence Models [14] | 35 |
| 3.5.1 | Eddy Viscosity Approach | 35 |
| 3.5.2 | k - ω -Model: | 36 |
| 3.5.3 | Standard k - ε -Model: | 37 |
| 3.5.4 | New Baseline Model [15] | 38 |
| 3.5.5 | k - ω -SST Model [15] | 39 |
| 3.5.6 | Large Eddy Simulation - LES [14] | 40 |
| 4 | Description of Used Programs | 43 |
| 4.1 | NUMSTA3 - Numerical Simulation of Tunnel Aerodynamics 3 | 43 |
| 4.2 | OpenFOAM - Open Source Field Operation and Manipulation | 44 |
| 4.3 | FDS 5 - Fire Dynamics Simulator 5 | 46 |
| 5 | Differences between used Software | 49 |
| 5.1 | Geometry | 49 |
| 5.1.1 | Geometry in NUMSTA3 | 49 |
| 5.1.2 | Geometry in OpenFOAM | 50 |
| 5.1.3 | Geometry in FDS | 50 |
| 5.2 | Boundary and Initial Conditions | 51 |
| 5.2.1 | Boundary and Initial Conditions - NUMSTA3 | 52 |
| 5.2.2 | Boundary and Initial Conditions - OpenFOAM | 52 |
| 5.2.3 | Boundary and Initial Conditions - FDS [24] | 52 |
| 5.3 | Calculation of the Pressure Loss in a straight pipe | 55 |
| 5.3.1 | Calculation of the Pressure Loss - NUMSTA3 | 55 |
| 5.3.2 | Calculation of the Pressure Loss - OpenFOAM | 55 |
| 5.3.3 | Calculation of the Pressure Loss - FDS [24] | 55 |
| 5.4 | Consideration of Heat Radiation | 56 |
| 6 | Simulations for Comparing the Programs | 58 |
| 6.1 | Straight Pipe Flow | 58 |
| 6.1.1 | Straight Pipe Flow - Empirical Calculation | 58 |
| 6.1.2 | Straight Pipe Flow - NUMSTA3 | 60 |
| 6.1.3 | Straight Pipe Flow - OpenFOAM | 62 |
| 6.1.4 | Straight Pipe Flow - FDS | 66 |
| 6.1.5 | Straight Pipe Flow - Comparison of Results | 69 |
| 6.1.6 | Straight Pipe Flow - Reasons for Differing Results | 73 |
| 6.2 | Sudden Change of the Cross-Sectional Area | 75 |
| 6.2.1 | Sudden Change of the Cross-Sectional Area - Empirical Calculation | 75 |
| 6.2.2 | Sudden Change of the Cross-Sectional Area - NUMSTA3 | 77 |
| 6.2.3 | Sudden Change of the Cross-Sectional Area - OpenFOAM | 78 |



Contents

| | | |
|----------|--|------------|
| 6.2.4 | Sudden Change of the Cross-Sectional Area - FDS | 84 |
| 6.2.5 | Sudden Change of the Cross-Sectional Area - Comparison of Results | 87 |
| 6.2.6 | Sudden Change of the Cross-Sectional Area - Reasons for Dif- fering Results | 87 |
| 6.3 | Flow in an Elbow | 88 |
| 6.3.1 | Flow in an Elbow - Empirical Calculation | 88 |
| 6.3.2 | Flow in an Elbow - NUMSTA3 | 89 |
| 6.3.3 | Flow in an Elbow - OpenFOAM | 91 |
| 6.3.4 | Flow in an Elbow - FDS | 94 |
| 6.3.5 | Flow in an Elbow - Comparison of Results | 95 |
| 6.3.6 | Flow in an Elbow - Reasons for Differing Results | 95 |
| 6.4 | Heat Source | 102 |
| 6.4.1 | Memorial Tunnel Fire Ventilation Test Program [30] | 102 |
| 6.4.2 | Heat Source - NUMSTA3 | 111 |
| 6.4.3 | Heat Source - OpenFOAM | 115 |
| 6.4.4 | Heat Source - FDS | 119 |
| 6.4.5 | Heat Source - Comparison of the Results | 123 |
| 6.4.6 | Heat Source - Reasons for Deviating Results | 125 |
| 7 | Summary | 127 |
| 8 | Conclusion | 129 |
| | Appendix | I |
| A.1 | FDS: Straight Pipe Flow | I |
| A.2 | FDS: Sudden Change of the Cross-Sectional Area | IV |
| A.3 | FDS: Change of Flow Direction | VII |
| A.4 | FDS: Heat Source | X |

List of Figures

| | | |
|------|---|----|
| 1.1 | Results of a NUMSTA3 simulation | 2 |
| 1.2 | OpenFOAM simulation | 2 |
| 1.3 | FDS simulation model of a train inside a hall | 3 |
| 1.4 | Results of a FDS simulation | 3 |
| | | |
| 2.1 | Conservation of mass for a two-dimensional grid | 6 |
| 2.2 | Balance of momentum in x-direction for a two-dimensional grid | 6 |
| 2.3 | Laminar and turbulent flow [4] | 10 |
| 2.4 | Boundary layer [5] | 13 |
| 2.5 | Layers of the boundary layer [5] | 14 |
| 2.6 | Moody Diagram [8] | 17 |
| 2.7 | Eddy-areas for a sudden change in velocity [9] | 18 |
| 2.8 | Sketch of a sudden cross-sectional change | 19 |
| 2.9 | Eddy-areas for a change of flow direction [9] | 20 |
| 2.10 | Change of flow direction | 20 |
| | | |
| 3.1 | Grid types [10] | 25 |
| 3.2 | Types of grid structures [10] | 26 |
| 3.3 | Trends of variables over the time for different schemes [16] | 32 |
| 3.4 | Schematic presentation of the shock tube flow [11] | 33 |
| 3.5 | Schematic representation of the V-cycle [11] | 34 |
| 3.6 | Modeling degree of different approaches | 42 |
| | | |
| 4.1 | Overview of a NUMSTA3 file | 44 |
| 4.2 | Basic structure of an OpenFOAM case | 45 |
| | | |
| 5.1 | Geometry in NUMSTA | 49 |
| 5.2 | Mesh in NUMSTA | 50 |
| 5.3 | Geometry in OpenFOAM | 50 |
| 5.4 | FDS: define geometry | 51 |
| 5.5 | Round geometry in FDS | 51 |
| 5.6 | Log Law vs. Power Law | 54 |
| | | |
| 6.1 | Straight pipe flow: Examined geometry | 59 |
| 6.2 | Straight pipe flow: Model in NUMSTA3 | 61 |
| 6.3 | Straight pipe flow: Definition of the flow rate | 61 |
| 6.4 | Straight pipe flow: Definition of λ | 61 |
| 6.5 | Straight pipe flow: Pressure loss in NUMSTA3 | 62 |



List of Figures

| | | |
|------|---|----|
| 6.6 | Straight pipe flow: Tunnel's profile created in Gmsh | 63 |
| 6.7 | Straight pipe flow: Tunnel's grid created in enGrid | 64 |
| 6.8 | Straight pipe flow: Flow profile at $U_{input} = 20 \text{ m/s}$, OpenFOAM | 65 |
| 6.9 | Straight pipe flow: Flow profile at $U_{input} = 10 \text{ m/s}$, OpenFOAM | 65 |
| 6.10 | Straight pipe flow: Flow profile at $U_{input} = 5 \text{ m/s}$, OpenFOAM | 66 |
| 6.11 | Straight pipe flow: FDS model | 67 |
| 6.12 | Straight pipe flow: Grid in FDS | 67 |
| 6.13 | Straight pipe flow: Flow profile at $U_{input} = 20 \text{ m/s}$, FDS | 68 |
| 6.14 | Straight pipe flow: Flow profile at $U_{input} = 10 \text{ m/s}$, FDS | 68 |
| 6.15 | Straight pipe flow: Flow profile at $U_{input} = 5 \text{ m/s}$, FDS | 69 |
| 6.16 | Straight pipe flow: Comparison of pressure losses for $U = 20 \text{ m/s}$ | 70 |
| 6.17 | Straight pipe flow: Errors of pressure losses for $U = 20 \text{ m/s}$ | 70 |
| 6.18 | Straight pipe flow: Comparison of pressure losses for $U = 10 \text{ m/s}$ | 71 |
| 6.19 | Straight pipe flow: Errors of pressure losses for $U = 10 \text{ m/s}$ | 71 |
| 6.20 | Straight pipe flow: Comparison of pressure losses for $U = 5 \text{ m/s}$ | 72 |
| 6.21 | Straight pipe flow: Errors of pressure losses for $U = 5 \text{ m/s}$ | 72 |
| 6.22 | Straight pipe flow: y^+ near the wall for $U = 5 \text{ m/s}$ of the original simulation | 74 |
| 6.23 | Straight pipe flow: y^+ near the wall for $U = 5 \text{ m/s}$ of the compare simulation | 75 |
| 6.24 | Sudden change of the cross-sectional area: Examined geometry | 76 |
| 6.25 | Sudden change of the cross-sectional area: model in NUMSTA3 | 77 |
| 6.26 | Sudden change of the cross-sectional area: surfaces in Blender | 79 |
| 6.27 | Sudden change of the cross-sectional area: triangled surfaces in Blender | 79 |
| 6.28 | Sudden change of the cross-sectional area: unstructured mesh | 79 |
| 6.29 | Sudden change of the cross-sectional area: refined cells in the Open- FOAM model | 80 |
| 6.30 | Sudden change of the cross-sectional area: Flow through the examined geometry for an inlet velocity of 20 m/s , OpenFOAM | 81 |
| 6.31 | Sudden change of the cross-sectional area: Flow profile for an inlet velocity of 20 m/s , OpenFOAM | 81 |
| 6.32 | Sudden change of the cross-sectional area: Velocity trend for an inlet velocity of 20 m/s , OpenFOAM | 82 |
| 6.33 | Sudden change of the cross-sectional area: Trend of static pressure for an inlet velocity of 20 m/s | 83 |
| 6.34 | Sudden change of the cross-sectional area: FDS grid | 85 |
| 6.35 | Sudden change of the cross-sectional area: Velocity distribution for an inlet velocity of 20 m/s , FDS | 86 |
| 6.36 | Sudden change of the cross-sectional area: Eddies for an inlet velocity of 20 m/s , FDS | 86 |
| 6.37 | Flow in an Elbow: Examined geometry | 88 |
| 6.38 | Flow in an Elbow: Model in NUMSTA3 | 90 |
| 6.39 | Flow in an Elbow: Triangulated surfaces in Blender | 91 |
| 6.40 | Flow in an Elbow: Unstructured mesh | 92 |
| 6.41 | Flow in an Elbow: Velocity distribution for an inlet velocity of 20 m/s , OpenFOAM | 93 |



List of Figures

| | | |
|------|---|-----|
| 6.42 | Flow in an Elbow: Total pressure distribution in OpenFOAM for an inlet velocity of 20 m/s | 93 |
| 6.43 | Flow in an Elbow: geometry in Blender | 94 |
| 6.44 | Flow in an Elbow: Grid in FDS | 94 |
| 6.45 | Generation of a polyhedral mesh | 97 |
| 6.46 | Flow in an Elbow: Sketch of a tetrahedral grid | 98 |
| 6.47 | Flow in an Elbow: Sketch of a upwind scheme | 99 |
| 6.48 | Cells of a numeric grid | 100 |
| 6.49 | Heat Source: Cross-section of Memorial Tunnel [30] | 102 |
| 6.50 | Heat Source: Fan group | 104 |
| 6.51 | Heat Source: Measurement Tree [30] | 105 |
| 6.52 | Heat Source: Basic setup for Longitudinal Ventilation Tests | 107 |
| 6.53 | Heat Source: Initial conditions for the simulation [30] | 109 |
| 6.54 | Heat Source: Comparative data for the simulations [30] | 110 |
| 6.55 | Heat Source: NUMSTA3 model | 111 |
| 6.56 | Heat Source: Temperature trend of the transient NUMSTA3 simulation | 113 |
| 6.57 | Heat Source: Temperature trend of the steady NUMSTA3 simulation | 113 |
| 6.58 | Heat Source: Geometry in Gmesh | 115 |
| 6.59 | Heat Source: Geometry in enGrid | 116 |
| 6.60 | Heat Source: Orientation of the gravity vector | 116 |
| 6.61 | Heat Source: Results of the simulation in OpenFOAM | 118 |
| 6.62 | Heat Source: Geometry in Blender | 119 |
| 6.63 | Heat Source: FDS model | 120 |
| 6.64 | Heat Source: Fans in FDS | 120 |
| 6.65 | Heat Source: Results of the simulation in FDS | 121 |
| 6.66 | Spreading of smoke in FDS | 122 |
| 6.67 | Velocity profiles at several Loops | 124 |
| 6.68 | Heat Source: Comparison between the temperature trend of NUMSTA3 and OpenFOAM | 125 |
| 6.69 | Heat Source: Different geometries in OpenFOAM and FDS | 126 |

List of Tables

| | | |
|------|---|-----|
| 6.1 | Straight pipe flow: Pressure losses, empirical | 60 |
| 6.2 | Straight pipe flow: Pressure losses, NUMSTA3 | 60 |
| 6.3 | Straight pipe flow: Transformed pressure losses, OpenFOAM | 66 |
| 6.4 | Straight pipe flow: Pressure losses, FDS | 68 |
| 6.5 | Straight pipe flow: Required computing time of the programs | 69 |
| 6.6 | Straight pipe flow: Pressure losses for refined meshes, OpenFOAM | 74 |
| 6.7 | Sudden change of the cross-sectional area: Pressure losses, empirical | 76 |
| 6.8 | Sudden change of the cross-sectional area: Pressure losses, NUMSTA3 | 78 |
| 6.9 | Sudden change of the cross-sectional area: Reference pressure losses, empirical | 84 |
| 6.10 | Sudden change of the cross-sectional area: Reference pressure losses, NUMSTA3 | 84 |
| 6.11 | Sudden change of the cross-sectional area: Pressure losses, OpenFOAM | 84 |
| 6.12 | Sudden change of the cross-sectional area: Pressure losses, FDS | 87 |
| 6.13 | Sudden change of the cross-sectional area: : summarized results | 87 |
| 6.14 | Flow in an Elbow: Pressure losses, empirical | 89 |
| 6.15 | Flow in an Elbow: Pressure losses, NUMSTA3 | 90 |
| 6.16 | Flow in an Elbow: Pressure losses, OpenFOAM | 92 |
| 6.17 | Flow in an Elbow: Pressure losses, FDS | 95 |
| 6.18 | Flow in an Elbow: Summarized results | 95 |
| 6.19 | Flow in an Elbow: Results for refined mesh, OpenFOAM | 95 |
| 6.20 | Flow in an Elbow: Results for polyhedral mesh, OpenFOAM | 98 |
| 6.21 | Flow in an Elbow: Results for polyhedral mesh, OpenFOAM | 101 |
| 6.22 | Heat Release: Elevation Levels of sensors | 105 |
| 6.23 | Heat Release: Position of the measurement trees in longitudinal direction | 106 |
| 6.24 | Heat Release: Fan Operating Time | 107 |
| 7.1 | Strengths/Weaknesses of the used software | 128 |

Used Symbols

| Symbol | Meaning | Unit |
|---------------------|---|---------------------------------|
| * | denotes dimensionless quantities | - |
| ' | denotes fluctuating quantities | - |
| $\langle \ \rangle$ | time averaged value | - |
| α | heat transfer coefficient | $\text{W}/\text{m}^2\text{K}$ |
| α | turbulence closure coefficient | - |
| a | constant | - |
| a | height of a rectangular cross section | m |
| a_1 | proportionality factor | - |
| A | area | m^2 |
| A | function of bow angle | - |
| b | width of a rectangular cross section | m |
| β | degree of implicity | - |
| β | turbulence closure coefficient | - |
| β_o | turbulence closure coefficient | - |
| β^* | turbulence closure coefficient | - |
| B | empiric constant | - |
| c | sonic speed | m/s |
| c | velocity | m/s |
| c | wave velocity | <i>/unitfracms</i> |
| c_j | numerical error | variable |
| C | empirical constant | - |
| C_1 | geometry factor | - |
| C_1 | constant | Wm^2 |
| C_2 | constant | mK |
| C_S | Stefan-Boltzmann Constant | $\text{W}/\text{m}^2\text{K}^4$ |
| C_S | Smagorinsky coefficient | - |
| CFL | Courant Number | - |
| C_D | turbulence closure coefficient | - |
| C_{lim} | constant | - |
| C_μ | turbulence closure coefficient | - |
| $C_{\varepsilon 1}$ | turbulence closure coefficient | - |
| $C_{\varepsilon 2}$ | turbulence closure coefficient | - |
| C_{ij} | cross-term stress tensor | m^2/s^2 |
| c_p | specific heat capacity at constant pressure | J/kgK |

Continued on next page

List of Tables

| | | |
|---------------|---|--------------------------------|
| c_v | specific heat capacity at constant volume | J/kgK |
| χ_ω | turbulence closure coefficient | - |
| δ | thickness of the boundary layer | m |
| δ | angle of a bow | ° |
| δ_{ij} | auxiliary matrix | - |
| δ_Z | height of the grid cell in FDS | m |
| Δ | filter width | m |
| d | diameter | m |
| d | thickness of the boundary layer | m |
| d | hydraulic diameter | m |
| D | diameter | m |
| D_0 | reference diffusion coefficient | s/m ² |
| D_{hydr} | hydraulic diameter | m |
| ε | emissivity | - |
| ε | dissipation of kinetic energy | m ² s ³ |
| e | energy | J |
| e | wall roughness | m |
| e_s | emission of a black body | W/m ² |
| Ec | Eckert Number | - |
| $f_{i,j,k}^b$ | body force | N |
| f | vector of boundary conditions | variable |
| f | general function | variable |
| f_β | turbulence closure coefficient | - |
| F | force | N |
| F | area | m ² |
| F | general function | variable |
| F_1 | blending function | - |
| F_2 | blending function | - |
| Fr | Froude Number | - |
| F_S | thrust | N |
| $g_{i,j,k}$ | gravity component in i,j,k direction | m/s ² |
| g | gravity | m/s ² |
| G | filter function | - |
| G | intensification factor | - |
| h | increment | m |
| H | enthalpy | J/kg |
| HRR | heat release rate | W |
| HRR_{eff} | effective heat release rate | W |
| I | imaginary unit | - |
| κ | isentropic exponent | - |
| κ | Karman-Constant | - |
| k | turbulent kinetic energy | m ² /s ² |
| k | number of cell edges | - |

Continued on next page

List of Tables

| | | |
|----------------|--|--------------|
| K | cumulative matrix | - |
| λ | thermal conductivity | W/mK |
| λ | pipe drag coefficient | - |
| λ | wavelength | m |
| λ_m | wavelength at maximum of spectral energy | m |
| l | pipe length | m |
| l | turbulent mixing length | m |
| L | characteristic length | m |
| L_0 | reference length | m |
| $L(\Phi)$ | discretization function | variable |
| L_{ij} | Leonard-stress term | m^2/s^2 |
| μ | absolute viscosity | kg/ms |
| μ_0 | reference absolute viscosity | kg/ms |
| m | empirical constant | - |
| \dot{m} | mass flux | kg/s |
| Ma | Mach Number | - |
| M_{in} | momentum entering the examined domain | kgm/s |
| M_{out} | momentum exiting the examined domain | kgm/s |
| ν | kinematic viscosity | m^2/s |
| ν_0 | reference kinematic viscosity | m^2/s |
| ν_T | eddy viscosity | m^2/s |
| n | empirical constant | - |
| O | surface | m^2 |
| \vec{n} | vector in normal direction | m |
| $\vec{\nabla}$ | nabla operator | 1/m |
| Nu | Nusselt Number | - |
| ω | rate of dissipation of energy per unit volume and time | 1/s |
| Ω | mean-rotation | 1/s |
| $O(\Delta x)$ | error of first order | variable |
| Φ | heat performance caused by friction | W |
| Φ | vector of unknown function values | variable |
| Φ | general function | variable |
| p | pressure | N/m^2 (Pa) |
| p_{tot} | total pressure | N/m^2 (Pa) |
| p_{stat} | static pressure | N/m^2 (Pa) |
| p_{dyn} | dynamic pressure | N/m^2 (Pa) |
| Δp_V | pressure loss | N/m^2 (Pa) |
| P | pressure | N/m^2 (Pa) |
| Pr | Prandtl Number | - |
| q | transmitted heat performance | W/m |
| q | heat flow density | W/m^2 |

Continued on next page

| | | |
|----------------------|--------------------------------------|-------------------------|
| \dot{Q} | heat performance of a heat source | W |
| Q_{ij} | subgrid-scale stress tensor | m^2/s^2 |
| ρ | density | kg/m^3 |
| ρ_0 | reference density | kg/m^3 |
| R | gas constant | J/kgK |
| Re | Reynolds Number | - |
| R_{ij} | Reynolds Stress Tensor | m^2/s^2 |
| σ | stress tensor | N/m^2 |
| σ | turbulence closure coefficient | - |
| σ^* | turbulence closure coefficient | - |
| σ_{do} | turbulence closure coefficient | - |
| σ_ε | turbulence closure coefficient | - |
| σ_k | turbulence closure coefficient | - |
| S | stress tensor | N/m^2 |
| Sc | Schmidt Number | - |
| τ | shear stress | N/m^2 |
| τ_{lam} | laminar shear stresses | N/m^2 |
| τ_{turb} | turbulent shear stresses | N/m^2 |
| τ_w | wall shear stresses | N/m^2 |
| Θ | local temperature difference | K |
| T | temperature | K |
| T_0 | reference temperature | K |
| T_∞ | inflow temperature | K |
| T_w | wall temperature | K |
| t | time | s |
| du | differential velocity in i direction | m/s |
| u | velocity in i direction | m/s |
| u^+ | nondimensional streamwise velocity | - |
| U | perimeter | m |
| \bar{U} | mean velocity | m/s |
| U_0 | reference velocity | m/s |
| U_∞ | inflow velocity | m/s |
| $u_{i,j,k}$ | velocity in i,j,k direction | m/s |
| u_m | mean velocity | m/s |
| u_τ | shear stress velocity | m/s |
| v | velocity in j direction | m/s |
| v | velocity | m/s |
| ν | Courant Number | - |
| \dot{V} | flow rate | m^3/s |
| w | index of quantities at walls | - |
| w | velocity in k direction | m/s |
| W_∞ | inflow velocity | m/s |
| dx | differential distance in i direction | m |

Continued on next page



List of Tables

| | | |
|-------------|--|---|
| x | distance in i direction | m |
| x_E | lead-in length | m |
| $x_{i,j,k}$ | distance in i,j,k direction | m |
| ξ | coefficient of drag | - |
| ξ | coordinate | m |
| ξ_{fr} | coefficient of drag caused by friction | - |
| ξ_{loc} | local coefficient of drag | - |
| ξ_M | coefficient of drag caused by a bow | - |
| dy | differential distance in j direction | m |
| y | distance in j direction | m |
| y^+ | dimensionless wall distance | - |
| z | distance in k direction | m |
| z | geodetic height | m |
| z | coordinate normal to the wall | m |

1 Introduction

At the current state of technology, the computer is an indispensable tool for the calculation of technical processes. Particularly in the field of fluid mechanics, physical processes are difficult or impossible to calculate without the help of computers. Computational fluid dynamics (CFD) is an established method to approximately solve fluid mechanic problems with numerical methods. The fast increase of computer performance in the last years enabled the simulation of complex flow processes. CFD codes are developed further all the time in order to consider more and more physical effects in computer simulations.

1.1 Motivation

The department for mechanical and electrical engineering at ILF Consulting Engineers is among others concerned with smoke control and ventilation systems. Various CFD calculations are regularly needed in order to dimension and optimize these systems. A one-dimensional CFD code has been developed by Alexander Rudolf in the course of his dissertation [1]. This CFD code has been developed further by ILF Consulting Engineers to enable a fast and efficient simulation of ventilation and smoke control systems. ILF Consulting Engineers uses two other programs besides NUMSTA3 for the simulation of flows: FDS 5 and . All these programs have been used for projects at ILF. Small calculations are performed on local workstations, while large calculations are processed on a cluster.

Some guidelines like the RVS [2] or the RABT [3] dictate that in the case of a fire the velocity in tunnels must not fall below a certain value. NUMSTA3 can be used to simulate velocities in tunnels. An example for the velocities in a tunnel and in exhaust ducts of a NUMSTA3 simulation is shown in Figure 1.1. OpenFOAM or FDS could also be used to determine the velocity in a tunnel. So why is NUMSTA3 used?

The goal of an other project was to investigate a flow entering a fan. Because of structural conditions, the fan was situated right after an elbow. Directing shovels were placed in the elbow to enable a homogenous inflow into the fan. To investigate the behavior of the flow in the elbow OpenFOAM has been used. Figure 1.2 shows the flow velocity distribution in the elbow. The same process could have been simulated in FDS.

In other projects the smoke spreading and temperature distribution in the case of a fire was investigated. The soot extinction coefficient and the maximum air tempera-

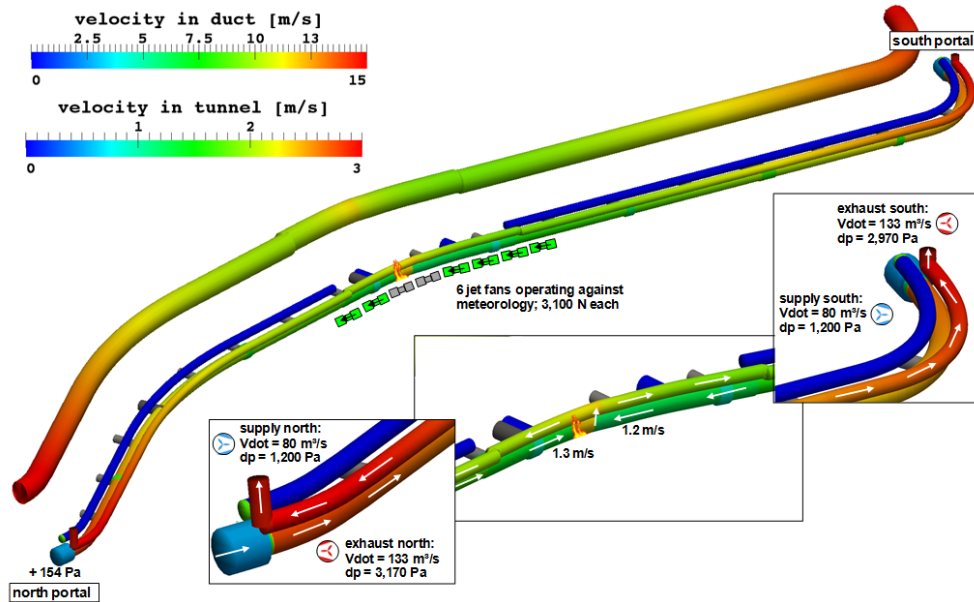


Figure 1.1: Results of a NUMSTA3 simulation

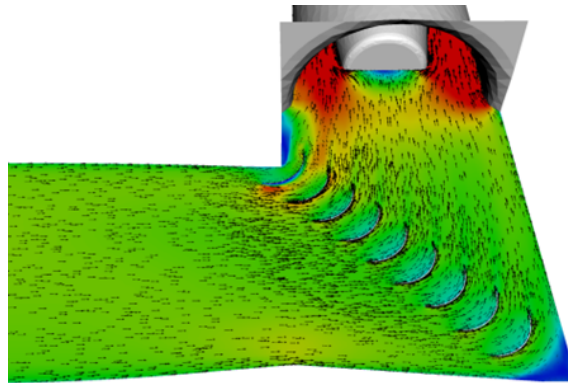


Figure 1.2: OpenFOAM simulation

ture in evacuation paths are limited. FDS has been used to determine the extinction coefficient and the temperature in evacuation paths. Figure 1.3 shows the simulation model of a train inside a hall. In this simulation a fire is placed underneath the train. The fire produces smoke and heat. Figure 1.4 shows the soot extinction coefficient in a height of 2.5 m above the ground level of the evacuation path.

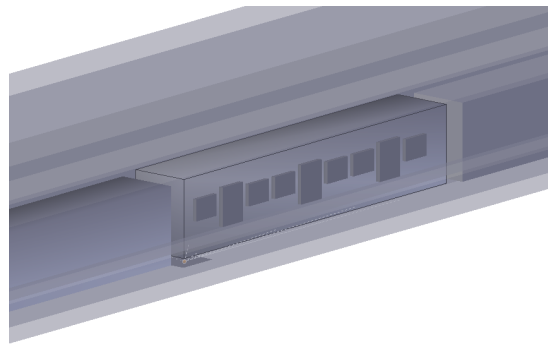


Figure 1.3: FDS simulation model of a train inside a hall

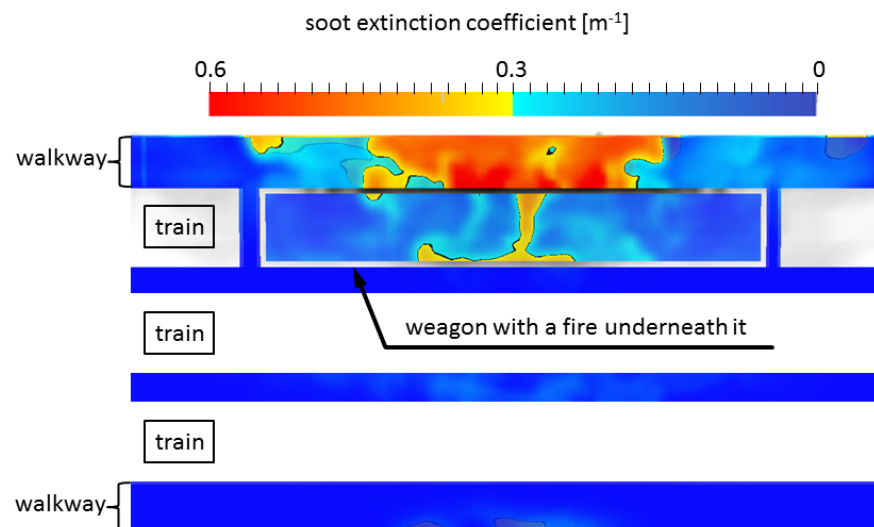


Figure 1.4: Results of a FDS simulation

1.2 Purpose of this Diploma Thesis

Each of the above mentioned programs has its strengths and weaknesses. Therefore, two questions arise, which will be answered in the course of this diploma thesis:

Which program is suited best for which application? What errors occur for standard flow problems?

To find an answer to these questions, several comparative calculations will be performed. These calculations will show the advantages and disadvantages of the individual programs and will give an estimation of the numerical error for each program.

In Chapter 2, all the necessary basics of fluid mechanics will be explained. Especially the basic equations, the Navier-Stokes Equations, are important to understand the basics behind CFD methods. Particularly turbulent flows require the help of computers to solve the Navier-Stokes Equations, since a closure problem occurs while solving them.

In order to understand how computer software solves the arising partial differential equations, the basics of computational fluid dynamics are summarized in Chapter 3. Possible methods for the numerical approximation of the Navier-Stokes Equations are mentioned, and used methods will be explained in more detail. One of the most important points is the solution of the closure problem.

Since all used programs are different, they will be described in a few words in Chapter 4 and their differences will be explained in Chapter 5.

Afterwards a variation of test cases will be performed in each program in Chapter 6. The test cases will be the same for all three programs and their solution will be compared to empirical solutions and experimental data. Based on the results, the strengths and weaknesses of each program will be derived, and relevant ambit will be provided to ILF Consulting Engineers.

2 Principles of Fluid Mechanics and Heat Transfer

Fluid Mechanics deals with the behavior of fluids, which are infinitely deformed under the influence of shear forces. The forces to deform a liquid body become zero, if the velocity of deformation becomes zero. This characteristic is used for the definition of a fluid. [5] [4]

2.1 Basic Equations of Fluid Mechanics

In contrast to the mechanics of solid bodies, in the fluid mechanics the volume of the control mass consists of particles, which are moving through the flow field. Because of that fact, the control volume moved by the flow is time-dependent. The conservation equations are describing the total temporal change of three physical quantities in the considered control volume:

- the mass
- the momentum
- and the energy

2.1.1 Equation of Continuity [5] [4]

Axiom: Mass can be neither produced nor eliminated.

$$\frac{\partial \rho}{\partial t} + \frac{\partial u_j \rho}{\partial x_j} = 0 \quad (2.1)$$

or in words:

$$\left(\begin{array}{l} \text{temporal change of the mass} \\ \text{inside of the control volume} \end{array} \right) + \left(\begin{array}{l} \text{convective transport} \\ \text{of the mass} \\ \text{over the} \\ \text{system boundaries} \end{array} \right) = 0$$

2.1.2 Conservation of Momentum [5] [4]

Axiom: The temporal change of a body's momentum is equal to the sum of all surface- and volume forces influencing the body.

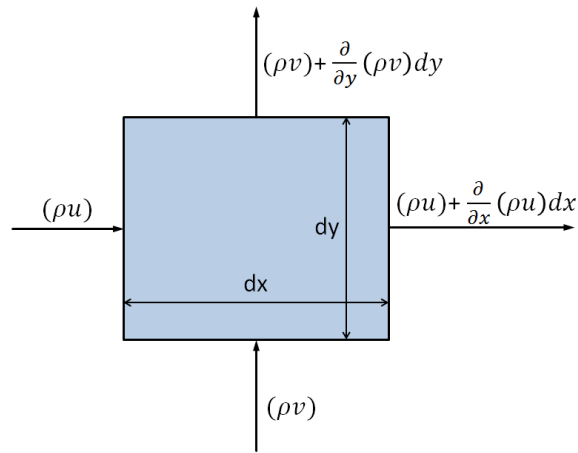


Figure 2.1: Conservation of mass for a two-dimensional grid

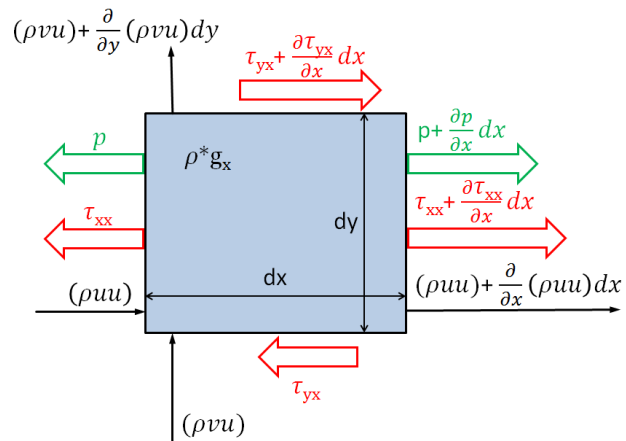


Figure 2.2: Balance of momentum in x-direction for a two-dimensional grid

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i^b \quad (2.2)$$

or in words:

$$\left(\begin{array}{c} \text{temporal change} \\ \text{of the momentum} \\ \text{inside of the} \\ \text{control volume} \end{array} \right) + \left(\begin{array}{c} \text{convective transport} \\ \text{of the momentum} \\ \text{over the} \\ \text{system boundaries} \end{array} \right) = \sum \mathbf{F}$$

The variable σ_{ij} in Equation 2.2 is the stress tensor.

$$\sigma_{ij} = -p\delta_{ij} - \frac{2}{3}\mu S_{kk}\delta_{ij} + 2\mu S_{ij} \quad (2.3)$$

$$\text{with: } S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.4)$$

$$S_{kk} = \frac{\partial u_k}{\partial x_k} = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} \quad (2.5)$$

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \quad (2.6)$$

By using Equation 2.3 and 2.4 in Equation 2.2:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left[-p\delta_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \rho f_i^b \quad (2.7)$$

This form of the momentum equation is the **Navier-Stokes Equation**. The basic function of fluid mechanics is to solve the Navier-Stokes Equation in combination with the Equation of Continuity.

2.1.3 Conservation of Total Energy [5]

Axiom: The temporal change of a body's total energy (kinetic and internal energy) is equal to the sum of all performances of forces which influence the body, the heat performance transported by heat conduction and the heat supply by heat sources.

$$\frac{\partial \rho \left(e + \frac{1}{2} u_k u_k \right)}{\partial t} + \frac{\partial \rho u_j \left(e + \frac{1}{2} u_k u_k \right)}{\partial x_j} = \frac{\partial \sigma_{ji} u_i}{\partial x_j} + \rho u_j f_i^b - \frac{\partial q_j}{\partial x_j} + \dot{Q} \quad (2.8)$$

or in words:

$$\left(\begin{array}{c} \text{temporal change} \\ \text{of the energy} \\ \text{inside of the} \\ \text{control volume} \end{array} \right) + \left(\begin{array}{c} \text{convective transport} \\ \text{of the energy} \\ \text{over the} \\ \text{system boundaries} \end{array} \right) =$$

$$\left(\begin{array}{c} \text{performance} \\ \text{of external} \\ \text{forces} \end{array} \right) - \left(\begin{array}{c} \text{transmitted} \\ \text{heat performance} \end{array} \right) + \left(\begin{array}{c} \text{transferred} \\ \text{heat performance} \\ \text{caused from} \\ \text{internal heat sources} \end{array} \right)$$

2.1.4 Transport Equation of Temperature [5]

The transport equation of temperature is a special form of Equation 2.8 with the assumption of an ideal gas, where:

$$e = c_v T \quad (2.9)$$

and:

$$c_v = \text{const.} \quad (2.10)$$

as well as:

$$q_j = -\lambda \frac{\partial T}{\partial x_j} \quad (\text{where } \lambda = \text{const.}) \quad (2.11)$$

Equation 2.11 is known as **Fourier's Law of Conduction**. So the **transport equation of temperature** is:

$$\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} = \frac{p}{\rho c_v} \frac{\partial u_j}{\partial x_j} + \frac{\lambda}{\rho c_v} \frac{\partial^2 T}{\partial x_j \partial x_j} + \frac{\Phi_\mu}{\rho c_v} + \frac{\dot{Q}}{\rho c_v} \quad (2.12)$$

2.1.5 Basic Equations for Incompressible Fluids [5]

Not only water but also air can be treated incompressible, even if only under a certain precondition. The important influence factor is the **Mach Number**.

$$Ma = \frac{v}{c} \quad (2.13)$$

$$c = \sqrt{\kappa(T)RT} \quad (2.14)$$

If $Ma > 0.3$ there are many compressible effects and the flow must be treated compressible [25]. But these effects are small if $Ma < 0.3$ and the flow can be treated incompressible. In the course of this diploma thesis only low flow speeds will be examined (for the reason look at Chapter 6.1.1) and all flows can be treated incompressible.

$$\text{Continuity: } \frac{\partial u_j}{\partial x_j} = 0 \quad (2.15)$$

$$\text{Momentum: } \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} \left(\nu \frac{\partial u_i}{\partial x_j} \right) + f_i^b \quad (2.16)$$

$$\text{Energy: } \frac{\partial \rho \left(e + \frac{1}{2} u_k u_k \right)}{\partial t} + \frac{\partial \rho u_j \left(e + \frac{1}{2} u_k u_k \right)}{\partial x_j} = \frac{\partial \sigma_{ji} u_i}{\partial x_j} + \rho u_j f_i^b - \frac{\partial q_j}{\partial x_j} + \dot{Q} \quad (2.17)$$

Energy equation in temperature formulation:

$$\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} = \frac{\lambda}{\rho c_v} \frac{\partial^2 T}{\partial x_j \partial x_j} + \frac{\Phi_\mu}{\rho c_v} + \frac{\dot{Q}}{\rho c_v} \quad (2.18)$$

$$\Phi_\mu = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} \quad (2.19)$$

2.2 Dimensionless Parameters

Dimensionless parameters are parameters in a dimensionless mathematical model. If two processes are described by the same dimensionless mathematical model and the dimensionless parameters of these two processes have the same value, all physical quantities of the first system can be calculated by using the physical quantities of the second system and a translation rule.[7]

2.2.1 Reynolds Number [5]

The Reynolds Number is named after Osborne Reynolds, a British physicist. This number is dimensionless and relevant to characterize flows, e.g. to differentiate between laminar and turbulent flows.

For the definition of a dimensionless parameter it is necessary to transform the **Navier-Stokes Equation** or any other equation of conservation to a dimensionless form by using the following equations:

$$x_i^* = \frac{x_i}{L_0} \quad (2.20)$$

$$t^* = \frac{tU_0}{L_0} \quad (2.21)$$

$$u_i^* = \frac{u_i}{U_0} \quad (2.22)$$

$$\rho^* = \frac{\rho}{\rho_0} \quad (2.23)$$

$$p^* = \frac{p}{\rho_0 U_0^2} \quad (2.24)$$

$$T^* = \frac{\Delta T}{T_w - T_\infty} = \frac{T - T_0}{\Delta T_0} \quad (2.25)$$

$$g_i^* = \frac{g_i}{g} \quad (2.26)$$

After putting Equation 2.20 to 2.26 into Equation 2.7, the dimensionless Navier-Stokes Equation follows (gravity has been neglected):

$$\frac{\partial u_i^*}{\partial t^*} + u_j^* \frac{\partial u_i^*}{\partial x_j^*} = -\frac{1}{\rho^*} \frac{\partial p^*}{\partial x_i^*} + \frac{\mu_0}{\rho_0 U_0 L_0} \frac{1}{\rho^*} \frac{\partial}{\partial x_j^*} \left[\left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} \right) - \frac{2}{3} \frac{\partial u_k^*}{\partial x_k^*} \delta_{ij} \right] \quad (2.27)$$

$$\text{Reynolds Number: } Re = \frac{\rho_0 U_0 L_0}{\mu_0} = \frac{U_0 L_0}{\nu_0} \hat{=} \frac{\text{inertia forces}}{\text{viscosity forces}} \quad (2.28)$$

2.2.2 More Dimensional Parameters [5]

$$\text{Froude Number: } Fr = \frac{U_0}{\sqrt{gL_0}} \cong \frac{\text{inertia forces}}{\text{gravity forces}} \quad (2.29)$$

$$\text{Prandtl Number: } Pr = \frac{\mu_0 c_p}{\lambda_0} \cong \frac{\text{velocity diffusion}}{\text{temperature diffusion}} \quad (2.30)$$

$$\text{Eckert Number: } Ec = \frac{U_0^2}{c_p \Delta T_0} \cong \frac{\text{kinetic energy}}{\text{enthalpy difference}} \quad (2.31)$$

$$\text{Schmidt Number: } Sc = \frac{\nu_0}{D_0} \cong \frac{\text{velocity diffusion}}{\text{substance diffusion}} \quad (2.32)$$

2.3 Turbulent and Laminar Flow [5] [4]

A laminar flow is a flow of liquids or gases, which contains no visible turbulences. That means that no crossflows occur and that the fluid flows in layers that do not mix with each other.

Turbulent flows are characterized by random fluctuations of velocity and pressure (Figure 2.3). Their existence has been realized by Leonardo da Vinci, who sketched the eddy structure of turbulent water flows and the appearance of turbulence bales. A proof that laminar flows can change suddenly into turbulent flows has been given

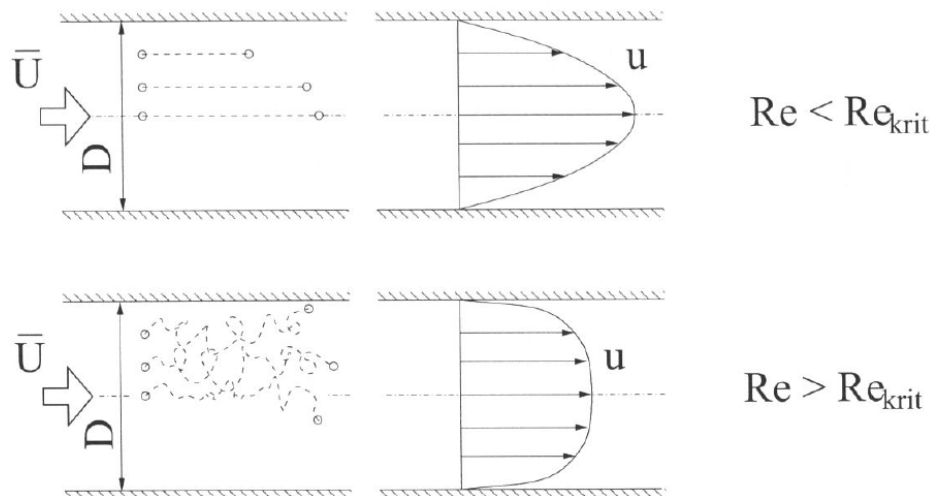


Figure 2.3: Laminar and turbulent flow [4]

by Osborne Reynold 1883 with his dye injection experiment. If $Re > Re_{critical}$, the laminar flow suddenly changes into a turbulent flow and the viscosity of the fluid

seems to be higher \rightarrow the drag rises. This “higher viscosity” causes different flow profiles in a laminar and turbulent flow. The critical Reynolds Number is determined experimentally. The critical Reynolds Number for pipe flows is $Re_{crit} \approx 2300$.

2.3.1 DNS - Direct Numerical Solution

The Navier Stokes Equations are describing laminar and turbulent flows. But solving the Navier Stokes Equations for turbulent flows requires numerical methods with very high resolution for time and space. This method is called **DNS** (**D**irect **N**umerical **S**olution). In most cases, the computing effort is too high. For that reason the fluctuations must be modeled somehow (there aren't any universal theories).

2.3.2 RANS-Method - Reynolds Averaged Navier-Stokes Method [5]

The basic idea behind many turbulence models are the **Reynolds Equations**. The turbulent movement of each particle is split up in an averaged movement and a fluctuating movement. The result of a subsequent temporal averaging is information about temporal averaged quantities of the turbulent flow.

$$u_i = \langle u_i \rangle + u'_i \tag{2.33}$$

$$\left(\begin{array}{c} \text{value at} \\ \text{the current} \\ \text{time} \end{array} \right) = \left(\begin{array}{c} \text{average value} \end{array} \right) + \left(\begin{array}{c} \text{fluctuation} \end{array} \right)$$

Note that:

$$\begin{aligned} \langle \langle u_i \rangle \rangle &= \langle u_i \rangle \\ \langle u_i + u_j \rangle &= \langle u_i \rangle + \langle u_j \rangle \\ \langle a u_i \rangle &= a \langle u_i \rangle \quad \text{for: } a = \text{const} \\ \langle u'_i \rangle &= 0 \end{aligned}$$

but: $\langle u'_i u'_j \rangle \neq \langle u'_i \rangle \langle u'_j \rangle \neq 0$

Reynolds averaging of the Equation of Continuity (Equation 2.15):

$$\left\langle \frac{\partial u_j}{\partial x_j} \right\rangle = \frac{\partial \langle u_j + u'_j \rangle}{\partial x_j} = \frac{\partial \langle u_j \rangle}{\partial x_j} \tag{2.34}$$

Reynolds averaging of the Equation of Momentum (Equation 2.16):

$$\frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial \langle u_j u_i \rangle}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j \partial x_j} \tag{2.35}$$

The convection term in Equation 2.35 is not linear and causes problems:

$$\begin{aligned} \langle u_i u_j \rangle &= \langle (\langle u_i \rangle + u'_i)(\langle u_j \rangle + u'_j) \rangle \\ &= \langle \langle u_i \rangle \langle u_j \rangle + \langle u_i \rangle u'_j + u'_i \langle u_j \rangle + u'_i u'_j \rangle \\ &= \underbrace{\langle u_i \rangle \langle u_j \rangle}_{\text{Product of means}} + \underbrace{\langle u'_i u'_j \rangle}_{\text{Covariance}} \end{aligned}$$

So the averaged momentum equation is:

$$\frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial \langle u_j \rangle \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u'_i u'_j \rangle}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j \partial x_j} \quad (2.36)$$

The third term in this equation has the same effect as shear stresses. That's why the term often is written on the right side of the equation:

$$\frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial \langle u_j \rangle \langle u_i \rangle}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j \partial x_j} - \frac{\partial \langle u'_i u'_j \rangle}{\partial x_j} \quad (2.37)$$

The covariance of the velocity is also called **Reynolds Stress Tensor**:

$$R_{ij} = \begin{pmatrix} \langle u'_1 u'_1 \rangle & \langle u'_1 u'_2 \rangle & \langle u'_1 u'_3 \rangle \\ \langle u'_2 u'_1 \rangle & \langle u'_2 u'_2 \rangle & \langle u'_2 u'_3 \rangle \\ \langle u'_3 u'_1 \rangle & \langle u'_3 u'_2 \rangle & \langle u'_3 u'_3 \rangle \end{pmatrix} \hat{=} \text{seemingly shear stresses} \quad (2.38)$$

Basically there are only two possibilities to determine R_{ij} and solve the Reynolds Equations:

- Modeling R_{ij}
- Reynolds Stress Transport (RST)

Modeling the Reynolds Stress Tensor

Usually, the Eddy Viscosity Approach is used:

$$R_{ij} = \langle u'_i u'_j \rangle = \frac{2}{3} k \delta_{ij} - \nu_T \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) \quad (2.39)$$

with turbulent kinetic energy: $k = \frac{\langle u'_1 u'_1 \rangle + \langle u'_2 u'_2 \rangle + \langle u'_3 u'_3 \rangle}{2} \quad (2.40)$

ν_T is called eddy viscosity and must be modeled.

Reynolds Stress Transport

Separate transport equations are used for R_{ij} . This is a very complex method, because up to six additional RST Equations are necessary to solve the Navier-Stokes Equations. Several terms of the RST Equations must be modeled.

2.4 Boundary Layer [5] [4]

An important attribute of a fluid is its viscosity. All particles in a fluid are bonded to each other and are limited in their movement. This effect is also called internal friction. The flow velocity at walls is zero ($U|_{i,wall} = 0$), because fluids adhere to solid walls. This boundary condition at static walls and the viscosity of fluids cause a certain flow profile (Figure 2.4). The area in which the velocity changes from zero to the free-stream velocity W_∞ or U_∞ is called **boundary layer**. Near the walls the Reynolds Number diminishes because of the low velocities and the viscosity forces are dominant. The boundary layer can be categorized in several layers (Figure 2.5). A dimensionless wall distance y^+ is used to enable general descriptions of the boundary layer. The definition of the shear stress velocity is necessary to calculate y^+ .

$$y^+ = \frac{u_\tau y}{\nu} \quad (2.41)$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad \tau_w = \mu \left(\frac{du}{dy} \right)_w \quad (2.42)$$

$$u^+ = \frac{u}{u_\tau} \quad (2.43)$$

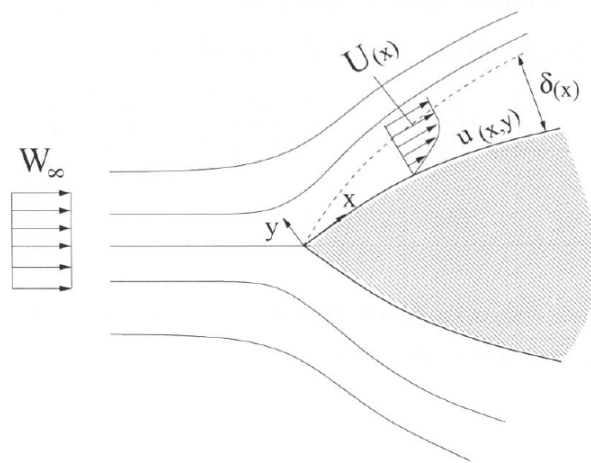


Figure 2.4: Boundary layer [5]

I Laminar inner layer: $y^+ < 5$

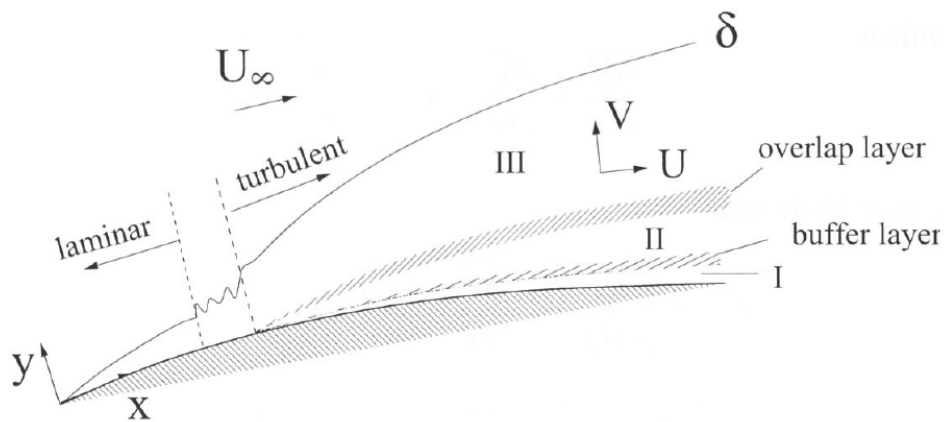


Figure 2.5: Layers of the boundary layer [5]

$$\tau_{turb} \ll \tau_{lam} \rightarrow \tau \approx \tau_{lam}$$

For the laminar inner layer, where the laminar shear stresses are dominant, following linear dependence can be formulated:

$$y^+ = u^+ \quad (2.44)$$

I – II Buffer layer: $5 < y^+ < 30$

$$\tau = \tau_{turb} + \tau_{lam}$$

In further distance from the wall, there is a buffer layer. In this layer the shear stresses are a sum of laminar and turbulent shear stresses.

II Turbulent inner layer: $y^+ > 30, \frac{y}{\delta} < 0.2$

$$\tau_{turb} \gg \tau_{lam} \rightarrow \tau \approx \tau_{turb}$$

Even further away from the wall, in the inner turbulent layer, the log-law can be formulated:

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B$$

In this equation, the Karman-Constant $\kappa \approx 0.41$ and the empiric constant $B \approx 5.0$.

II – III Overlap layer:

$$\tau_{turb} \gg \tau_{lam} \rightarrow \tau \approx \tau_{turb} \quad (2.45)$$

III Turbulent outer layer: $y^+ > 30, \frac{y}{\delta} > 0.2$

$$\tau_{turb} \gg \tau_{lam} \rightarrow \tau \approx \tau_{turb} \quad (2.46)$$

A fluid flowing inside a pipe or along a wall causes a boundary layer. The boundary layer is the area near the wall in which the friction has an influence on the flow profile normal to the wall. The boundary layer grows along the direction of flow and grows together after a certain length. From this location, the flow profile in flow direction can't change anymore. This flow is named a nondisturbed flow. The length from the entrance of the flow to where the boundary layers grow together is called the lead-in area. This length can be calculated for laminar and turbulent flows:

$$\text{Laminar flow: } \frac{x_E}{D_{hydr}} \approx 0.05 Re_D \quad \text{with: } Re_D = \frac{u_m D_{hydr}}{\nu} \quad (2.47)$$

$$\text{Turbulent flow: } 10 \leq \frac{x_E}{D_{hydr}} \leq 60 \quad (2.48)$$

2.5 Pressure Loss

Pressure loss is caused by wall friction in pipes, fittings, bows, etc. Usually a coefficient of drag ξ is used to calculate the pressure loss:

$$\Delta p_V = \xi \frac{\rho}{2} v^2 \quad (2.49)$$

For many geometries like a bow or a diffuser, it's possible to read the coefficient of drag from **Idelchik Diagrams** [9]. These values for ξ have been ascertained by I.E. Idelchik in experiments.

2.5.1 Static Pressure vs. Total Pressure

The total pressure is the sum of dynamic pressure and static pressure (Equation 2.50) and can be seen in the **Bernoulli Equation** (Equation 2.51).

$$p_{tot} = p_{stat} + p_{dyn} \quad (2.50)$$

$$\underbrace{\rho g z_1}_{\text{geodetic pressure}} + \underbrace{p_1}_{\text{static pressure}} + \underbrace{\frac{\rho}{2} u_1^2}_{\text{dynamic pressure}} = \rho g z_2 + p_2 + \frac{\rho}{2} u_2^2 \quad (2.51)$$

The total pressure is a measure for the energy content and is constant along a streamline of an ideal fluid without any losses. It's important to distinguish between static

and dynamic pressure. A difference of two total pressures is always an energy loss, while a difference between two static pressures could be a transformation of energy (static pressure can be transformed into dynamic pressure). The extended Bernoulli Equation 2.52 additionally contains losses.

$$\rho g z_1 + p_1 + \frac{\rho}{2} u_1^2 = \rho g z_2 + p_2 + \frac{\rho}{2} u_2^2 + \Delta p_V \quad (2.52)$$

2.5.2 Pressure Loss of a Pipe Flow [4]

In the simple case of a straight pipe ξ can be calculated:

$$\xi = \lambda \frac{l}{d} \quad (2.53)$$

d is the hydraulic diameter and can be calculated with:

$$d = 4 \frac{A}{U} \quad (2.54)$$

λ is the pipe drag coefficient and can be obtained by two different ways. The first one is to calculate λ . If the flow is laminar, the pipe drag coefficient of a circular pipe is:

$$\lambda = \frac{64}{Re} \quad (2.55)$$

λ of a hydraulic smooth pipe also can be determined for $Re < 10^5$ with Equation 2.56, discovered by Blasius [4]:

$$\lambda = 0.3164 * Re^{-\frac{1}{4}} \quad (2.56)$$

The second way is to read λ from diagrams. In the case of a turbulent flow:

$$\lambda = f \left(Re, \frac{e}{D} \right) \quad (2.57)$$

The value of the function for λ can be read from the **Moody Diagram** (Figure 2.6). Most tunnel and ventilation surfaces are smooth. The ratio of e/D is near 0 and the line for hydraulic smooth pipes can be used.

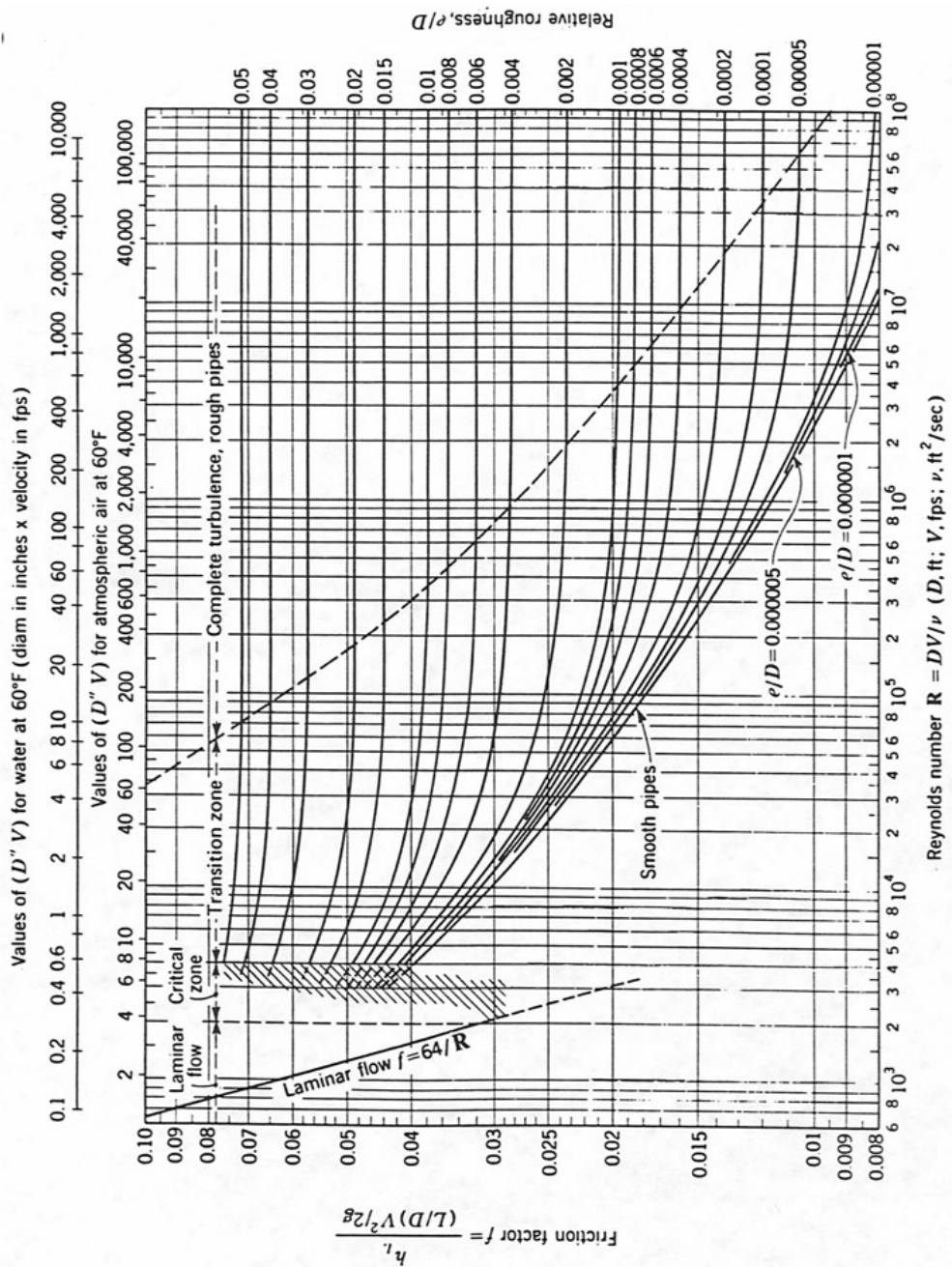


Figure 2.6: Moody Diagram [8]

2.5.3 Pressure Loss for a Sudden Change in Velocity [9]

An abrupt enlargement of a tube's cross-sectional area gives rise to so-called shock losses. In the case of uniform velocity distribution over the cross section of the smaller upstream channel in a turbulent flow, the local resistance coefficient of the "shock" depends only on the cross-sectional area ratio $\frac{F_0}{F_2}$ and is calculated from the Borda-Carnot formula as:

$$\xi_{loc} = \left(1 - \frac{F_0}{F_2}\right)^2 \quad (2.58)$$

The total resistance coefficient of the section with an abrupt expansion is:

$$\xi = \xi_{loc} + \xi_{fr} \quad (2.59)$$

Where ξ_{fr} can be read from an Idelchik Diagram [9]. The term ξ_{fr} must not be incorporated, if the pressure loss in the straight pipe downstream of the change in velocity is considered separately.

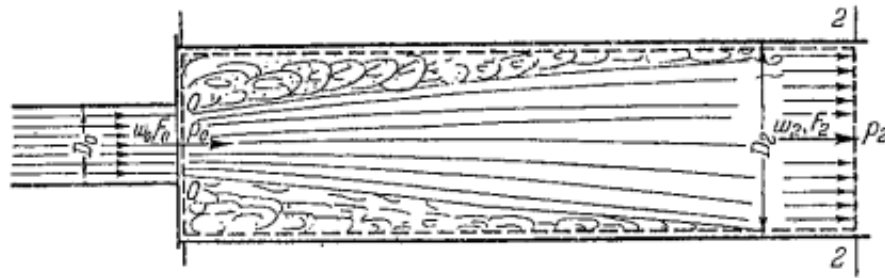


Figure 2.7: Eddy-areas for a sudden change in velocity [9]

Derivation of ξ_{loc} : The extended Bernoulli Equation describes the relation between point 0 and point 2 (Figure 2.8). The geodetic pressure is canceled because both points are at the same geodetic height.

$$\rho g z_0 + p_0 + \frac{\rho}{2} u_0^2 = \rho g z_2 + p_2 + \frac{\rho}{2} u_2^2 + \Delta p_V \quad (2.60)$$

u_0 and u_2 are connected by the Equation of Continuity:

$$\rho F_0 u_0 = \rho F_2 u_2 \quad (2.61)$$

$$u_2 = \frac{F_0}{F_2} u_0 \quad (2.62)$$

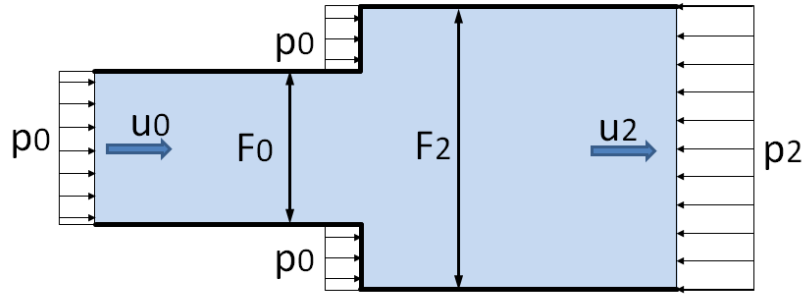


Figure 2.8: Sketch of a sudden cross-sectional change

The relation between p_0 and p_2 is given by the conservation of momentum:

$$\sum F = M_{out} - M_{in} \quad (2.63)$$

$$p_0 * F_0 + p_0 * (F_2 - F_0) - p_2 * F_2 = \underbrace{\rho F_2 u_2}_{\dot{m}} u_2 - \underbrace{\rho F_0 u_0}_{\dot{m}} u_0 \quad (2.64)$$

$$p_0 - p_2 = \rho u_2 (u_2 - u_0) \quad (2.65)$$

Inserting Equation 2.62 and 2.65 in Equation 2.60 leads to:

$$p_0 + \frac{\rho}{2} u_0^2 = \Delta p_V + p_2 + \frac{\rho}{2} u_2^2 \quad (2.66)$$

$$p_0 - p_2 + \frac{\rho}{2} (u_0^2 - u_2^2) = \Delta p_V \quad (2.67)$$

$$\rho u_2 (u_2 - u_0) + \frac{\rho}{2} (u_0^2 - u_2^2) = \Delta p_V \quad (2.68)$$

$$\rho \frac{F_0}{F_2} u_0 \left(\frac{F_0}{F_2} u_0 - u_0 \right) + \frac{\rho}{2} \left(u_0^2 - \left(\frac{F_0}{F_2} u_0 \right)^2 \right) = \Delta p_V \quad (2.69)$$

$$\left[\left(\frac{F_0}{F_2} \right)^2 - 2 \frac{F_0}{F_2} + 1 \right] \frac{\rho}{2} u_0^2 = \Delta p_V \quad (2.70)$$

$$\underbrace{\left(1 - \frac{F_0}{F_2} \right)^2}_{\xi_{loc}} \frac{\rho}{2} u_0^2 = \Delta p_V \quad (2.71)$$

2.5.4 Pressure Loss for a Change of Flow Direction [9]

Turning the flow in curved pipes or channels, results in the appearance of centrifugal forces. The pressure increases on the outer wall of the bow and decreases on the inner wall. Therefore, the flow will be slower on the outer wall and faster on the inner wall.

Thus, in this bend a diffuser effect occurs near the outer wall and a bell mouth effect occurs near the inner wall. The passage of flow from the curve into the straight section is accompanied by these effects in reverse order. Eddy zones arise, which reduce the cross-sectional area (Figure 2.9). This reduced cross-sectional area causes a pressure loss. The pressure loss can be calculated with Equation 2.49, where ξ can be calculated with Equation 2.72.

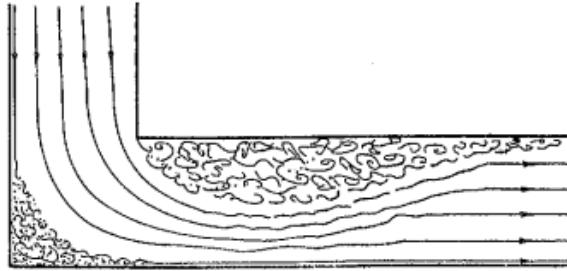


Figure 2.9: Eddy-areas for a change of flow direction [9]

$$\xi = C_1 A \xi_M \quad \text{if} \quad b_0 = b_1 \quad (2.72)$$

The factor C_1 depends from the geometry of the cross section, while A and ξ_M are a function of the bow-angle (Figure 2.10).

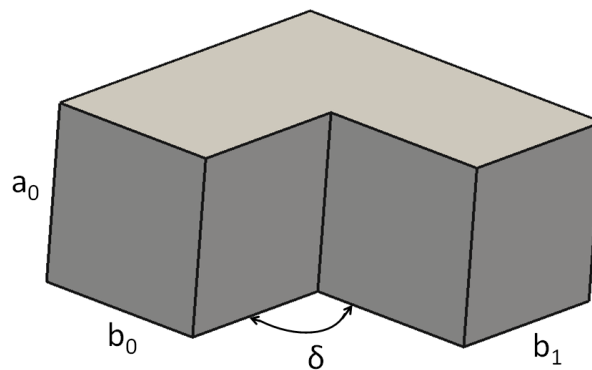


Figure 2.10: Change of flow direction

$$C_1 = f\left(\frac{a_0}{b_0}\right) \quad (2.73)$$

$$A = f(\delta) \quad (2.74)$$

$$\xi_M = f(\delta) \quad (2.75)$$

2.6 Heat Transfer

Every substance is built from small particles (atoms and molecules). The temperature of a medium is associated with the kinetic energy of these particles. This energy is a combination of translational movement, rotation and vibration of the particles. High temperature means high energy and low temperature means low energy of the particles. [4]

Heat can be transferred by:

- Conduction
- Convection
- Radiation

2.6.1 Heat Conduction [6]

Heat conduction is a molecular heat transport caused by temperature differences and takes place in solid bodies, liquids and gases. These temperature differences cause a difference of the particles' kinetic energy. Proper motion causes collisions which then causes energy transfer. Statistically the energy flux always flows in the direction of low energy.

The ability to transfer energy by molecular motion is called heat conductivity λ and differs from material to material. The transferred energy depends on the temperature gradient and the heat conductivity. This relation has already been used in Equation 2.11.

2.6.2 Heat Convection [5]

The transport of a physical quantity caused by a flow is called **convection**. In the field of heat transfer it's the transport of heat. The mathematical function for the heat flow density is:

$$q = \alpha (T_W - T_\infty) \quad (2.76)$$

The problem with Equation 2.76 is that the heat transfer coefficient α is unknown, but this equation allows expressing α as a function of the heat flow density and the temperature difference:

$$\alpha = \frac{q}{T_W - T_\infty} \quad (2.77)$$

To calculate α it's necessary to look at the region near the wall ($y \rightarrow 0$). Near the wall the velocity is low, or rather $\vec{v} = 0$ when $y = 0$ (at the wall). If the velocity becomes 0, the heat is only transferred through heat conduction. So it's possible to

calculate the heat flux density in Equation 2.77 with Equation 2.11 for **Fourier's Law of Conduction**.

$$\alpha = -\lambda \left. \frac{\partial T}{\partial y} \right|_{y=0} \frac{1}{T_W - T_\infty} \quad (2.78)$$

Equation 2.78 is a boundary condition. Transforming this equation into a dimensionless form by using Equation 2.20 and 2.79:

$$\Theta^* = \frac{T - T_W}{T_w - T_\infty} \quad (2.79)$$

$$\frac{\alpha L}{\lambda} = - \left. \frac{\partial \Theta^*}{\partial y^*} \right|_{y^*=0} \quad (2.80)$$

The left term of Equation 2.80 is known as the **Nusselt-Number**:

$$\text{Nusselt Number: } Nu = \frac{\alpha L}{\lambda} = \frac{\text{real heat flow density}}{\text{heat flow density with a stationary fluid}} \quad (2.81)$$

Nu and therefore α are functions of the location on the wall surface. According to Wilhelm Nusselt, it's possible to calculate the Nusselt Number with the following equation:

$$Nu_x = C Re_x^m Pr^n \quad (2.82)$$

In this case it's necessary to collect the constants C , m and n out of experiments.

2.6.3 Heat Radiation [5]

Electromagnetic waves are used to transport energy. Heat radiation only takes place in a small spectrum of the electromagnetic waves:

$$0.1 \leq \lambda \leq 10[\mu m]$$

Definition of a black body:

A black body is an idealized body, which emits the maximum possible amount of heat radiation at a certain temperature. The emission spectrum of a black body is described by the **Planck's Radiation Law**:

$$e_{s\lambda} = \frac{C_1}{\lambda^5 \left[\exp\left(\frac{C_2}{\lambda T}\right) - 1 \right]} \quad (2.83)$$

where $C_1 = 0.741 * 10^{-16} [Wm^2]$ and $C_2 = 1.438 * 10^{-2} [mK]$.

The maximum of the spectral energy $e_\lambda(T)$ moves to a lower wavelength, if the surface temperature of the emitting surface rises. This phenomenon is described by **Wien's Displacement Law**:

$$\lambda_m T = 0.29 * 10^{-2} [mK] \quad (2.84)$$

The total emitted emission of a black body at the temperature T can be calculated with Equation 2.85:

$$e_s(T) = \int_0^{\infty} e_{s,\lambda}(T) d\lambda \quad (2.85)$$

The total emission of a black body at a certain temperature is described by **Stefan-Boltzmann's Law**:

$$e_s(T) = \int_0^{\infty} \frac{C_1}{\lambda^5 [\exp(\frac{C_2}{\lambda T}) - 1]} d\lambda = C_S \left(\frac{T}{100} \right)^4 \quad (2.86)$$

with the Stefan-Boltzmann's Constant

$$C_S = 5.67 \left[\frac{W}{m^2 K^4} \right]$$

The emissivity of a gray body is defined with:

$$\varepsilon_{\lambda}(T) = \frac{e_{\lambda}(T)}{e_{s,\lambda}(T)} \neq f(\lambda) \quad (2.87)$$

$$\varepsilon_{\lambda}(T) = \varepsilon(T) \quad (2.88)$$

Stefan-Boltzmann's Law for gray bodies says:

$$e(T) = \varepsilon(T) C_S \left(\frac{T}{100} \right)^4 \quad (2.89)$$

3 Basics of Computational Fluid Dynamics

Many phenomena in engineering, like flow processes and heat transfer processes, are described by systems of differential equations. In most cases an analytical solution of these systems is not possible. The appearance of computers and the great enhancement of the computing performance in the last years enabled the opportunity to solve these problems numerically. **CFD** (Computational Fluid Dynamics) deals with numerical solutions of the basic equations of fluid dynamics (Chapter 2).

3.1 Discretization of the Examined Domain [10]

Continuous problems (systems of differential or integral equations) must be approximated by a suitable discrete problem. That means that all physical quantities, which must be calculated, are approximated by a finite amount of values. This process is called discretization and includes two tasks:

- Discretization of the examined domain
- Discretization of the equations

The discretization of the examined domain approximates the continuous domain (in time and space) by a finite amount of sub-domains, in which the numerical values of the unknown variables are determined. On one hand, the geometry should be modeled as exact as possible, but on the other hand a “good” grid should be used to enable an efficient calculation. A basic statement about the discretization of the examined area is: On one hand, a more regular grid leads to a fast calculation, but on the other hand a more regular grid includes disadvantages in terms of modeling complex geometries. Besides the shape of the cells, numerical grids can be distinguished in three types:

- **Boundary Approximated Grids:** The boundary of the examined domain is approximated by grid lines (Figure 3.1a).
- **Cartesian Grids:** The whole domain is covered by a regular grid, but irregular cells occur on the boundaries. These irregular cells require a special treatment (Figure 3.1b).
- **Overlapping Grids:** Different domains are discretized independently. Overlapping zones, which require a special treatment, are allowed at interfaces (Figure 3.1c)

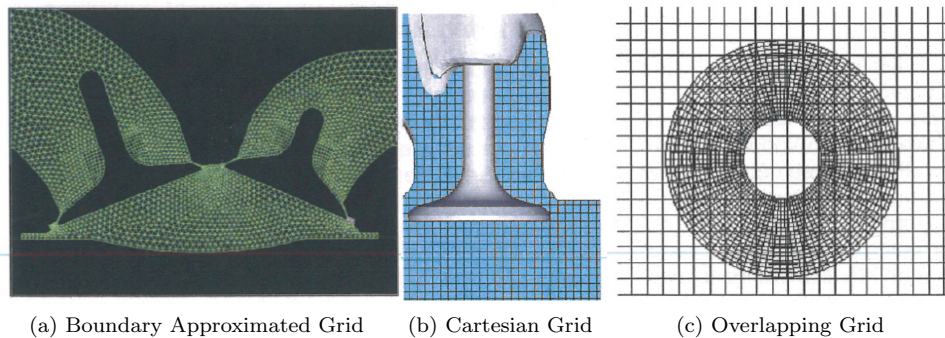


Figure 3.1: Grid types [10]

Another important distinguishing characteristic of numerical grids is the logical arrangement of the cells. Two different classes can be distinguished:

- Structured Grids (Figure 3.2a)
- Unstructured Grids (Figure 3.2b).

Structured Grids are characterized by a regular arrangement of the cells. Used cells may be curvilinear. There are directions, in which the amount of grid points is constant. Because of the regular arrangement, the relationships between neighboring cells always follow a fixed pattern. Only the coordinates of the grid points must be memorized, while the identity of the neighboring cells is clearly defined by the grid structure.

In **Unstructured Grids** there is no regularity in the arrangement of the grid points. The possibility to distribute the cells irregularly enables a high flexibility for modeling complex geometries because the cells can be placed very accurately of the boundaries. Besides the coordinates of the grid points, the relationship between neighboring cells must be memorized.

In the course of this diploma thesis only boundary approximated, structured, and unstructured grids are used.

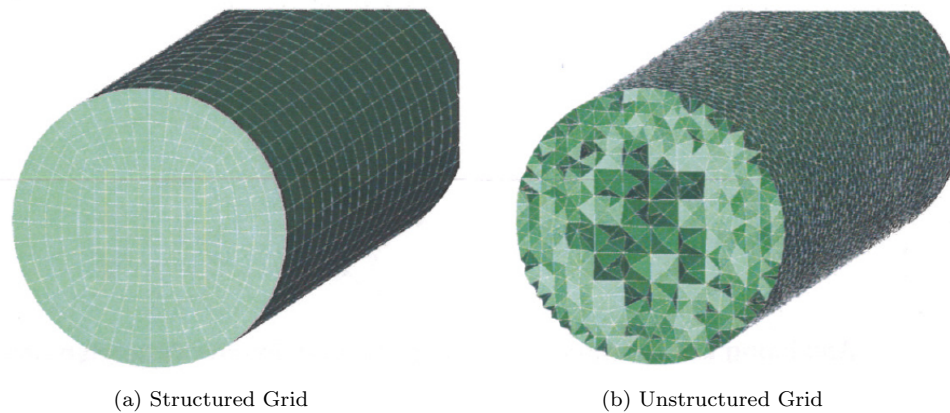


Figure 3.2: Types of grid structures [10]

3.2 Spatial and Temporal Discretization [10] [11]

The discretization of the examined domain approximates the continuous domain (in time and space) by a finite amount of sub-domains, in which the numerical values of the unknown variables are determined. Necessary relationships for the calculation of these numerical values can be obtained from the discretization of the equations. There are basically three approaches for discretization methods:

- Finite Differences Method (FDM)
- Finite Volume Method (FVM)
- Finite Elements Method (FEM)

Finite Elements Method : [12]

The examined domain is subdivided into a finite amount of sub-domains, the finite elements. Trial functions are defined for each element, which are zero for most of the other elements. A linear combination of these trial functions is used to solve the differential equations approximately. These linear combinations can be written into an element matrix for each element. The sum of all element matrices builds the cumulative matrix, which is a quadratic matrix and most of its entries are equal to zero. If the same trial functions are used for all elements, the cumulative matrix is additionally symmetric. The basic equation behind the finite elements method is:

$$K\Phi = f \quad (3.1)$$

In Equation 3.1 the variables stand for:

- K for the cumulative matrix

- Φ for the vector of unknown function values
- f for the vector of boundary conditions

Finite Volume Method :

The first step of a finite volume discretization is the dissection of the examined domain in a finite amount of small volumes (control volumes). All volumes together cover the whole examined domain.

The second step is to determine where the unknown variables should be calculated. Usually the center or the edges of the cells are chosen.

Equations of Conservation are formulated for each volume in integral form. The most important part of the discretization is to approximate the integrals of the Equations of Conservation by a suitable mean value for each integrand on each face of the control volume and build a relationship between the values at the faces and the unknown values in the center of the volume.

Finite Differences Method:

With the finite differences method, differential expressions are replaced by a difference formulation. The resulting equations are called difference equations.

$$\left. \frac{\partial f}{\partial x} \right|_x = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (3.2)$$

The resulting error depends on Δx and therefore is defined by the fineness of the grid. The difference equations can be derived with the help of a **Taylor Series Development**:

$$f(x + h) = f(x) + \frac{h}{1!} \left. \frac{\partial f}{\partial x} \right|_x + \frac{h^2}{2!} \left. \frac{\partial^2 f}{\partial x^2} \right|_x + \frac{h^3}{3!} \left. \frac{\partial^3 f}{\partial x^3} \right|_x + \dots \quad (3.3)$$

The variable Δx in Equation 3.2 is equal to the variable h in Equation 3.3, $h > 0$ is called a forward development and $h < 0$ is called a backward development. For the approximation of the first derivative, the row in Equation 3.3 is aborted after the second term. The result is a finite difference formulation for the first derivative with an abort error of first order $O(\Delta x)$:

$$\left. \frac{\partial f}{\partial x} \right|_x = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x) \quad (3.4)$$

The difference approximation for a first derivative of a function Φ at the point x_i is:

Forward difference quotient: $\left. \frac{\partial f}{\partial x} \right|_{x_i} \approx \frac{\Phi_{i+1} - \Phi_i}{\Delta x} + O(\Delta x) \quad (3.5)$

Backward difference quotient: $\left. \frac{\partial f}{\partial x} \right|_{x_i} \approx \frac{\Phi_i - \Phi_{i-1}}{\Delta x} + O(\Delta x) \quad (3.6)$

Central difference quotient: $\left. \frac{\partial f}{\partial x} \right|_{x_i} \approx \frac{\Phi_{i+1} - \Phi_{i-1}}{2\Delta x} + O(\Delta x) \quad (3.7)$

3.2.1 Stability of a Numerical Scheme [13]

Until now all treated functions in this chapter were variable in space. But in reality nearly every process is unsteady and therefore variable in space and time. The distinguishing characteristic of a time dependent scheme is the explicit or implicit solution of the equations of conservation. **Explicit Schemes** calculate the flow conditions at the new time step (n+1) with the values of the time step n. This is done separately for each node of the mesh. The advantage of these schemes is the simple application and implementation of boundary conditions. But the stability limits are low, which causes a long computing time for very fine grids (which are used for friction influenced flows).

$$\Phi_i^{n+1} = F(\Phi_i^n, \Phi_i^{n-1}, \dots) \quad (3.8)$$

Implicit Schemes are calculating the flow conditions in all nodes simultaneously and therefore they are unlimitedly stable. The higher numerical efforts are compensated by faster convergence of implicit schemes.

$$\Phi_i^{n+1} = F(\Phi_i^{n+1}, \Phi_i^n, \dots) \quad (3.9)$$

The Von-Neumann-Stability-Analysis (or L2-Analysis) is a procedure to investigate the stability of a numerical scheme for the solution of time dependent differential equations. According to Von Neumann a numerical scheme is L2-stable, if the numerical error is limited for given step sizes Δx and Δt . The first step of the Von-Neumann-Stability-Analysis is to assume the solution periodic. The periodic error E_i^n can be written as a Fourier Series:

$$E_i^n = \sum_{-N}^N c_j^n e^{ij\Delta x I} \quad (3.10)$$

A numerical scheme is stable if:

$$G(\Phi) = \frac{c_j^{n+1}}{c_j^n} \quad (3.11)$$

$$|G(\Phi)| \leq 1 \quad (3.12)$$

$$\text{for: } \Phi = j\Delta x \quad (3.13)$$

3.2.2 Explicit Euler Scheme [10]

The simplest example of an explicit time discretization scheme is the Explicit Euler Scheme. The unsteady term of an equation can be approximated at the time step t_n by a forward discretization:

$$\frac{\partial \Phi_i(t_n)}{\partial t} \approx \frac{\Phi_i^{n+1} - \Phi_i^n}{\Delta t_n} = L(\Phi) \quad (3.14)$$

Equation 3.14 can be dissolved for Φ_i^{n+1} :

$$\Phi_i^{n+1} = \Phi_i^n + \Delta t_n * L(\Phi) \quad (3.15)$$

3.2.3 Stability of a Unilateral Explicit Euler Scheme [10] [11]

An easy way to check the stability of a scheme is to perform a Von-Neuman-Stability-Analysis for the first two terms in Equation 2.16. The velocity u_j is set to a value c , which can be interpreted as a spreading rate of a disturbance in the flow, and the index i defines the location. Equation 3.16 is called a **Wave Equation**.

$$\frac{\partial u_i}{\partial t} + c \frac{\partial u_i}{\partial x} = 0 \quad (3.16)$$

For a backward spatial discretization:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \quad (3.17)$$

Equation 3.17 defines the spreading of u and the expansion of each term in the Fourier Series. For this reason Equation 3.18 can be written for the j -th summand of the Fourier Series (note that all terms are divided by $e^{ij\Delta x I}$).

$$c_j^{n+1} - c_j^n + c \frac{\Delta t}{\Delta x} c_j^n (e^0 - e^{-I\Phi}) = 0 \quad (3.18)$$

The last step is to check the limitation of the intensification factor:

$$G(\Phi) = \frac{c_j^{n+1}}{c_j^n}$$

$$G(\Phi) = 1 - c \frac{\Delta t}{\Delta x} + c \frac{\Delta t}{\Delta x} \cos(\Phi) - I * c \frac{\Delta t}{\Delta x} \sin(\Phi) \leq 1$$

$$\left(1 - c \frac{\Delta t}{\Delta x} (1 + \cos(\Phi))\right)^2 - \left(c \frac{\Delta t}{\Delta x} \sin(\Phi)\right)^2 \leq 1$$

or: $(1 - \nu + \nu \cos(\phi))^2 + (\nu \sin(\Phi))^2 \leq 1$

Where ν is the **Courant Number**:

$$\nu = CFL = c \frac{\Delta t}{\Delta x} \quad (3.19)$$

This leads to following stability requirement:

$$0 \leq c \frac{\Delta t}{\Delta x} \leq 1 \quad (3.20)$$

Equation 3.20 shows that the algebraic sign of the Courant Number depends on the velocity c . A backward discretization is only stable for positive c -values (as shown above), while a forward discretization is stable for negative c -values. The reason for the different algebraic signs is the second node of the local discretization. If the second node is located upstream of the first node (as it is in the case of a backward discretization), the direction of the information transport is equal to the direction of the physical transport. Some schemes, like the **Upwind Scheme**, consider the direction of the information transport.

3.2.4 Implicit Euler Scheme [10]

If the time derivation is approximated at the time step t_{n+1} with a backward discretization of first order, the following equation can be written:

$$\frac{\partial \Phi(t_{n+1})}{\partial t} \approx \frac{\Phi_i^{n+1} - \Phi_i^n}{\Delta t_n} = L(\Phi^{n+1}) \quad (3.21)$$

Equation 3.21 cannot be explicitly dissolved for Φ_i^{n+1} . The difference between the explicit and implicit Euler Scheme is the time level. An implicit scheme solves the right side of Equation 3.21 for a new (unknown) time level. A linear equation system is necessary for the calculation of the new time level.

3.2.5 Stability of an Unilateral Implicit Euler Scheme

The Von-Neumann-Stability-Analysis is performed a second time for Equation 3.16. However this time it is done for an implicit Euler Scheme:

$$\frac{\partial u_i}{\partial t} + c \frac{\partial u_i}{\partial x} = 0 \quad (3.22)$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^{n+1} - u_{i-1}^{n+1}}{\Delta x} = 0 \quad (3.23)$$

Again the j-th summand of the periodic error can be written:

$$c_j^{n+1} - c_j^n + c \frac{\Delta t}{\Delta x} c_j^{n+1} (e^0 - e^{-I\Phi}) = 0 \quad (3.24)$$

The next step is to check the limitation of the intensification factor:

$$\begin{aligned} G(\Phi) &= \frac{c_j^{n+1}}{c_j^n} \\ G(\Phi) &= \frac{1}{1 + v(1 - \cos(\Phi)) + I \sin(\Phi)} = \frac{(1 + v - v \cos(\Phi)) - Iv \sin(\Phi)}{(1 + v - v \cos(\Phi))^2 + (v \sin(\Phi))^2} \\ G(\Phi) &= \frac{(1 + v - v \cos(\Phi)) - Iv \sin(\Phi)}{D} \\ |G(\Phi)| &\leq 1 \\ \frac{D}{D^2} &\leq 1 \\ D &\leq D^2 \end{aligned} \quad (3.25)$$

Equation 3.25 is fulfilled, if $D \geq 1$:

$$\begin{aligned} (1 + v - v \cos(\Phi))^2 + (v \sin(\Phi))^2 &\geq 1 \\ 1 + 2v + 2v^2 + 2v \cos(\Phi) + 2v^2 \cos(\Phi) &\geq 1 \end{aligned}$$

$$1 + 2v + 2v^2 \geq 1 - 2v \cos(\Phi) - 2v^2 \cos(\Phi) \quad (3.26)$$

To prove the observance of Equation 3.26, each term must be checked separately (coefficient comparison):

$$\text{for } v^0 : \quad 1 \geq 1 \quad (3.27)$$

$$\text{for } v^1 : \quad 2 \geq -2 \cos(\Phi) \quad (3.28)$$

$$\text{for } v^2 : \quad 2 \geq -2 \cos(\Phi) \quad (3.29)$$

$$(3.30)$$

The Equations 3.27 to 3.29 are always fulfilled (independent from CFL!), because $\cos(\Phi)$ is in the range of -1 to 1. For this reason implicit schemes are always stable and a larger time step Δt than for explicit schemes is permissible. Because of that, the effort for solving a linear equation system is compensated and an implicit scheme is more efficient in most cases.

3.2.6 Combination of Explicit and Implicit Schemes [10] [16]

Predictor-Corrector-Scheme

The basic idea behind a predictor-corrector-scheme is the combination of explicit and implicit equations. In the first step, a good initial value $\tilde{\Phi}^{n+1}$ is estimated with an explicit scheme (the predictor scheme). This could be:

$$\tilde{\Phi}_i^{n+1} = \Phi_i^n + \Delta t_n * L(\Phi) \quad (3.31)$$

In a second step this initial value is inserted in an implicit scheme (the corrector scheme) to calculate a new value Φ^{n+1} :

$$\Phi_i^{n+1} = \Phi_i^n + \Delta t_n * L(\tilde{\Phi}^{n+1}) \quad (3.32)$$

The advantage of such a scheme is the possibility to calculate all values explicitly. The solution of a linear equation system can be avoided.

Linear Combination of Explicit and Implicit Schemes

In many numeric programs an explicit or implicit scheme can simply be chosen by a parameter in the discretization equation:

$$\frac{\Phi_i^{n+1} - \Phi_i^n}{\Delta t} \approx \beta L(\Phi_i^{n+1}) + (1 - \beta)L(\Phi_i^n) \quad (3.33)$$

$$\text{Scheme} = \begin{cases} \text{completely implicit,} & \text{if } \beta = 1 \\ \text{completely explicit,} & \text{if } \beta = 0 \end{cases}$$

A special form of a linear combination is the Crank-Nicolson Scheme. It is an implicit (but not completely implicit) scheme, which is always stable, but does not always

produce physically plausible results for large time steps. However the Crank-Nicolson Scheme is more accurate for small time steps, which can be seen in Figure 3.3. It shows an explicit scheme tends to overestimate, while an implicit scheme tends to underestimate. The Crank-Nicolson scheme is a compromise between both schemes.[16] [17]

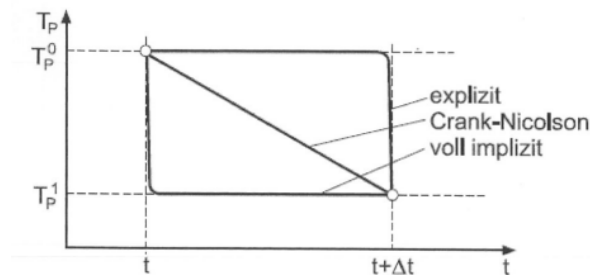


Figure 3.3: Trends of variables over the time for different schemes [16]

3.3 Upwind Schemes [11]

The family of upwind schemes considers the physical characteristics of the flow equations in a discrete formulation. Part of the upwind schemes, among others, are **Flux Vector Splitting** and **Riemann Schemes**.

3.3.1 Flux Vector Splitting

The stability of a unilateral discretization of the one-dimensional wave equation (Equation 3.16) depends on the wave velocity c :

$$u_i^{n+1} = \begin{cases} u_i^n - c \frac{\Delta t}{\Delta x} (u_i^n - u_{i-1}^n), & \text{if } c \geq 0 \\ u_i^n - c \frac{\Delta t}{\Delta x} (u_{i+1}^n - u_i^n), & \text{if } c \leq 0 \end{cases} \quad (3.34)$$

To consider the different algebraic signs of the spreading rate, both equations above can be combined:

$$c^+ = \max(c, 0) = \frac{1}{2}(c + |c|) \quad (3.35)$$

$$c^- = \max(-c, 0) = \frac{1}{2}(c - |c|) \quad (3.36)$$

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{\Delta x} [c^+(u_i^n - u_{i-1}^n)] + \frac{\Delta t}{\Delta x} [c^-(u_{i+1}^n - u_i^n)] \quad (3.37)$$

In Equation 3.37 c^+ is always positive (or zero) and c^- is always negative (or zero). This equation contains the condition that the information in point u_i only depends on

the neighboring points upstream and not from the neighboring points downstream.

The stability limit is now:

$$CFL = |c| \frac{\Delta t}{\Delta x} \leq 1 \quad (3.38)$$

If this method is used for the Navier-Stokes Equations, it is called **Flux Vector Splitting**.

3.3.2 Riemann Solver (Godunov Scheme)

Godunov treated the solution at a certain time step piecewise constant along a mesh cell. The flow at the following time step depends on the waves, which spread from the boundaries between the individual cells. The solution of this problem is equal to the solution of the **shock tube or Riemann Problem**.

The shock tube describes the spread of disturbances after the membrane between two unequal still fluids is suddenly removed. The exact solution is a sum of a shock wave, a contact discontinuity and an expansion wave. This process is imaged in Figure 3.4.

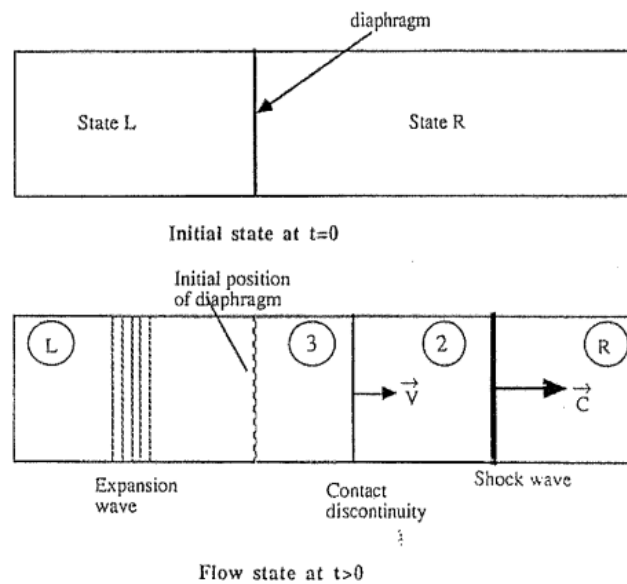


Figure 3.4: Schematic presentation of the shock tube flow [11]

After the calculation of the wave spreading, a new piecewise constant flow state at the time step t_{n+1} is gained. The result of the scheme is a conservative explicit algorithm with an accuracy of first order. The exact solution of the Riemann Problem requires

the solution of nonlinear algebraic equations. Because of the partly high effort for solving these equations, approximate Riemann Solvers have been developed, which are numerically more effective. The two most important approximate Riemann Solvers are the Osher Scheme and the Roe Scheme.

3.4 Techniques for Faster Convergence [11]

In Chapter 3.2.3 the CFL-Number has been mentioned. The CFL condition defines the maximum permissible time step, which barely leads to a convergent solution. Because of this relation, the smallest cell determines the time step and therefore the convergence velocity. If the temporal accuracy is not very important (e.g. in a steady flow), a faster convergence can be reached with a **local time step**. This local time step varies from cell to cell and enables the solution to progress with different velocities in each point of the mesh. Instead of a uniform time step, a uniform CFL-Number is used. The time for a global convergence still depends on cells with slowest convergence velocities.

Another technique for faster convergence is the **Multigrid Method**. The solution algorithm runs on levels with variable cell sizes. Coarser grids progress much faster in time, based on their bigger time step. Interpolation helps to transmit the solution of coarser grids to finer grids. This method causes a much faster convergence. Different strategies can be used for the Multigrid Method. One of the simplest forms is the V-cycle pictured in Figure 3.5. The calculation starts on a fine mesh and is restricted to coarser meshes. The solution converges on the coarsest grid and is interpolated back on the fine mesh.

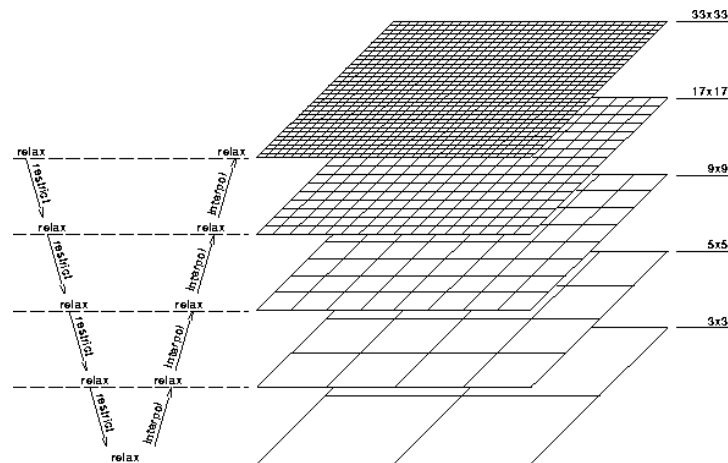


Figure 3.5: Schematic representation of the V-cycle [11]

3.5 Turbulence Models [14]

3.5.1 Eddy Viscosity Approach

The Bousinesq Approach has already been mentioned (Equation 2.39) and is the basic idea behind the following turbulence models. The aim of turbulence models is to determine the turbulent viscosity ν_T . The simplest models use an empirical turbulent mixing length. These models are called **Algebraic Models**:

$$\nu_T = l \left| \frac{\partial u}{\partial y} \right| \quad (3.39)$$

Another possibility to determine the turbulent viscosity is using **One-Equation Models**. These models consider transport processes and require the solution of an additional differential equation. Transport equations for the turbulent kinetic energy were used originally:

$$\frac{\partial k}{\partial t} = u_j \frac{\partial k}{\partial x_j} = R_{ij} \frac{\partial u_i}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (3.40)$$

In Equation 3.40 the variable ε is the dissipation of kinetic energy.

$$\varepsilon = \nu \left\langle \frac{\partial u'_i}{\partial x_k} \frac{\partial u'_i}{\partial x_k} \right\rangle \quad (3.41)$$

$$\varepsilon \sim \frac{k^{\frac{3}{2}}}{l} \quad (3.42)$$

And for One-Equation Models:

$$\varepsilon = C_D \frac{k^{\frac{3}{2}}}{l} \quad \text{where } C_D \text{ is a closure coefficient} \quad (3.43)$$

According to Prandtl:

$$\mu_T \sim l \sqrt{k} \quad (3.44)$$

And therefore:

$$\nu_T = l \sqrt{k} = C_D \frac{k^2}{\varepsilon} \quad (3.45)$$

Hence another algebraic equation for the turbulent mixing length must be solved. To solve the transport equation for the turbulent kinetic energy, additional information which considers local flow conditions is necessary. To get this additional information, a second transport equation can be used. The resulting turbulence models are called **Two- and More-Equation Models**. Two of them are used most widely, the k - ε -Model and the k - ω -Model. Since these two models are important in the context of this diploma thesis, they are explained in more detail.

3.5.2 k - ω -Model:

The first Two-Equation Model of turbulence was proposed by Kolomogorov. Kolomogorov's original reasoning has gone like this:

- Because k already appears in Equation 2.39, it is plausible that $\nu_T \sim k$.
- The dimensions of ν_T are (length)²/(time) while the dimensions of k are (length)²/(time)².
- Consequently ν_T/k has the dimension (time).
- Turbulence dissipation ε has dimensions (length)²/(time)³ (Equation 3.42).
- Equation 2.39 and 3.40 can be closed by introducing a variable with dimensions (time) or 1/(time).

This variable is ω , “the rate of dissipation of energy in unit volume and time”:

$$\omega \sim \frac{\sqrt{k}}{l} \quad (3.46)$$

The next step is to establish an equation for ω . Kologmorov postulated the following equation for ω :

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = -\beta \omega^2 + \frac{\partial}{\partial x_j} \left[\sigma \nu_T \frac{\partial \omega}{\partial x_j} \right] \quad (3.47)$$

Equation 3.47 is the transport equation for ω , where β and σ are closure coefficients. There is no analog to the k -equation's (Equation 3.40) turbulence production term. The absence of a production term is consistent with Kolomogorov's perception that ω is associated with the smallest scales of the turbulence. There is no molecular diffusion term so that this equation applies strictly to high-Reynolds-number flows. Therefore, it cannot be integrated through the viscous sublayer as it stands. The k - ω -model has been developed further over the past decades. The following version of the k - ω -model improves predictive accuracy for free shear flows and strongly separated flows (**Wilcox (2006) k - ω model**):

Kinematic Eddy Viscosity:

$$\nu_T = \frac{k}{\tilde{\omega}}, \quad \tilde{\omega} = \max \left[\omega, C_{\text{lim}} \sqrt{\frac{2S_{ij}S_{ij}}{\beta^*}} \right], \quad C_{\text{lim}} = \frac{7}{8} \quad (3.48)$$

Turbulence Kinetic Energy:

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = R_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[\left(\nu + \sigma^* \frac{k}{\omega} \right) \frac{\partial k}{\partial x_j} \right] \quad (3.49)$$

Specific Dissipation Rate:

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \alpha \frac{\omega}{k} R_{ij} \frac{\partial u_i}{\partial x_j} - \beta \omega^2 + \frac{\sigma_d}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \sigma \frac{k}{\omega} \right) \frac{\partial \omega}{\partial x_j} \right] \quad (3.50)$$

Closure Coefficients and Auxiliary Relations :

$$\alpha = \frac{13}{25}, \quad \beta = \beta_0 f_\beta, \quad \beta^* = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{3}{5}, \quad \sigma_{do} = \frac{1}{8} \quad (3.51)$$

$$\sigma_d = \begin{cases} 0, & \text{if } \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \leq 0 \\ \sigma_{do}, & \text{if } \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} > 0 \end{cases} \quad (3.52)$$

$$\beta_o = 0.0708, \quad f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega}, \quad \chi_\omega \equiv \left| \frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*\omega)^3} \right| \quad (3.53)$$

$$\text{Mean-rotation:} \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (3.54)$$

$$\text{Mean-strain-rate:} \quad S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.55)$$

$$\varepsilon = \beta^* \omega k \quad \text{and} \quad l = \frac{\sqrt{k}}{\omega} \quad (3.56)$$

3.5.3 Standard k - ε -Model:

The most popular two-equation model over many years was the k - ε model. In idea behind formulating the k - ε model is not only to derive the exact equation for ε , but also to find suitable closure approximations for the exact equation, which specifies its behavior. ε is defined by Equation 3.41.

Kinematic Eddy Viscosity:

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \quad (3.57)$$

Turbulence Kinetic Energy:

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = R_{ij} \frac{\partial u_i}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (3.58)$$

Dissipation Rate :

$$\frac{\partial \varepsilon}{\partial t} + u_j \frac{\partial \varepsilon}{\partial x_j} = C_{\varepsilon 1} \frac{\varepsilon}{k} R_{ij} \frac{\partial u_i}{\partial x_j} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \quad (3.59)$$

Closure Coefficients and Auxiliary Relations :

$$C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3 \quad (3.60)$$

$$\omega = \frac{\varepsilon}{C_\mu k} \quad \text{and} \quad l = C_\mu \frac{k^{\frac{3}{2}}}{\varepsilon} \quad (3.61)$$

3.5.4 New Baseline Model [15]

According to Menter [15], the k - ω model is the model of choice in the sublayer of the boundary layer. Unlike any other two-equation model, the k - ω model does not involve damping functions and allows simple Dirichlet boundary conditions to be specified. (A Dirichlet boundary condition specifies the values which the solution of a partial differential equation needs to take on the boundary of the domain.) One point of criticism is that the k - ω model does not correctly predict the asymptotic behavior of the turbulence at the wall. However, the mean flow profile and the wall friction are still predicted correctly [?]. The k - ω model is also used in the logarithmic part of the boundary layer. It has been shown [15] that the behavior of the k - ω model in the logarithmic region is superior to that of the k - ε model in equilibrium adverse pressure gradient flows and in compressible flows. In the wake region of the boundary layer, the k - ω model has to be abandoned in favor of the k - ε model. The reason for the switch between the two models is that the k - ω model has a very strong sensitivity to the free stream value ω_f specified for ω outside the boundary layer. In the free shear layers away from surfaces, the standard k - ε model will be utilized. There does not seem to be a model that accurately predicts all free shear flows and the k - ε model seems to be a fair compromise.

The idea behind the BSL model is to retain the robust and accurate formulation of Wilcox k - ω model in the near wall region, and to take advantage of the free stream independence of the k - ε model in the outer part of the boundary layers. To achieve this, the k - ε model is transformed into a k - ω formulation. The difference between this formulation and the original k - ω model is that an additional cross-diffusion term appears in the ω equation and the closure coefficients are different.

Original k - ω model :

$$\frac{\partial \rho k}{\partial t} + u_j \frac{\partial \rho k}{\partial x_j} = R_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{k1} \mu_T) \frac{\partial k}{\partial x_j} \right] \quad (3.62)$$

$$\frac{\partial \rho \omega}{\partial t} + u_j \frac{\partial \rho \omega}{\partial x_j} = \frac{\gamma_1}{\nu_T} R_{ij} \frac{\partial u_i}{\partial x_j} - \beta_1 \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{\omega 1} \mu_T) \frac{\partial \omega}{\partial x_j} \right] \quad (3.63)$$

Transformed k - ε model :

$$\frac{\partial \rho k}{\partial t} + u_j \frac{\partial \rho k}{\partial x_j} = R_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{k2} \mu_T) \frac{\partial k}{\partial x_j} \right] \quad (3.64)$$

$$\begin{aligned} \frac{\partial \rho \omega}{\partial t} + u_j \frac{\partial \rho \omega}{\partial x_j} &= \frac{\gamma_2}{\nu_T} R_{ij} \frac{\partial u_i}{\partial x_j} - \beta_2 \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{\omega 2} \mu_T) \frac{\partial \omega}{\partial x_j} \right] \\ &+ 2\rho \frac{\sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \end{aligned} \quad (3.65)$$

Now Equations 3.62 and 3.63 are multiplied by F_1 and Equations 3.64 and 3.65 are multiplied by $(1 - F_1)$. The corresponding equations of each set are added together to

give the new model:

$$\frac{\partial \rho k}{\partial t} + u_j \frac{\partial \rho k}{\partial x_j} = R_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_T) \frac{\partial k}{\partial x_j} \right] \quad (3.66)$$

$$\begin{aligned} \frac{\partial \rho \omega}{\partial t} + u_j \frac{\partial \rho \omega}{\partial x_j} &= \frac{\gamma}{\nu_T} R_{ij} \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] \\ &+ 2\rho(1 - F_1) \frac{\sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \end{aligned} \quad (3.67)$$

The blending function F_1 is 1 in the sublayer and logarithmic region of the boundary layer and gradually switches to zero in the wake region.

3.5.5 k - ω -SST Model [15]

The concept of the k - ω -SST model is the same as the BSL model's concept, but an additional modification of the eddy viscosity's definition is done. One of the major differences between eddy-viscosity and full Reynolds-Stress models is that the later takes the important effect of the transport of the principal turbulent shear stresses into account. The one-equation Johnson-King model includes a transport equation for the turbulent shear stress τ [15]:

$$\tau = \rho a_1 k \quad (3.68)$$

with a_1 being a constant. In Two-Equation models, the shear stress is computed from:

$$\tau = \mu_t \frac{\partial u}{\partial y} \quad (3.69)$$

According to [15], for conventional Two-Equation models, Equation 3.69 can be written to give:

$$\tau = \rho \sqrt{\frac{\text{Production}_k}{\text{Dissipation}_k}} a_1 k \quad (3.70)$$

In adverse pressure gradient flows Equation 3.70 leads to an overprediction of τ . Therefore the eddy viscosity is redefined in the following way to satisfy Equation 3.68 within the framework of a two-equation model:

$$\nu_T = \frac{a_1 k}{\max \left[a_1 \omega, \frac{\partial u}{\partial y} F_2 \right]} \quad (3.71)$$

where F_2 is a function that is 1 for boundary-layer flows and zero for free shear layers. In an adverse pressure gradient layer, production of k is larger than its dissipation ($\frac{\partial u}{\partial y} > a_1 \omega$) and the formulation $\nu_T = k/\omega$ is used. To get the original formulation of the eddy viscosity for free shear layers the modification to the shear-stress transport (SST) model is limited to wall bounded flows.

3.5.6 Large Eddy Simulation - LES [14]

A large eddy simulation is a computation in which large eddies are computed, and the smallest, sub-grid-scale (SGS) eddies, are modeled. The underlying assumption is that the largest eddies, which are directly affected by the boundary conditions, carry most energy of the Reynolds stresses and must be computed. The small-scale turbulence is weaker and is therefore less critical. Small-scale turbulence is also more isotropic and has nearly universal characteristics. Thus, it is more accessible to modeling. Because of modeling the smallest eddies, the smallest finite difference cells can be much larger and much larger time steps can be taken than in a DNS. Hence it is possible to achieve much higher Reynolds numbers with LES than with DNS with the same computing performance. A major difficulty in LES is that near a solid surface all eddies are small. If the stress behavior near solid walls is of interest, the grid spacing and time step required by LES fall towards that needed for full DNS as the surface is approached. To understand the difference between DNS and LES, the concept of **filtering** is important. Filtering can be explained using an explicit central-difference approximation for the first derivative of a variable $u(x)$. The distance between two neighbored points is named h . Values of flow properties at discrete points in a numerical simulation represent averaged values. The derivative can be written as follows.

$$\frac{u(x+h) - u(x-h)}{2h} = \frac{d}{dx} \left[\frac{1}{2h} \int_{x-h}^{x+h} u(\xi) d\xi \right] \quad (3.72)$$

The central-difference approximation can be thought as an operator that filters out scales smaller than the mesh size. Furthermore, the approximation yields the derivative of an averaged value of $u(x)$.

All physical quantities can be separated in two parts:

$$u_i = \bar{u}_i + u'_i$$

\bar{u}_i denotes the resolvable-scale filtered velocity. The sub-grid-scale (SGS) velocity is named u'_i . This is not the same allocation that has been done in Chapter 2.3. Because filtering is not equal to time averaging, all equations are time dependent and a steady solution of the problem is not possible. The filter width is defined as follows:

$$\Delta = \sqrt[3]{\Delta x \Delta y \Delta z} \quad (3.73)$$

Leonard (1974) defined a generalized filter as a convolution integral:

$$\bar{u}_i(\mathbf{x}, t) = \int \int \int G(\mathbf{x} - \xi; \Delta) u_i(\xi, t) d^3\xi \quad (3.74)$$

The filter function G is normalized by requiring that

$$\int \int \int G(\mathbf{x} - \xi; \Delta) u_i(\xi, t) d^3\xi = 1 \quad (3.75)$$

There are many filters that can be used. In all cases however, the filter introduces a scale Δ that represents the smallest turbulence scale allowed by the filter. The filter provides a formal definition of the averaging process and separates the resolvable scales from the sub-grid scales. Filtering differs from standard averaging in one important respect:

$$\overline{\overline{u_i}} \neq \overline{u_i} \quad (3.76)$$

Therefore, the convective flux in the momentum equation is given by:

$$\overline{u_i u_j} = \overline{u_i} * \overline{u_j} + L_{ij} + C_{ij} + R_{ij} \quad (3.77)$$

where

$$L_{ij} = \overline{\overline{u_i} * \overline{u_j}} - \overline{u_i} * \overline{u_j} \quad (3.78)$$

$$C_{ij} = \overline{u_i u_j'} + \overline{u_j u_i'} \quad (3.79)$$

$$R_{ij} = \overline{u_i' u_j'} \quad (3.80)$$

According to [14] the term L_{ij} , the Leonard-stress term, removes significant energy from the resolvable scale, but can be computed and need not be modeled. The term C_{ij} , the cross-term stress tensor, also drains significant energy from the resolvable scales. The sum of C_{ij} and R_{ij} must be modeled. For that reason Equation 2.37 is rearranged into a more conventional form:

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial \overline{u_j} * \overline{u_i}}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu \frac{\partial \overline{u_i}}{\partial x_j} + \tau_{ij} \right] \quad (3.81)$$

where

$$\tau_{ij} = - \left(Q_{ij} - \frac{1}{3} Q_{kk} \delta_{ij} \right) \quad (3.82)$$

$$P = \overline{p} + \frac{1}{3} \rho Q_{kk} \delta_{ij} \quad (3.83)$$

$$Q_{ij} = R_{ij} + C_{ij} \quad (3.84)$$

The fundamental problem of Large Eddy Simulation is to model the SGS stresses represented by the tensor Q_{ij} . One of the most popular models is the Smagorinsky Model. This model assumes the SGS stresses follow a gradient-diffusion process:

$$\tau_{ij} = 2\nu_T S_{ij}, \quad S_{ij} = \frac{1}{2} \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) \quad (3.85)$$

where S_{ij} is called the “resolved strain rate” and ν_T is the Smagorinsky eddy viscosity

$$\nu_T = (C_S \Delta)^2 \sqrt{S_{ij} S_{ji}} \quad (3.86)$$

and C_S is the Smagorinsky coefficient. Smagorinsky used a definition for ν_T , which is similar to Equation 3.39, with a mixing length $C_S \Delta$.

In terms of computational effort, LES is a compromise between RANS and DNS. Direct Numerical Solution does not need any modeling to solve the Navier-Stokes Equations. To enable a direct solution of the Navier-Stokes Equations, a high resolution in time and space is necessary as well as a high computing performance. In a Large Eddy Simulation all large eddies are computed and sub-grid-scaled eddies are modeled. The highest degree of modeling is necessary for Reynolds Averaged Navier-Stokes models, where all turbulent quantities must be modeled.

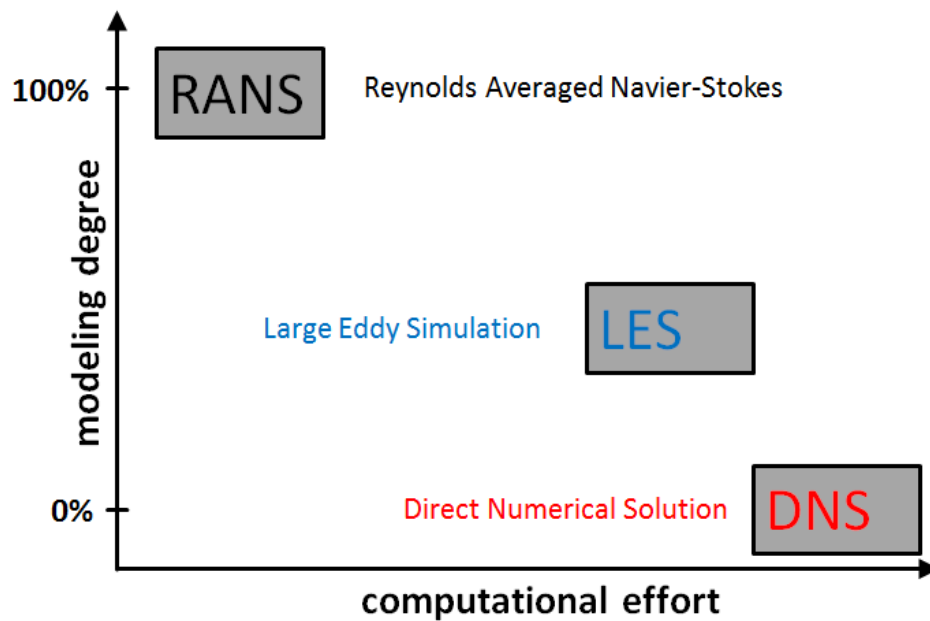


Figure 3.6: Modeling degree of different approaches

4 Description of Used Programs

4.1 NUMSTA3 - Numerical Simulation of Tunnel Aerodynamics 3

NUMSTA3 is a simulation package for tunnel aerodynamics and ventilation problems. It allows simulating various influences, such as:

- Fans
- Louvers
- Traveling vehicles
- Buoyancy effects
- Humid air
- etc.

It can be controlled with a graphical user interface or by editing input files. In its current state it is essentially a one-dimensional solver. [18] The basics of NUMSTA3 is a one-dimensional, conservative form of the unsteady, incompressible Euler Equations. Three-dimensional effects can be taken into account with empiric coefficients. A numeric scheme, based on the Roe Scheme [19], is used to discretize the system with a finite element approach. NUMSTA3 generates a three-dimensional tube model of the user defined tunnel. The tunnel is split up into segments depending on the defined resolution. All relevant physical quantities are calculated for each element during the simulation.

A calculation case in NUMSTA3 consists of elements, which are connected to each other. Properties (name, length, cross-section, resolution, heat transfer coefficient, etc.) must be defined explicitly for each element, while solver settings (gravity, maximal simulation time, maximal number of iterations, initial pressure, initial temperature, numerical schemes, CFL number, etc.) are general settings for all elements. An overview of a NUMSTA3 file is given in Figure 4.1.

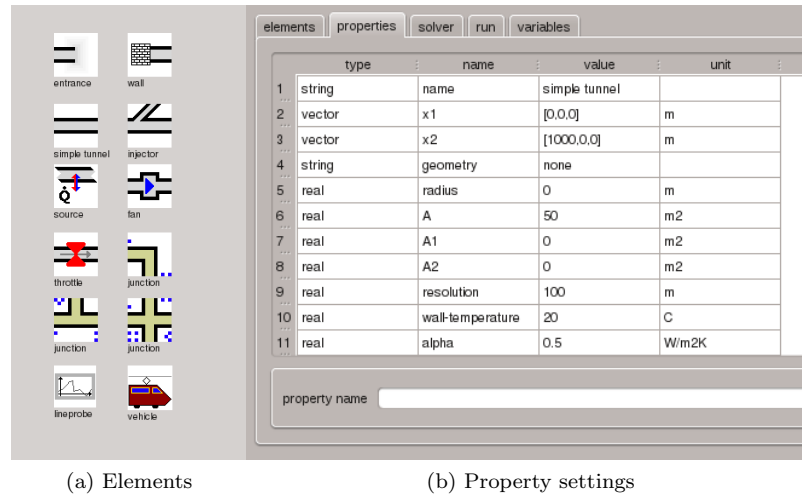


Figure 4.1: Overview of a NUMSTA3 file

4.2 OpenFOAM - Open Source Field Operation and Manipulation

OpenFOAM is foremost a C++ library, used primarily to create executables, known as applications. The applications fall into two categories: solvers that are each designed to solve a specific problem in continuum mechanics, and utilities that are designed to perform tasks that involve data manipulation. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetism. One of OpenFOAM's strengths is that new solvers and utilities can be created by its users.[20] [21]

The input for an OpenFOAM case is located in several folders. A certain folder structure is necessary to enable a simulation without any errors. Figure 4.2 shows an example of the basic structure of an OpenFOAM case, where words in blue letters are folders and words in black letters are files.

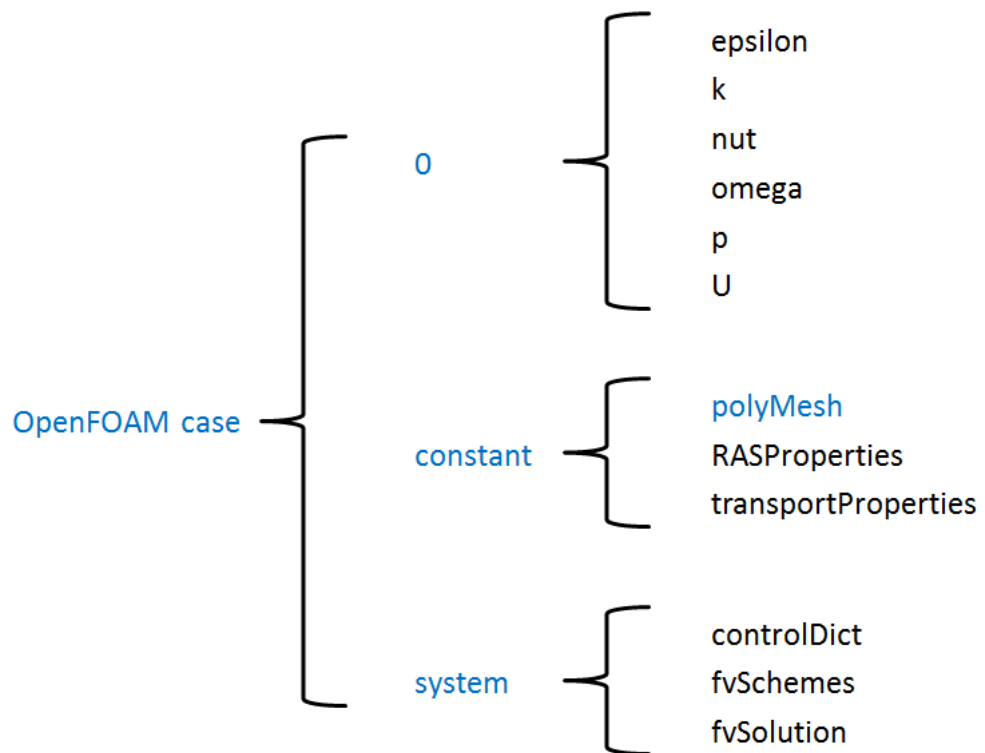


Figure 4.2: Basic structure of an OpenFOAM case

All files in the “0” folder are used to define initial values at the boundaries and the internal field. The folder “polyMesh” includes the geometry. Different turbulence models can be chosen and edited in the “RASProperties” file. The transport model and viscosity are defined in the “transportProperties” file. In the “controlDict” file general settings like steady or unsteady simulation, start time, end time, write interval and several more are defined. The “fvSchemes” file is used to define numerical schemes (upwind differencing, central differencing, linear upwind differencing, etc.) and interpolation schemes (e.g. linear). In the “fvSolution” file solver settings, relaxation factors and tolerances (the stop criterion of the simulation) are defined. OpenFOAM’s solvers and schemes, which are used in the course of this diploma thesis, are explained on the official OpenFOAM homepage. [22]

4.3 FDS 5 - Fire Dynamics Simulator 5

FDS is an open source CFD model of fire-driven fluid flow. It solves numerically a form of Navier-Stokes Equations appropriate for low-speed, thermally-driven flow with an emphasis on smoke and heat transport from fires. The conservation equations of mass, momentum and energy are approximated with a finite differences scheme on a three-dimensional, rectilinear grid. Thermal radiation is computed using a finite volume technique on the same grid as the flow solver. The core algorithm is an explicit predictor-corrector scheme that is second order accurate in space and time. Turbulence is treated by means of the Smagorinsky form of Large Eddy Simulation (**LES**). It is possible to use a Direct Numerical Solution (**DNS**) if the underlying numerical grid is fine enough. Lagrangian particles are used to simulate smoke movement, sprinkler discharge and fuel sprays. FDS can be used to model the following phenomena:

- Low speed transport of heat and combustion products from fire
- Radiative and convective heat transfer between the gas and solid surface
- Pyrolysis
- Flame spread and fire growth
- Sprinkler, heat detector and smoke detector activation
- Sprinkler sprays and suppression by water

Although FDS has been designed for fire simulations, it can be used for low-speed fluid flows without including fires or thermal effects. [23] [24] DNS can’t be used for big geometries like tunnels, because the fine grid would cause very long computing periods.

A case in FDS must be written in a FDS file (text file), which makes an immediate check of changes due to missing visualization impossible. However the FDS file can be exported from other softwares like Blender, which enables a visualization of the geometry and immediate checks. An FDS input file consists of blocks which are started with an “&” sign and ended with a “/” sign. Inside the different blocks parameters

Description of Used Programs

can be defined. The basic blocks, which are used within the context of this diploma thesis, are:

- &HEAD - to define the job name
- &TIME - to define the simulation time
- &MISC - to define the ambient state
- &MESH - to define the computing domain
- &MATL - to define the material of a surface
- &SURF - to define the surface of an obstruction or a vent
- &OBST - to create an obstruction
- &VENT - to create a vent
- &DEVC - to create a device, which writes CVS files
- &SLCF - to create a slice

Parameters in a block are separated with a “,” or a line break.

The basic structure of a FDS input file could be:

```

&HEAD  CHID          =      'job name'
        TITLE        =      'discriptive text' /

&TIME  T_BEGIN      =      t1
        T_END        =      t2 /

&MESH  ID           =      'name_MESH'
        XB           =      x1,x2,y1,y2,z1,z2 /

&MATL  ID           =      'name_MATL'
        CONDUCTIVITY =      C1
        SPECIFIC_HEAT =     C2
        DENSITY      =      C3 /

&SURF  ID           =      'name_SURF'
        THICKNESS    =      K1
        MATL_ID      =      'name_MATL' /

&OBST  ID           =      'name_OBST'
        XB           =      x1,x2,y1,y2,z1,z2 /

```


Description of Used Programs

| | | | |
|-------|----------|---|-------------------|
| | SURF_ID | = | 'name_SURF' / |
| &VENT | ID | = | 'name_VENT' |
| | XB | = | x1,x2,y1,y2,z1,z2 |
| | SURF_ID | = | 'name_SURF' / |
| &DEVC | ID | = | 'name_DEVC' |
| | QUANTITY | = | 'PRESSURE' |
| | XYZ | = | x,y,z/ |
| &SLCF | ID | = | 'name_SLCF' |
| | PBZ | = | z |
| | QUANTITY | = | 'VELOCITY' / |

5 Differences between used Software

5.1 Geometry

5.1.1 Geometry in NUMSTA3

NUMSTA3 can't calculate all geometries because it is a one-dimensional CFD software. All geometries must be specified by the area of the cross-section and a perimeter. A hydraulic diameter is calculated and a **Hagen-Poiseulls-pipe-flow** is used for the calculation. Special elements with a length of zero can be used to define bows like a "L"- or "T"-element (Figure 5.1). For these elements ξ -values can be defined to enable the calculation of a pressure loss. The angle of the bow cannot be defined, bit it is represented by the ξ -value. Long pipes with a continuous curvature radius can't be defined. Tunnel elements are subdivided in small sections (cells) to calculate physical quantities in each cell. The mesh is one-dimensional and pictured in Figure 5.2.

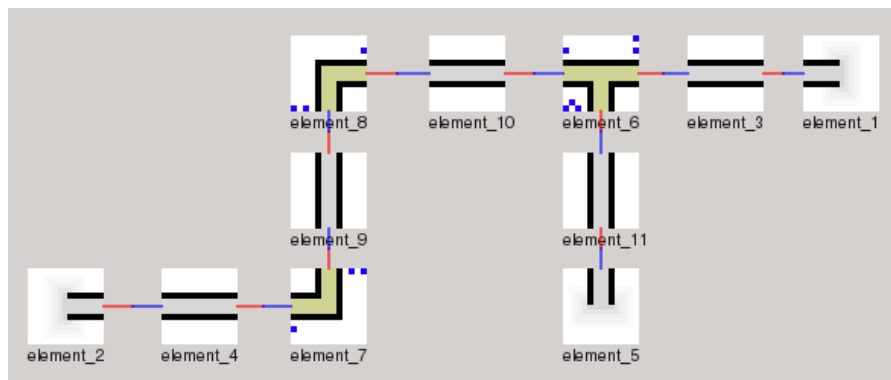


Figure 5.1: Geometry in NUMSTA

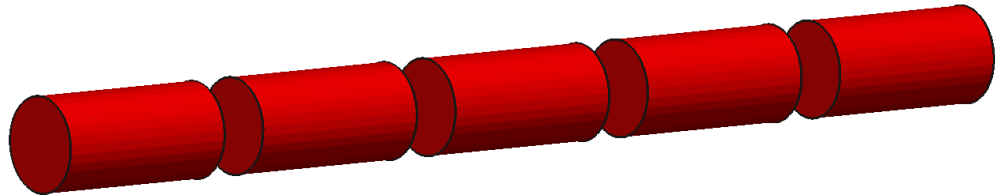


Figure 5.2: Mesh in NUMSTA

5.1.2 Geometry in OpenFOAM

The most flexible software which is used in the course of this diploma thesis is OpenFOAM. Structured and unstructured grids can be used for all geometries. It's even possible to refine the grids near walls or areas where high resolutions are necessary. Figure 5.3 shows a geometry with an unstructured grid.

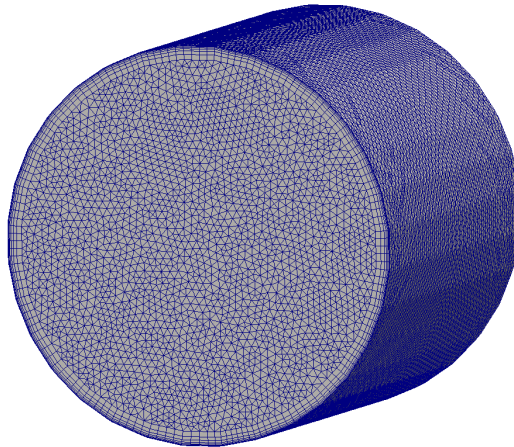


Figure 5.3: Geometry in OpenFOAM

5.1.3 Geometry in FDS

FDS also allows simulating any type of geometry, but the grid must be structured and cells are always cuboids. Based on the characteristics of FDS, all geometries like obstructions, holes, meshes, vents and slices are created by defining two points. This leads always to cubic geometries as pictured in Figure 5.4. So it's possible to create a round geometry, if the cell size is small enough, but the geometry is built up by cuboids. Figure 5.5 shows a round geometry, created in FDS with cells that are too large. Cells that are too small cause very long computing durations, while cells that are too big result in lower accuracy. For this reason FDS is limited in the application

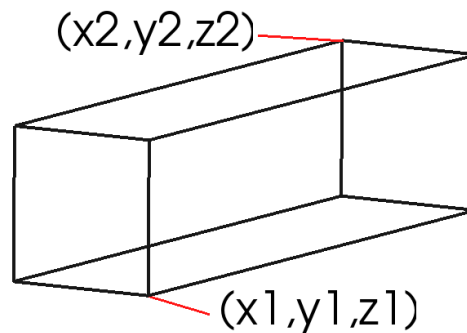


Figure 5.4: FDS: define geometry

of round and complex geometries.

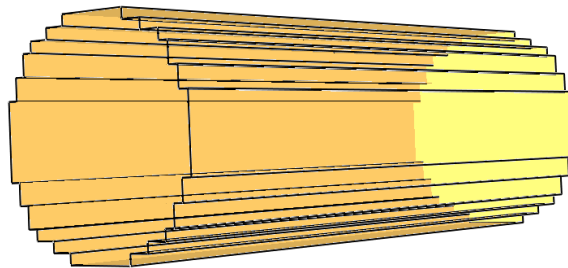


Figure 5.5: Round geometry in FDS

5.2 Boundary and Initial Conditions

All three programs calculate the values of physical quantities in cells. To determine the physical quantities, values of previous time steps and from neighboring cells are used. This procedure works as long as the examined cell isn't at the boundary of the computing domain or a value of the previous iteration step exists. If these prerequisites are not met, boundary and initial conditions are necessary.

5.2.1 Boundary and Initial Conditions - NUMSTA3

Initial conditions :

Temperature, pressure and velocity can be defined in the solver settings of NUMSTA3.

Inlet and outlet conditions :

The mode of an “entrance” element defines if it is an inlet or an outlet. The mode “inout” is the standard mode and allows the air to flow in both directions. The boundary conditions are automatically adapted to the local flow situation. The mode “flowrate” prescribes the specified flow rate which can either enter or exit the tunnel. The quantity “flowrate” defines a volume flux, which is entering or leaving the tunnel. Also the pressure and temperature of the inlet or outlet can be defined for each boundary element separately.

Boundary condition at walls :

NUMSTA3 is a one-dimensional CFD program. Because of that, no special boundary conditions must be defined at walls (except the wall temperature).

5.2.2 Boundary and Initial Conditions - OpenFOAM

OpenFOAM uses files to define the initial conditions and the boundary conditions at inlets, outlets and walls. The files are located in a folder named “0”. In these files the values or the gradient of the velocity, pressure, temperature etc. can be defined at each boundary.

Initial conditions :

Velocity, pressure and temperature at time step zero are defined as fixed values in the three files.

Inlet and outlet conditions :

Velocity, pressure and temperature at inlets and outlets can be defined as fixed values. Other possibilities are to define a zero gradient condition or to let OpenFOAM calculate the value at the boundary.

Boundary condition at walls :

The pressure at the walls has a zero gradient condition, while the velocity has a fixed value (the velocity usually is defined with 0,0,0 at static walls).

5.2.3 Boundary and Initial Conditions - FDS [24]

Initial conditions:

Both, temperature and pressure at the start of the simulation can be defined in the FDS input file. The velocity at the start of the calculation in the computing domain is zero by default. The default fluid in FDS is air.

Inlet and outlet conditions :

In FDS it's possible to define several surfaces and assign them values for pressure, temperature, velocity, volume flux, etc. One of the predefined surfaces in FDS has the ID 'OPEN'. This surface has the values for temperature and pressure which were defined at the initial conditions (if not defined explicitly) and is permeable. To induce a flow in the computing domain it is necessary to create a surface at the inlet of the tunnel. At this surface the velocity or the flow rate of the entering flow is defined. The temperature of the entering flow must be defined explicitly, otherwise the default value will be used.

Boundary condition at walls :

The temperature of walls can be defined by defining the surface of the wall and its temperature. The velocity of a flow is usually zero at a wall. But FDS uses a different definition for the velocity at walls. Usually the velocity follows a function like:

$$U(z) = \begin{cases} U_0 \left(\frac{z}{d}\right)^{\frac{1}{7}}, & \text{if } z < d \\ U_0, & \text{if } z > d \end{cases} \quad (5.1)$$

In Equation 5.1, z is the coordinate normal to the wall and d is the thickness of the boundary layer. A **slip boundary condition** can be worked out such that:

$$u_{ij,0} = u_{ij,1} \left(1 - \frac{d}{\delta_Z}\right) \quad ; \quad v_{ij,0} = v_{ij,1} \left(1 - \frac{d}{\delta_Z}\right) \quad (5.2)$$

The velocity $u_{ij,0}$ is the mean value of the velocity in the cell at the position $x = i, y = j, z = 0$ and δ_z is the height of the grid cell. For coarse grids, the boundary condition is almost a "free-slip condition" (the flow isn't slowed down at the wall), because $d \ll \delta_z$. FDS 5 uses a default boundary condition for LES where $u_{ij,0}$ and $v_{ij,0}$ are set to one-half the values of $u_{ij,1}$ and $v_{ij,1}$. The velocity in the z -direction is zero. This implies that the thickness of the boundary layer is assumed to be one half of the cell height.

Because of this boundary condition, a special method must be used to determine the wall shear stresses at walls. The wall shear stresses can be calculated with the help of Equation 2.42:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad \tau_w = \mu \left(\frac{du}{dy}\right)_w$$

But to get the wall shear stresses, the shear stress velocity must be determined. The shear stress velocity could be calculated by using Equation 2.44, the Log Law, Equation

2.43 and Equation 2.41:

$$\begin{aligned}
 u^+ &= y^+ && \text{for: } y^+ < 5 \\
 u^+ &= \frac{1}{\kappa} \ln(y^+) + B && \text{for: } y^+ > 30 \\
 u^+ &= \frac{u}{u_\tau} \\
 y^+ &= \frac{u_\tau y}{\nu}
 \end{aligned}$$

However, to eliminate the mathematical difficulties for handling the buffer layer and the log layer, Werner and Wengel proposed a simplification named the Power Law [31]:

$$u^+ = y^+ \quad \text{for: } y^+ < 11.81 \quad (5.3)$$

$$u^+ = 8.3 * (y^+)^{1/7} \quad \text{for: } y^+ > 11.81 \quad (5.4)$$

Figure 5.6 shows a comparison between the Log Law and the Power Law. In the region

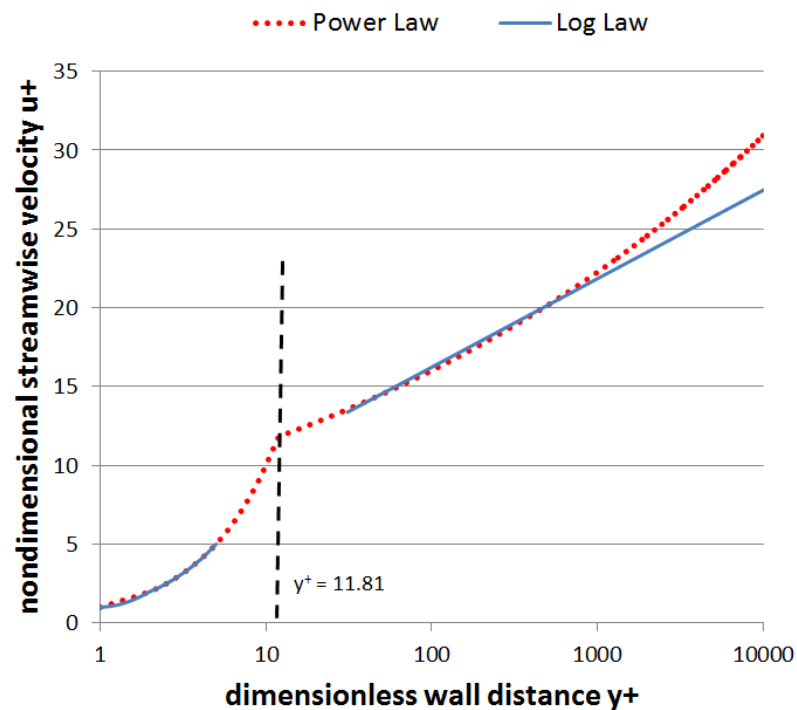


Figure 5.6: Log Law vs. Power Law

of $30 < y^+ < 1,000$ the difference between the Power Law and the Log Law is small

and the application of the Power Law is acceptable. So the wall shear stress at walls can be calculated with following equations (with $y = \delta_Z/2$):

$$|\tau_w| = \frac{2 * \mu * |u|}{\delta_Z} \quad \text{for: } y^+ < 11.81 \quad (5.5)$$

$$|\tau_w| = \rho \left[7.2 \left(\frac{\mu}{\rho * \delta_Z} \right)^{1.14} + 0.137 * \left(\frac{\mu}{\rho * \delta_Z} \right)^{1/7} |u| \right]^{1.75} \quad \text{for: } y^+ > 11.81 \quad (5.6)$$

5.3 Calculation of the Pressure Loss in a straight pipe

5.3.1 Calculation of the Pressure Loss - NUMSTA3

NUMSTA uses Equation 2.49 to calculate the pressure loss for each cell. The pressure loss along the entire tunnel is the sum of all cell pressure losses. This can be seen later in Chapter 6.1.5. The simulated pressure loss for simple geometries is nearly the same as the pressure loss of the empirical calculation.

5.3.2 Calculation of the Pressure Loss - OpenFOAM

OpenFOAM solves the equation for Conservation of Momentum. The exact approach for calculating the pressure differs from case to case. For example in the case of an incompressible solver, the equation for Conservation of Momentum looks different than in the case of a compressible solver.

5.3.3 Calculation of the Pressure Loss - FDS [24]

Fire Dynamics Simulator uses a special form of the Navier-Stokes Equations to calculate the pressure of each cell. This special form is a **Poisson Equation**, which will be established in this chapter. The pressure loss is the difference between the pressure values of two cells.

Derivative of the Poissons Equations:

The starting points are the Navier-Stokes Equations in vector form.

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \nabla) \mathbf{u} \right) + \nabla p = \rho \mathbf{g} + \mathbf{f}_b + \nabla \tau_{ij} \quad (5.7)$$

Transforming:

- subtract $(\rho_n(z, t)\mathbf{g})$ on both sides
- $(\mathbf{u}\nabla)\mathbf{u} = \nabla\frac{|u|^2}{2} - \mathbf{u}\times\omega$
- $\nabla p = \rho_n\mathbf{g} + \nabla\tilde{p}$
- divide both sides by ρ
- $\frac{\nabla\tilde{p}}{\rho} = \frac{\nabla\tilde{p}}{\rho_\infty} + \left(\frac{1}{\rho} - \frac{1}{\rho_\infty}\right)\nabla\tilde{p}$
- $H = \frac{|u|^2}{2} + \frac{\tilde{p}}{\rho_\infty}$
- ∇ on both sides

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \nabla\frac{|u|^2}{2} - \mathbf{u}\times\omega\right) + \nabla p - \rho_n\mathbf{g} = -\rho_n\mathbf{g} + \rho\mathbf{g} + \mathbf{f}_b + \nabla\tau_{ij}$$

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \nabla\frac{|u|^2}{2} - \mathbf{u}\times\omega\right) + \rho_n\mathbf{g} + \nabla\tilde{p} - \rho_n\mathbf{g} = -\rho_n\mathbf{g} + \rho\mathbf{g} + \mathbf{f}_b + \nabla\tau_{ij}$$

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\frac{|u|^2}{2} - \mathbf{u}\times\omega + \frac{\nabla\tilde{p}}{\rho} = \frac{1}{\rho}[(\rho - \rho_n)\mathbf{g} + \mathbf{f}_b + \nabla\tau_{ij}]$$

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\frac{|u|^2}{2} - \mathbf{u}\times\omega + \frac{\nabla\tilde{p}}{\rho_\infty} + \left(\frac{1}{\rho} - \frac{1}{\rho_\infty}\right)\nabla\tilde{p} = \frac{1}{\rho}[(\rho - \rho_n)\mathbf{g} + \mathbf{f}_b + \nabla\tau_{ij}]$$

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla H - \mathbf{u}\times\omega + \left(\frac{1}{\rho} - \frac{1}{\rho_\infty}\nabla\right)\tilde{p} = \frac{1}{\rho}[(\rho - \rho_n)\mathbf{g} + \mathbf{f}_b + \nabla\tau_{ij}]$$

$$\nabla^2 H = -\frac{\partial(\nabla\mathbf{u})}{\partial t} - \nabla\mathbf{F} \quad (5.8)$$

$$\mathbf{F} = -\mathbf{u}\times\omega + \left(\frac{1}{\rho} - \frac{1}{\rho_\infty}\right)\nabla\tilde{p} - \frac{1}{\rho}[(\rho - \rho_n)\mathbf{g} + \mathbf{f}_b + \nabla\tau_{ij}] \quad (5.9)$$

The pressure \tilde{p} exists on both sides of Equation 5.8. \tilde{p} on the left side is the pressure at the actual time step, and for \tilde{p} on the right side, the value of the previous time step is inserted.

5.4 Consideration of Heat Radiation

Radiation heat transfer processes are extremely important; they are involved in the exchange of thermal heat. However, Heat Radiation is not considered in all three

softwares. While OpenFOAM and FDS enable a consideration of heat radiation, there is no possibility to simulate heat radiation in the current version of NUMSTA3. According to RVS 09.02.31 [2], a fire efficiency parameter is used for the dimensioning of ventilation systems for road tunnels. Basic idea behind this parameter is the fact that a certain amount of the heat performance warms up the walls, but not the fluid. Hence, a reduced heat release rate is defined in NUMSTA3 (Equation 5.10).

$$HRR_{eff} = \eta_{fire} * HRR \quad (5.10)$$

The whole effective heat release is transferred to the fluid and no radiation warms up the wall. Theoretically, the rest of the heat performance ($HRR - HRR_{eff}$) would warm up the wall. But in NUMSTA3, the entire HRR and η_{fire} are not defined. Hence, no information about the fire efficiency or the entire heat release rate exists in NUMSTA3. Therefore the term $HRR - HRR_{eff}$ disappears and doesn't warm up the wall or the fluid. This is a major consequence for using NUMSTA3 and could cause wrong results.

6 Simulations for Comparing the Programs

It is necessary to calculate several test cases to investigate the differences between the three programs. An essential component of the analysis is to check the source codes (or technical reference guides) of the programs to identify the equations, numerical algorithms and boundary conditions which could cause differing results.

Starting with a complicated calculation is not very helpful because there are many effects which interfere with each other. For example a deviation of the velocity can be caused by different temperatures (different buoyancy), pressures, boundary conditions, etc. A more preferable way to investigate the differences is to do this step-by-step.

6.1 Straight Pipe Flow

One of the simplest examples of flows is the straight pipe flow. The pressure loss in a straight pipe can be calculated empirically and no measurement results are required to control the accuracy of the simulation results. The geometry of the pipe has to be the same for all three programs and the cells should have the same size, so that a comparison of the accuracies is possible. In order to make a statement about the magnitude of the error, which is not only valid for a single velocity, a variation of the velocity at the tunnel's entrance is advisable. Particularly FDS has a problem with its accuracy at flows of high speed.

6.1.1 Straight Pipe Flow - Empirical Calculation

In general the flow velocity in an air duct of tunnel ventilation systems won't exceed 20 m/s . Higher velocities would cause too high pressure losses, which would raise the required fan power and thus the costs of the system. Because of this fact, the maximum examined velocity is 20 m/s . Low velocities, short tunnel lengths and big cross-sectional areas cause a low pressure loss inside the tunnel. To obtain results for the pressure loss, which are approximately equal to or greater than 10 Pa , the geometry shown in Figure 6.1 is used. If pressure losses at this magnitude are rounded to three digits behind the comma, no additional uncertainty is produced. A quadratic cross section is needed for the calculation in FDS. A length of 50 m and a side length of 1 m at a minimum velocity of 5 m/s should create enough pressure loss to compare results without any mistakes caused by a lack of significant digits. The pressure losses are calculated with Equation 2.49 for 20 m/s , 10 m/s and 5 m/s . The complete analytical calculation is

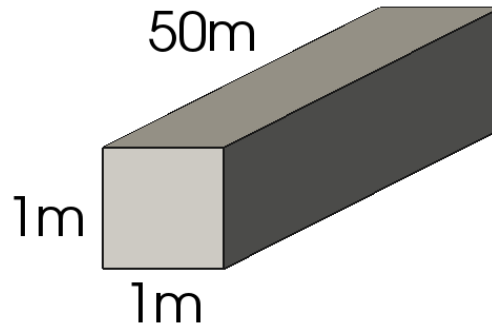


Figure 6.1: Straight pipe flow: Examined geometry

only performed for one velocity. The results for the other velocities can be read from Table 6.1.

Calculation for $U = 20 \text{ m/s}$:

$$D_{hydr} = 4 * \frac{A}{U} = 4 * \frac{1 \text{ m} * 1 \text{ m}}{4 * 1 \text{ m}} = 1 \text{ m}$$

$$Re = \frac{UD_{hydr}}{\nu} = \frac{20 \text{ m/s} * 1 \text{ m}}{15.35 * 10^{-6} \text{ m}^2/\text{s}} = 1,302,931$$

$$\lambda = 0.3164 * Re^{-\frac{1}{4}} = 0.3164 * 1,302,931^{-\frac{1}{4}} = 0.009365$$

$$\xi = \lambda \frac{L}{D_{hydr}} = 0.009365 \frac{50 \text{ m}}{1 \text{ m}} = 0.46825$$

$$\Delta p_V = \xi \frac{\rho}{2} U^2 = 0.46825 * \frac{1.188 \text{ kg/m}^3}{2} * (20 \text{ m/s})^2 = 111.256 \text{ Pa}$$

Because of the high Reynolds-Number, the flow is certainly turbulent and the length of the lead-in area can be calculated with Equation 2.48:

$$10 \leq \frac{x_E}{D_{hydr}} \leq 60 \quad \text{with} \quad \frac{x_E}{D_{hydr}} = 50$$

$$x_E = 50 D_{hydr} = 50 * 1 \text{ m} = 50 \text{ m}$$

The calculated lead-in length will be used in front and behind the examined area in all simulations to get reliable results.

Results for all velocities :

Table 6.1: Straight pipe flow: Pressure losses, empirical

| U [m/s] | Re | ξ | Δp_V [Pa] |
|-----------|-----------|-------|-------------------|
| 20 | 1,302,931 | 0.468 | 111.256 |
| 10 | 651,466 | 0.557 | 33.077 |
| 5 | 325,732 | 0.662 | 9.834 |

The empirical solution is valid for non-disturbed flows. For this reason all results of the simulations don't include the lead-in and lead-out area. Only the results for a length of 50 m in the middle of the tunnel (after the lead-in area and before the lead out area) are compared. In this region a non-disturbed flow is ensured.

6.1.2 Straight Pipe Flow - NUMSTA3

The model in NUMSTA3 is built up from simple blocks (shown in Figure 6.2). An entrance element (named inlet) is used to define the flow speed by defining the volume flux (Figure 6.3). Two tunnel elements (named tunnel1 and tunnel3) are used in the model to create the lead-in and lead-out area. The examined part of the tunnel is created with an element named tunnel2. The cells inside the examined area have a length of 0.2 m while the length of the cells in the lead-in area and lead-out area is 0.5 m to reduce the computing time. The pipe drag coefficient λ must be defined manually for every velocity (Figure 6.4). The same value as in Chapter 6.1.1 are used in the NUMSTA3 calculation. The second entrance block (named outlet) is used to enable outflow of the air. The velocity of the flow has the same value over the entire cross-section, because NUMSTA3 is a one-dimensional CFD program. The results for the pressure loss can be read directly from the user interface (Figure 6.5). The numbers in the Figure 6.5 are equal to the pressure loss in Pa along the entire element and the ">" marks the flow direction. To write the pressure value of each cell in a file a lineprobe is used. A lineprobe records all physical quantities along a defined line in a CVS-file. The results for all velocities are given in Table 6.2. The solver is set to solve the equations for a steady system. The heat model for adiabatic systems is used. An approximate Riemann solver of Roe (first order upwind difference scheme) is used to simulate the flow inside the tunnel.

Table 6.2: Straight pipe flow: Pressure losses, NUMSTA3

| U [m/s] | Δp_V [Pa] after 50 m | Error [%] |
|-----------|------------------------------|-----------|
| 20 | 113.311 | 1.85 |
| 10 | 33.373 | 0.89 |
| 5 | 9.928 | 0.96 |

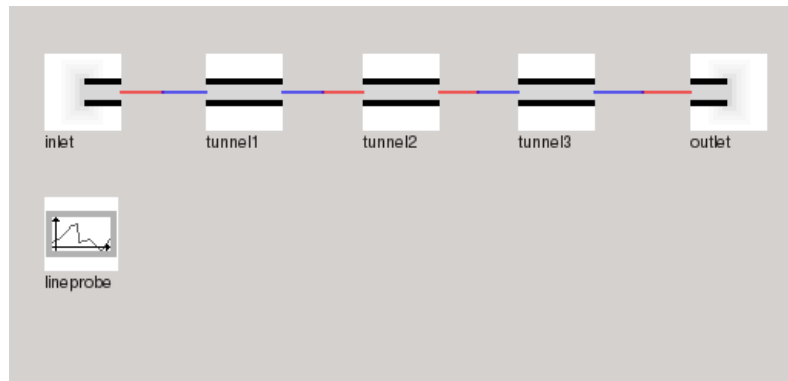


Figure 6.2: Straight pipe flow: Model in NUMSTA3

| elements | | | | |
|------------|--------|---------------------|---------|------|
| properties | | | | |
| solver | | | | |
| run | | | | |
| variables | | | | |
| | type | name | value | unit |
| 1 | string | name | inlet | |
| 2 | real | temperature | 20 | C |
| 3 | real | pressure | 1013.25 | mbar |
| 4 | real | reference-elevat... | 0 | m |
| 5 | real | lapse-rate | -999 | K/km |
| 6 | real | flowrate | 20 | m3/s |
| 7 | real | relative-humidity | 0 | % |

Figure 6.3: Straight pipe flow: Definition of the flow rate

| elements | | | | |
|------------|--------|----------------------|---------|--------|
| properties | | | | |
| solver | | | | |
| run | | | | |
| variables | | | | |
| | type | name | value | unit |
| 11 | real | alpha | 0.5 | W/m2K |
| 12 | real | perimeter | 4 | m |
| 13 | real | friction-coefficient | 0.0143 | 1 |
| 14 | real | water-flux | 0 | kg/m2h |
| 15 | real | traffic-heat | 0 | W/m |
| 16 | vector | up | [0,0,1] | m |
| 17 | string | write | no | |

Figure 6.4: Straight pipe flow: Definition of λ

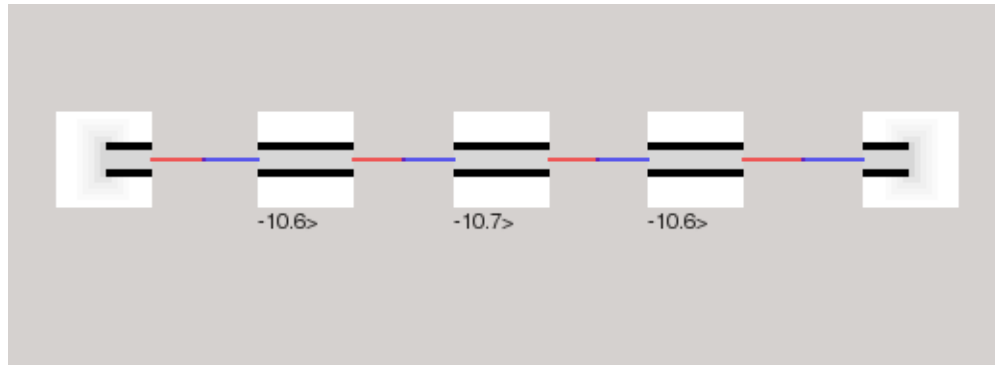


Figure 6.5: Straight pipe flow: Pressure loss in NUMSTA3

6.1.3 Straight Pipe Flow - OpenFOAM

The mesh in OpenFOAM is created with the help of two more programs: **Gmsh** and **enGrid**. Gmsh is a free three-dimensional finite element grid creator. It enables parametric meshing, which is very helpful to edit an existing mesh. **enGrid** is an open-source mesh generation software with CFD applications in mind. [26] [27]

The cross section of the tunnel is created in Gmsh (Figure 6.6a). All cells near the center of the tunnel have the size of $0.05 \text{ m} * 0.05 \text{ m}$. The cells near the wall are refined to enable the calculations of the flow conditions in the boundary layer (Figure 6.6b). The nearest cell to the wall has such a height, so that the y^+ -value is about 50. The height depends on the velocity of the flow. According to a discussion in “CFD Online” [28] the turbulence model ($k-\omega$ -SST model) produces the most accurate results in the region, where the log-law is applicable. As explained in Chapter 2.4 the log-law can be formulated for $y^+ > 30$.

Estimation of the Cell Height at Walls

The height of the first cell next to the wall is very important. To include the influence of the boundary layer on the pressure loss and on the flow profile, the first cell must not be too small or too big. How to calculate y^+ has already been explained in Chapter 2.4. To calculate the height of the first cell, this correlation must be rearranged:

$$y = \frac{y^+ \nu}{u_\tau} \quad (6.1)$$

y^+ and ν are already known. ν is a material characteristic and is a function of temperature. y^+ has been chosen above. The friction velocity is a function of the wall shear stresses (look Chapter 2.4). τ_w can be calculated with Equation 6.2 [4]:

$$\tau_w = \frac{\rho}{8} \lambda u^2 \quad (6.2)$$

Both, density and velocity are known. λ can be calculated with Equation 2.56 (already done in Chapter 6.1.1). Since all necessary variables are known, the height of the first cell can be estimated. The following example shows the calculation of y for a velocity of 20 m/s :

$$\tau_w = \frac{1.188 \text{ kg/m}^3}{8} * 0.009365 * (20 \text{ m/s})^2 = 0.556281 \text{ N/m}^2$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} = \sqrt{\frac{0.556281 \text{ N/m}^2}{1.188 \text{ kg/m}^3}} = 0.68428 \text{ m/s}$$

$$y = \frac{50 * 15.35 * 10^{-6} \frac{\text{m}^2}{\text{s}}}{0.68428 \text{ m/s}} = 1.1216 \text{ mm}$$

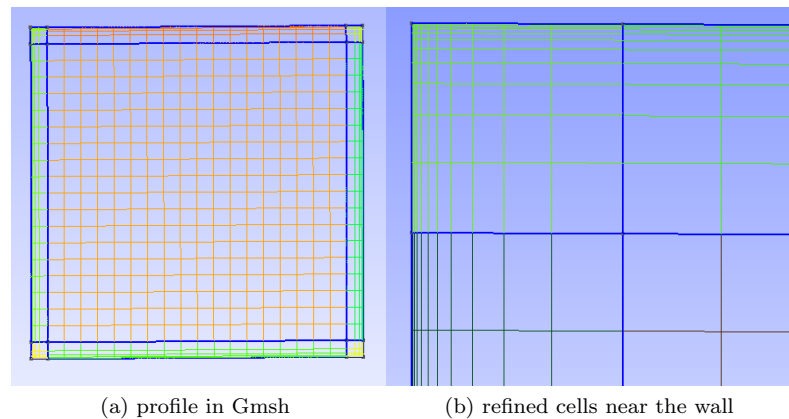


Figure 6.6: Straight pipe flow: Tunnel's profile created in Gmsh

A cross section with the calculated cell height near the wall can be created in Gmsh. This profile can be imported in enGrid and extruded along the x-axis to create the volume of the tunnel. After the extrusion, the cells near the core have the size of $0.05 \text{ m} * 0.05 \text{ m} * 0.2 \text{ m}$ in the examined area of the tunnel and $0.05 \text{ m} * 0.05 \text{ m} * 0.5 \text{ m}$ at the lead-in area and lead-out-area to reduce the computing time. A part of the created grid is shown in Figure 6.7. The cells on the left side of the figure have the same profile as the cells on the right side, but they are longer.

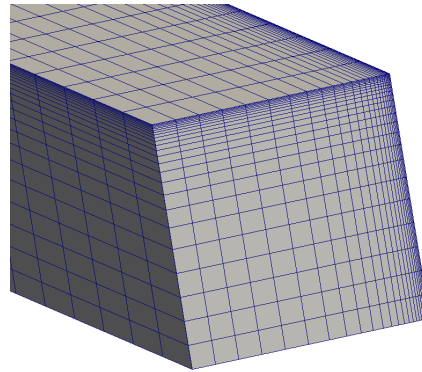


Figure 6.7: Straight pipe flow: Tunnel's grid created in enGrid

The used solver settings are:

- simpleFoam for a steady scheme
- RASModel $k-\omega$ -SST
- Gauss linear for gradSchemes
- Gauss linearUpwind Gauss for U divSchemes
- Gauss upwind for k divSchemes
- Gauss linear corrected for laplacianSchemes
- GAMG for the pressure
- smoothSolver for U, ω and k

The listed terms are no mathematical terms. They are predefined commands to select schemes in OpenFOAM. The meaning of each term can be read in the OpenFOAM User's Guide [20]. The resulting flow profiles are shown in Figures 6.8 to 6.10:

Even if the velocity at the wall is equal to zero (boundary condition), the value zero doesn't show up in the results for the flow profiles. That's because OpenFOAM displays the arithmetic average of the cells. The nearest cell to the wall contains low flow speeds, but the arithmetic mean is not zero. It is important to regard the unit of the pressure loss in OpenFOAM. OpenFOAM uses a normalized pressure for incompressible flows:

$$p' = \frac{p}{\rho} \quad (6.3)$$

For incompressible flows the density $\rho = \text{const.}$ Therefore the following transformation for the pressure term in Equation 2.16 is allowed:

$$\frac{1}{\rho} \frac{\partial p}{\partial x_i} = \frac{\partial p'}{\partial x_i} \quad (6.4)$$

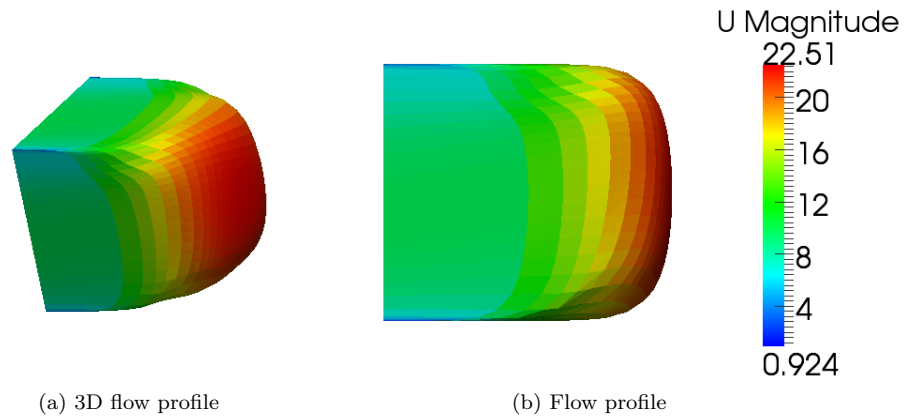


Figure 6.8: Straight pipe flow: Flow profile at $U_{input} = 20 \text{ m/s}$, OpenFOAM

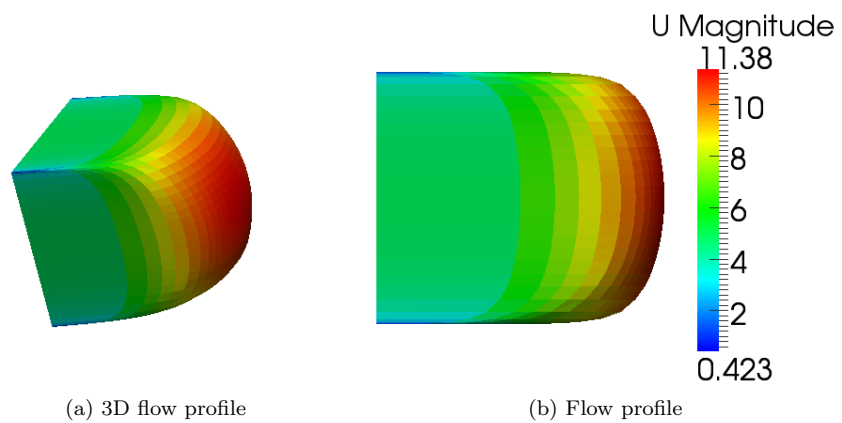
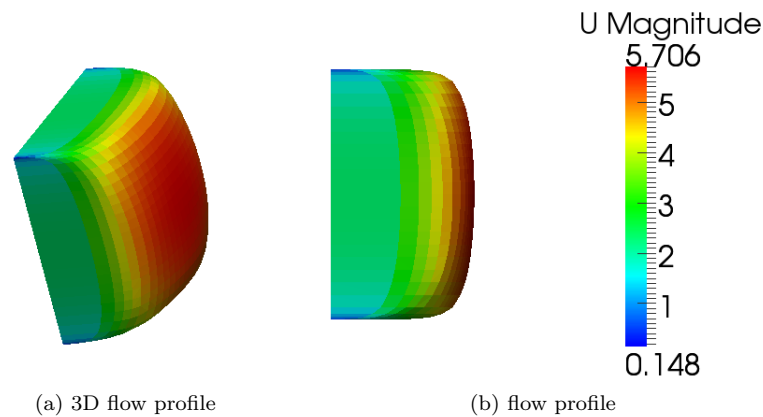


Figure 6.9: Straight pipe flow: Flow profile at $U_{input} = 10 \text{ m/s}$, OpenFOAM


 Figure 6.10: Straight pipe flow: Flow profile at $U_{input} = 5 \text{ m/s}$, OpenFOAM

Because of this transformation the pressure calculated in OpenFOAM must be multiplied by ρ to make a comparison between the pressures of OpenFOAM and the empirical solution possible. The transformed results for all velocities can be read from Table 6.3.

Table 6.3: Straight pipe flow: Transformed pressure losses, OpenFOAM

| $U [\text{m/s}]$ | $\Delta p_V [\text{Pa}]$ after 50 m | Error [%] |
|------------------|-------------------------------------|-----------|
| 20 | 108.556 | -2.43 |
| 10 | 29.460 | -10.21 |
| 5 | 9.049 | -7.98 |

6.1.4 Straight Pipe Flow - FDS

A free software called Blender is used to create the geometry for FDS. One of Blender's advantages is that the geometry is displayed on the screen. FDS has no graphical user interface that allows to view the geometry. The cells in the examined region of the tunnel have a size of $0.05 \text{ m} * 0.05 \text{ m} * 0.05 \text{ m}$. Because of the explicit unsteady solution scheme of FDS the Courant-Number is important.

$$CFL < \frac{U\Delta t}{\Delta x} \leq 1$$

$$\Delta t \leq \frac{CFL\Delta x}{U} = \frac{1 * 0.2\text{m}}{20 \text{ m/s}} = 0.01\text{s}$$

So at a velocity of 20 m/s Δt must be smaller than 0.01 s to achieve a convergent solution. This time step is automatically calculated in FDS. A simulation duration of

120 s should create a quasi-steady solution. Splitting the computational area in more than one mesh allows defining different cell sizes or to parallelize the calculation. The model is shown in Figure 6.11. Vents are placed on the inlet and outlet to define the boundary conditions. Several &DEVC blocks are used to obtain the velocity and pressure at different locations. These devices are placed on a symmetry plane of the geometry to make a visualization of the slip-condition possible. The grid of the model is shown in Figure 6.12. Appendix A.1 includes the FDS input file. The devices are also used to display the flow profile in Microsoft Excel. The results for the flow profiles and pressure losses are shown in the following figures and tables.

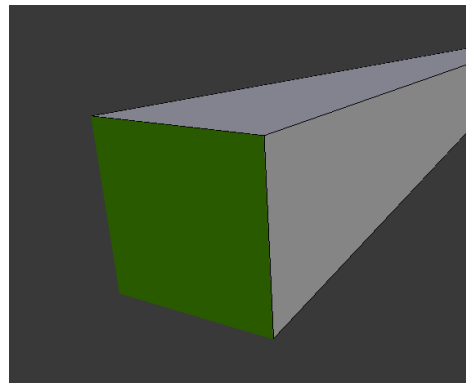


Figure 6.11: Straight pipe flow: FDS model

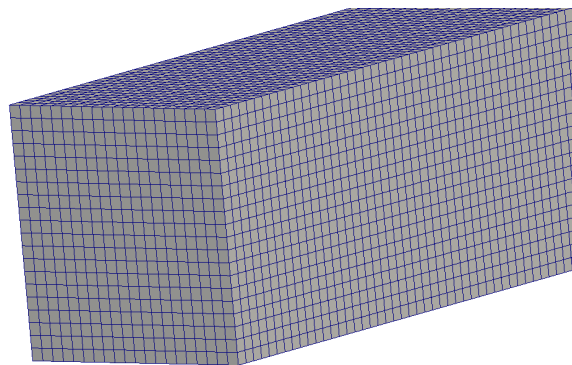


Figure 6.12: Straight pipe flow: Grid in FDS

In Figures 6.13 to 6.15 it is easy to see that there are high tangential velocities near the wall. Therefore the velocity gradient is not realistic and the pressure loss can't be calculated correctly (look at Equation 2.19).

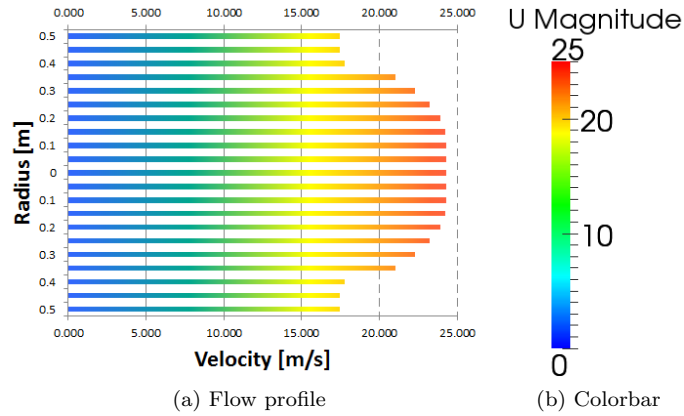


Figure 6.13: Straight pipe flow: Flow profile at $U_{input} = 20 \text{ m/s}$, FDS

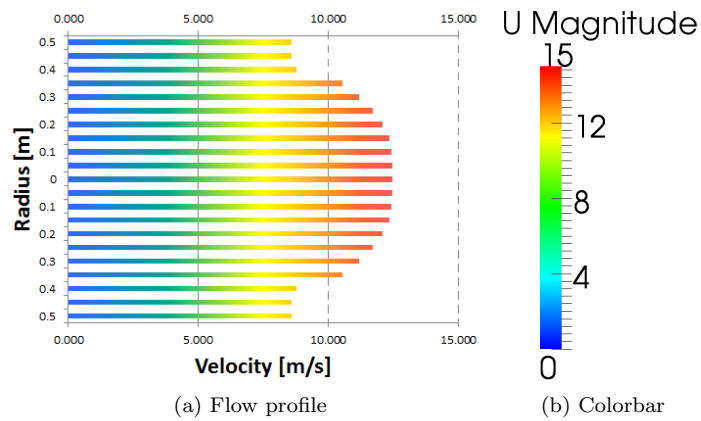


Figure 6.14: Straight pipe flow: Flow profile at $U_{input} = 10 \text{ m/s}$, FDS

Table 6.4: Straight pipe flow: Pressure losses, FDS

| $U [\text{m/s}]$ | $\Delta p_V [\text{Pa}]$ after 50 m | Error [%] |
|------------------|-------------------------------------|-----------|
| 20 | 141.910 | 27.55 |
| 10 | 39.795 | 20.31 |
| 5 | 11.910 | 21.11 |

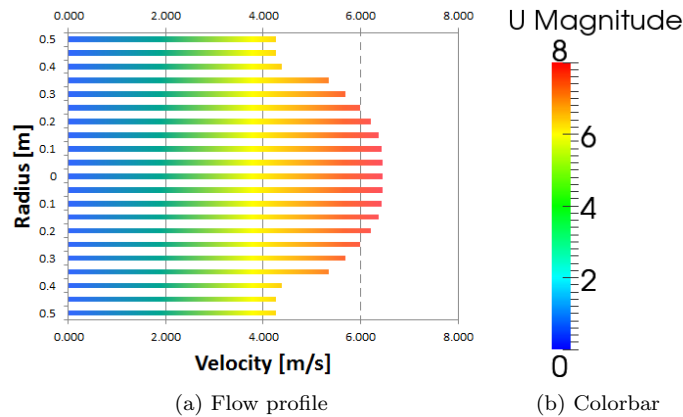


Figure 6.15: Straight pipe flow: Flow profile at $U_{input} = 5 \text{ m/s}$, FDS

6.1.5 Straight Pipe Flow - Comparison of Results

The first big difference between the results of the programs is the computing duration. The reason for that is the different amount of cells. NUMSTA3 has the least amount of cells, because it is a one-dimensional CFD program. FDS uses more cells to simulate the straight pipe flow because the computing domain is discretized in three directions. Additionally a time discretization is necessary for FDS to perform a LES. The highest amount of cells is used in OpenFOAM because Gmsh allows to refine the cell size near walls to simulate the boundary layer more accurately. Table 6.5 shows the number of cells of each program and the required computing time for a convergent solution. (A comparison of cell amount and computing time won't be explicitly mentioned for further simulations, because the proportions are always the same.)

Table 6.5: Straight pipe flow: Required computing time of the programs

| Name | Amount of cells | Computing time[s] |
|----------|-----------------|-------------------|
| NUMSTA3 | 450 | 300 |
| OpenFOAM | 352,800 | 5,720 |
| FDS | 300,000 | 28,800 |

Figure 6.16 to Figure 6.21 show the trends of the pressure losses inside the tunnel. NUMSTA3's results are nearly the same as the empirical results for all three velocities, while OpenFOAM's results are about 10% lower for certain velocities. The pressure losses in FDS are higher than the other ones, but for low velocities (5 m/s) they get close to the exact value.

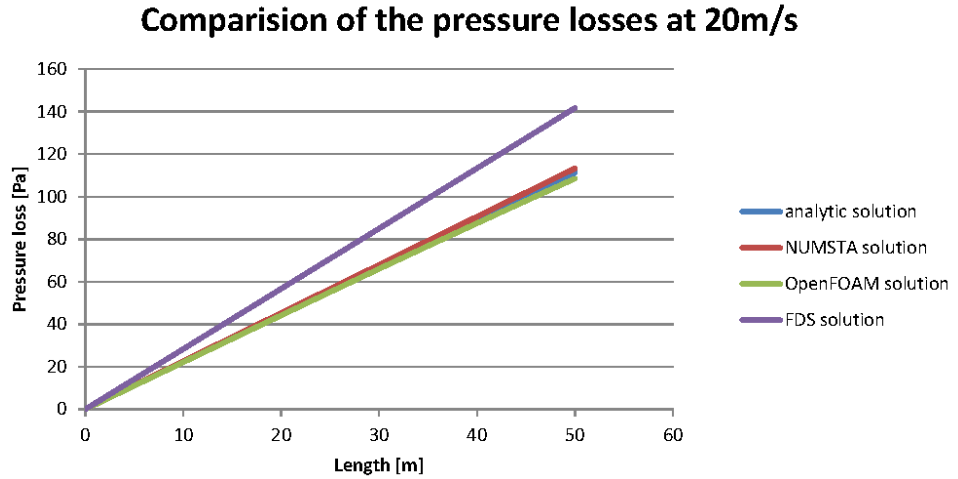


Figure 6.16: Straight pipe flow: Comparison of pressure losses for $U = 20 \text{ m/s}$

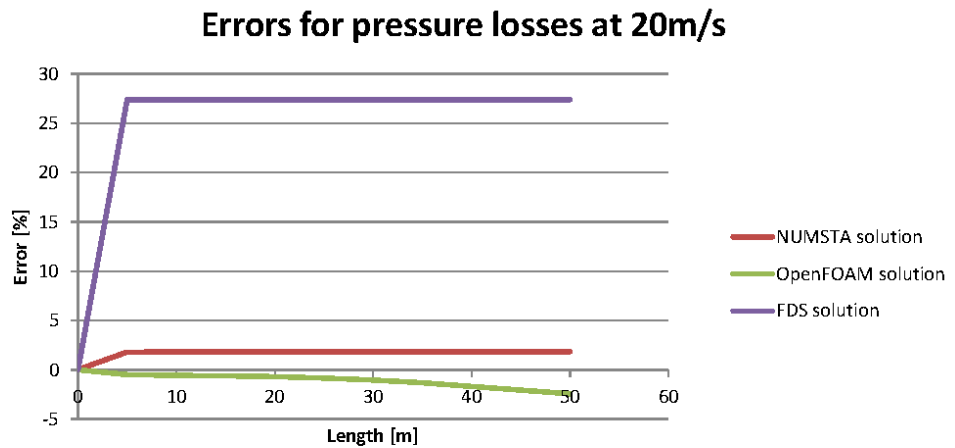


Figure 6.17: Straight pipe flow: Errors of pressure losses for $U = 20 \text{ m/s}$

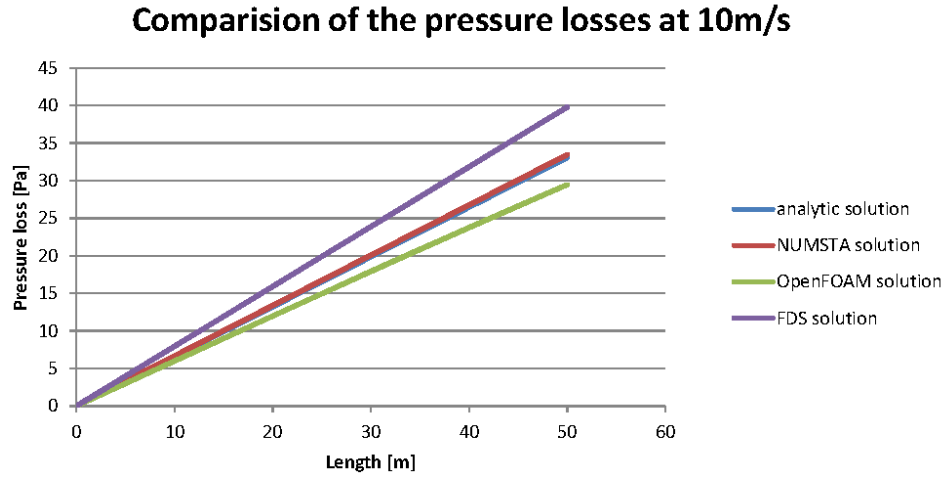


Figure 6.18: Straight pipe flow: Comparison of pressure losses for $U = 10 \text{ m/s}$

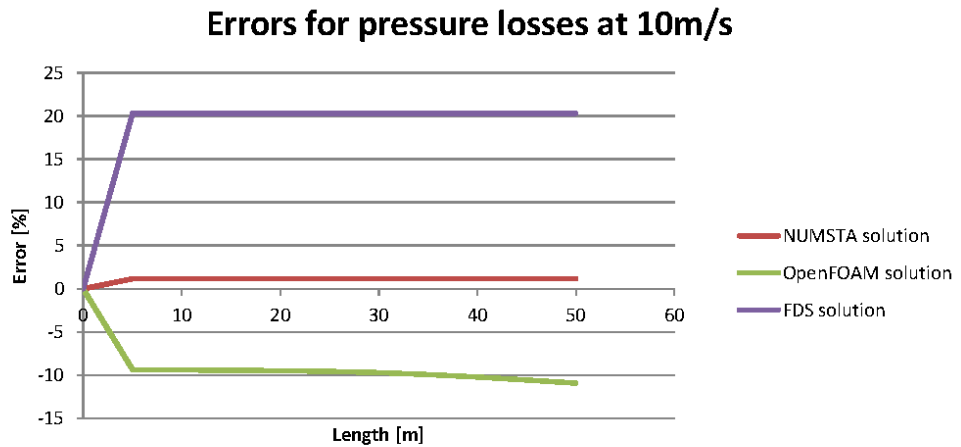


Figure 6.19: Straight pipe flow: Errors of pressure losses for $U = 10 \text{ m/s}$

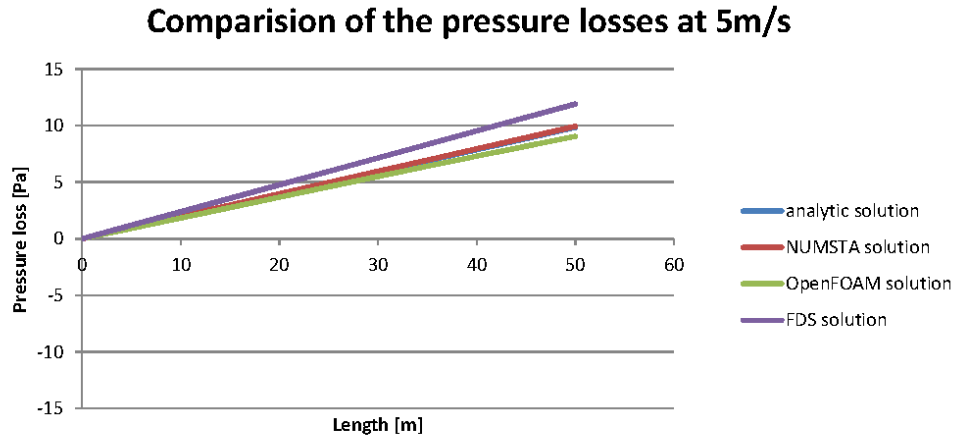


Figure 6.20: Straight pipe flow: Comparison of pressure losses for $U = 5 \text{ m/s}$

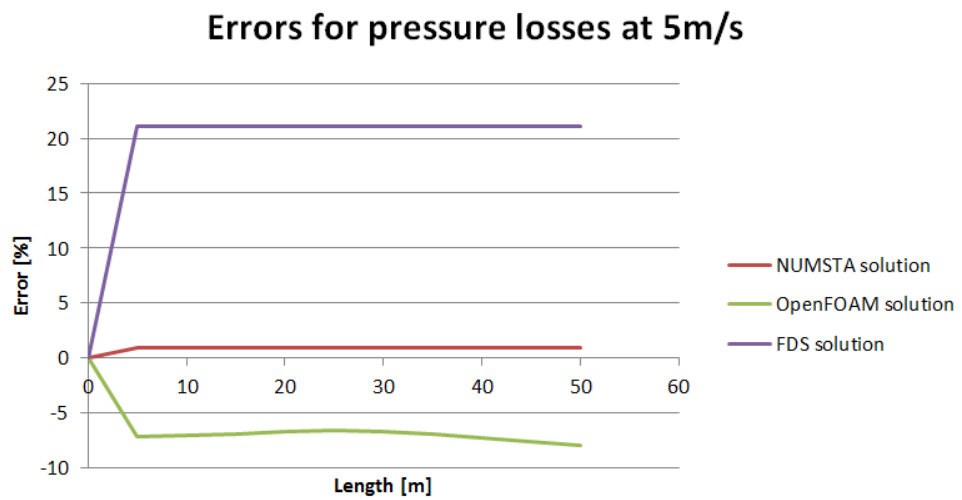


Figure 6.21: Straight pipe flow: Errors of pressure losses for $U = 5 \text{ m/s}$

6.1.6 Straight Pipe Flow - Reasons for Differing Results

NUMSTA3's results for the pressure loss are nearly identically to the empirical results. That is because identical equations are used.

OpenFOAM produces results which are quite exact for an inlet velocity of 20 m/s but the deviation for 10 m/s is too high to be neglected. A possibility for the deviation could be the resolution of the cells near the walls. The influence of the boundary layer has been described in Chapter 2.4. If the mesh is too coarse, the boundary layer can't be resolved good enough to take the influence of the boundary layer into account. If the height of the first cell is inside the buffer layer, the log-law is not applicable and the turbulence model cannot determine the correct pressure loss. As mentioned in Chapter 6.1.3 the dimensionless wall distance y^+ has been estimated to a value of 50 for all three velocities. Because of that, the height of the cell next to the wall is different for each velocity. However this was just an estimation and the mesh has been customized a few times to generate a $yPlus$ of about 50 in OpenFOAM. Simulations with a $yPlus$ of 50 produced results with unacceptable deviations like 10% as can be seen in Table 6.3. Additional simulations are performed to investigate the reason for the wrong pressure losses. To reduce the computing time, a tunnel with a length of 20 m is simulated and the pressure loss is extrapolated linearly to a length of 50 m. Additionally only a quarter of the tunnel section is simulated and two symmetry planes are used to produce the same results like a tunnel with the complete section.

Reason for the error: OpenFOAM uses a definition of $yPlus$, which is not equal to y^+ . Usually Equation 2.41 is used to calculate y^+ . The definition of OpenFOAM can be found in the Fluent User's Guide [29] and is named y^* or $ySTAR$. Because of the different definitions of y^+ and $yPlus$, the ideal value for $yPlus$ is not 50!

$$y^+ = \frac{\sqrt{\frac{\tau_w}{\rho}} * y}{\nu} \quad (6.5)$$

$$y^* = yPlus = \frac{C_\mu^{0.25} \sqrt{k} * y}{\nu} \quad (6.6)$$

The size of the smallest cell near the wall has been created with a height of 3.8617 mm (Chapter 6.1.3). OpenFOAM uses the center of the cell for the calculation of the dimensionless wall distance. In this example y would be:

$$y = \frac{3.630mm}{2} = 1.931 \text{ mm}$$

The factor C_μ is defined in the RASProperties file of OpenFOAM and has a default value of 0.09. k is the turbulent kinetic energy, ρ the density, ν the kinematic viscosity and τ_w the wall shear stresses. τ_w and k can be seen in the visualization of the OpenFOAM results in Paraview. The calculated value for $yPlus$ is:

$$yPlus = y^* = \frac{0.09^{0.25} * \sqrt{0.156621 \frac{m^2}{s^2}} * 1.931 * 10^{-3} m}{15.35 * 10^{-6} \frac{m^2}{s}} = 27.02$$

This value is equal to the maximum of the colorbar in Figure 6.23. The value for y^+ is:

$$y^+ = \sqrt{\frac{0.1697 \frac{N}{m^2}}{1.188 \frac{m}{s}}} * 1.931 * 10^{-3} m * \frac{1}{15.35 * 10^{-6} \frac{m^2}{s}} = 49.55$$

This example shows that $yPlus$ can't directly be compared with y^+ . And if the value $yPlus$ is set to 50, y^+ is much higher. Figure 6.22 to 6.23 show the difference of the mesh and the resolved y^+ near the walls.

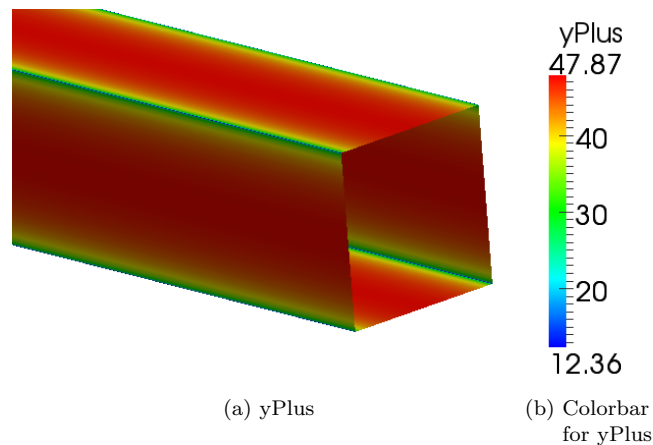


Figure 6.22: Straight pipe flow: y^+ near the wall for $U = 5 \text{ m/s}$ of the original simulation

After the refinement the results for all three velocities are near to the empirical solution. Several resolutions with different y^+ -values have been simulated and at a value of about 30 to 50, the results are very close to the exact solution. The pressure loss for all velocities can be read from Table 6.6.

Table 6.6: Straight pipe flow: Pressure losses for refined meshes, OpenFOAM

| $U [\text{m/s}]$ | $\Delta p_V [\text{Pa}]$ after 50 m | Error [%] |
|------------------|-------------------------------------|-----------|
| 20 | 133.444 | +2.11 |
| 10 | 33.606 | +1.60 |
| 5 | 9.823 | -0.11 |

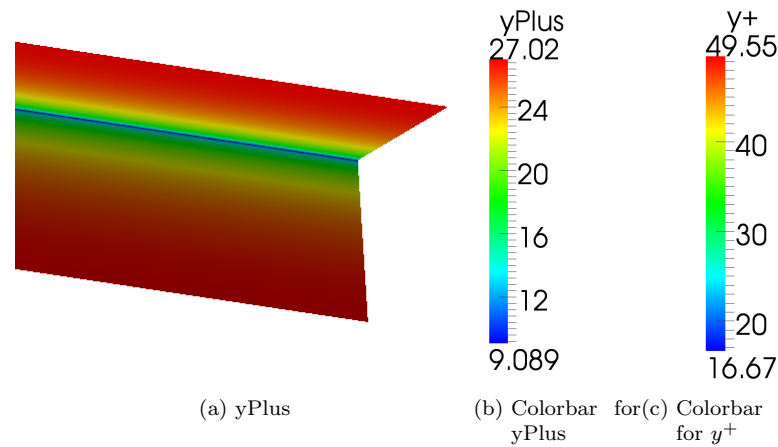


Figure 6.23: Straight pipe flow: y^+ near the wall for $U = 5 \text{ m/s}$ of the compare simulation

FDS uses a slip boundary condition and solves a **Poisson Equation** (Equation 5.8). This leads to deviations of the pressure losses. The smaller the flow rates are, the smaller is the error. Another reason for the error in FDS could be the coarse mesh. FDS provides no function to refine cells near walls and therefore it is not possible to simulate a correct pressure loss in FDS. To check whether the error is caused by the coarse grid, the entire mesh is refined to a cell length of 0.2 m, a cell height of 0.033 m and a cell width of 0.033 m. Although the number of cells is “low” (675,000 cells), the necessary computing performance is immense. Despite that, the accuracy of the simulation is not better than before. The comparative calculation shows that FDS is not suitable for the calculation of pressure losses caused by wall friction.

6.2 Sudden Change of the Cross-Sectional Area

The next test case is the calculation of a flow in a channel with a sudden change in the cross-sectional area. This abrupt enlargement causes recirculation and a detachment of the boundary layer. Additionally a pressure loss arises. All calculations are done under the same initial conditions as in Chapter 6.1 ($T = 293.15 \text{ K}$, $p_{amb} = 1013 \text{ hPa}$).

6.2.1 Sudden Change of the Cross-Sectional Area - Empirical Calculation

The same velocities as in Chapter 6.1 are used. A quadratic cross section is necessary for the calculation in FDS. A cross-sectional area ratio of 0.25 at a minimum velocity of 5 m/s should create enough pressure loss to compare results without any errors caused by low number of significant digits. The following pressure loss is calculated for a

velocity of 20 m/s using Equation 2.49. The ξ value is calculated with Equation 2.58. All results for the other flow velocities can be read from Table 6.7.

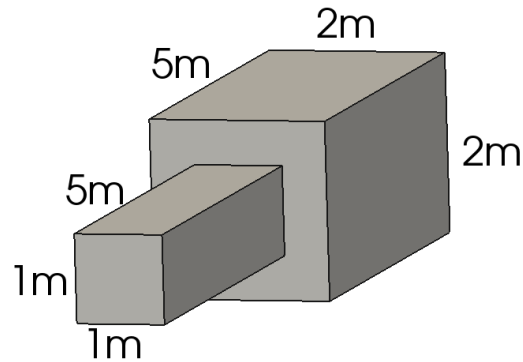


Figure 6.24: Sudden change of the cross-sectional area: Examined geometry

Calculation for $U = 20 \text{ m/s}$:

$$\xi = \left(1 - \frac{F_0}{F_2}\right)^2 = \left(1 - \frac{1 \text{ m}^2}{4 \text{ m}^2}\right)^2 = 0.75^2 = 0.5625$$

$$\Delta p_V = \xi \frac{\rho}{2} U^2 = 0.5625 \frac{1.188 \text{ kg/m}^3}{2} * (20 \text{ m/s})^2 = 133.65 \text{ Pa}$$

Because of the high Reynolds Number, the flow is certainly turbulent and the length of the lead-in area can be calculated with Equation 2.48:

$$10 \leq \frac{x_E}{D_{hydr}} \leq 60 \quad \text{with} \quad \frac{x_E}{D_{hydr}} = 50$$

$$x_E = 50 D_{hydr} = 50 * 1 \text{ m} = 50 \text{ m}$$

Results for all velocities :

Table 6.7: Sudden change of the cross-sectional area: Pressure losses, empirical

| $U [\text{m/s}]$ | ξ | $\Delta p_V [\text{Pa}]$ |
|------------------|--------|--------------------------|
| 20 | 0.5625 | 133.650 |
| 10 | 0.5625 | 33.413 |
| 5 | 0.5625 | 8.353 |

6.2.2 Sudden Change of the Cross-Sectional Area - NUMSTA3

The model in NUMSTA3 is built up from simple blocks (shown in Figure 6.25). An entrance (named inlet) is used to define the flow speed by defining the volume flux. Two tunnel elements (named tunnel1 and tunnel4) are used as a lead-in and lead-out length. The examined area is built up from tunnel2 and tunnel3. The cross section of tunnel2 is quadratic and has a side length of 1 m. Tunnel three has the same geometry but the side length of its cross section is 2 m. The cells inside the examined area have a length of 0.25 m while the length of the cells in the lead-in area and lead-out area have a length of 0.5 m to reduce the computing time. Bigger cells are necessary to produce a convergent solution. In the region near to the cross-section change, a too fine resolution of the mesh causes a solution that is fluctuating about ± 1 Pa. Such a fluctuation is too high, if small pressure losses are calculated as for the velocity of 5 m/s. The pipe drag coefficient λ must be defined manually for every velocity. The same values as in Chapter 6.1.1 are used for the tunnel elements 1 and 2 in the NUMSTA3 calculation. A different λ -value has been defined for the elements tunnel3 and tunnel4 because the flow slows down and the Reynolds Number changes. The second entrance block (named outlet) is used to enable outflow of the air. The velocity of the flow has the same value over the entire cross-section because NUMSTA3 is a one-dimensional CFD program. To write the pressure value of each cell in a file a lineprobe is used. A lineprobe records all physical quantities along a defined line in a CVS-file. The results for all velocities are given in Table 6.2. The solver is set to solve the equations for a steady system. The heat model for adiabatic systems is used. An approximate Riemann solver of Roe (first order upwind difference scheme) is used to simulate the flow inside the tunnel.

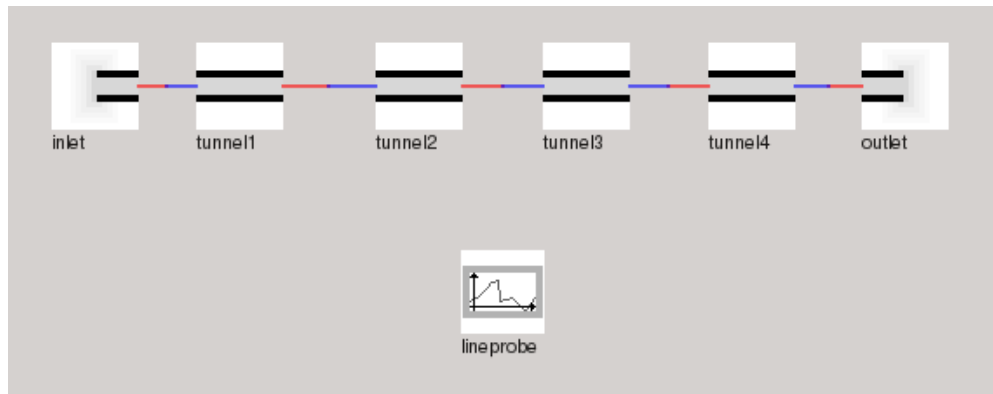


Figure 6.25: Sudden change of the cross-sectional area: model in NUMSTA3

The CSV-file that has been created by the lineprobe contains the values for p_{stat} and the velocity u . The pressure loss is the difference of the total pressure and must be calculated with Equation 2.52. This calculation is performed for the velocity of 20 m/s.

The index “0” stands for the state in the cell directly in front the cross-sectional area change and the index “2” for the cell directly behind it. The geodetic pressure is canceled because the entrance and the exit of the examined area are at the same height. All results are given in Table 6.8.

$$\rho g z_0 + p_0 + \frac{\rho}{2} u_0^2 = \rho g z_2 + p_2 + \frac{\rho}{2} u_2^2 + \Delta p_V$$

$$\Delta p_V = p_0 - p_2 + \frac{\rho}{2} (u_0^2 - u_2^2)$$

$$\Delta p_V = 101,240.006 \text{ Pa} - 101,330.292 \text{ Pa} + \frac{1.188 \frac{\text{kg}}{\text{m}^3}}{2} ((20.025 \text{ m/s})^2 - (5.002 \text{ m/s})^2)$$

$$\Delta p_V = 133.046 \text{ Pa}$$

Table 6.8: Sudden change of the cross-sectional area: Pressure losses, NUMSTA3

| $U [\text{m/s}]$ | $\Delta p_V [\text{Pa}]$ |
|------------------|--------------------------|
| 20 | 133.046 |
| 10 | 33.190 |
| 5 | 8.292 |

6.2.3 Sudden Change of the Cross-Sectional Area - OpenFOAM

Geometries like straight pipes are very simple and can be created very easily in Gmsh. However a change of the cross-sectional area is more complex and it is more difficult to create such a geometry in Gmsh. Therefore Blender is used to generate the surfaces of the geometry for enGrid (Figure 6.26). All surfaces must be built up from triangles to enable the mesh generation in enGrid as shown in Figure 6.27. Afterwards the Blender file is imported in enGrid, and a non-structured mesh with refined cells near walls can be generated. Figure 6.28 and 6.29 show the geometry of the examined area. All cells have a maximum side length of 8 cm. The disadvantage of this method is that a parametric mesh generation is not possible and changes take more time. To reduce the computing time, only a quarter of the geometry is simulated and two symmetry planes are used to get the same results as for the complete geometry. OpenFOAM offers a handy function called “checkMesh”. The function “checkMesh” allows calculating:

- max. aspect ratio
- max. skewness
- mesh non-orthogonally
- number of cells
- etc.

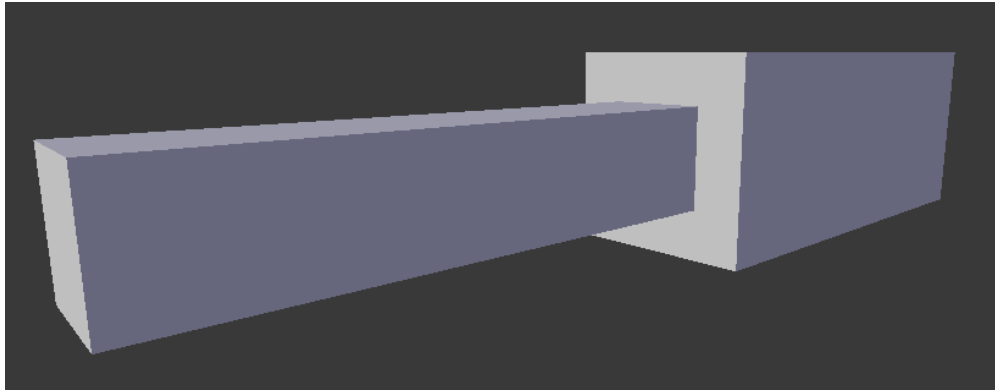


Figure 6.26: Sudden change of the cross-sectional area: surfaces in Blender

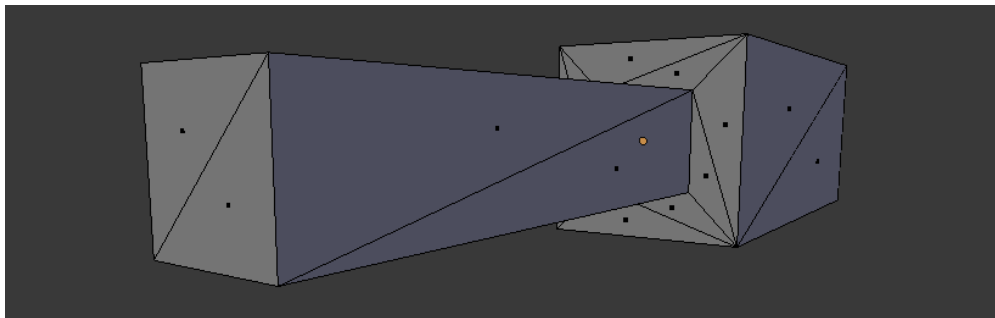


Figure 6.27: Sudden change of the cross-sectional area: triangled surfaces in Blender

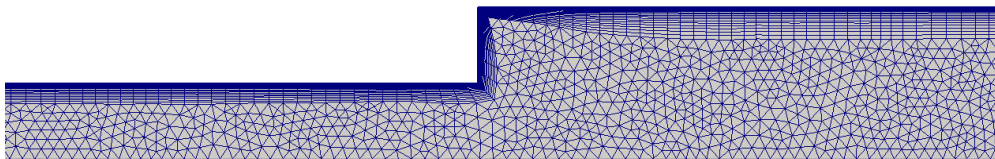


Figure 6.28: Sudden change of the cross-sectional area: unstructured mesh

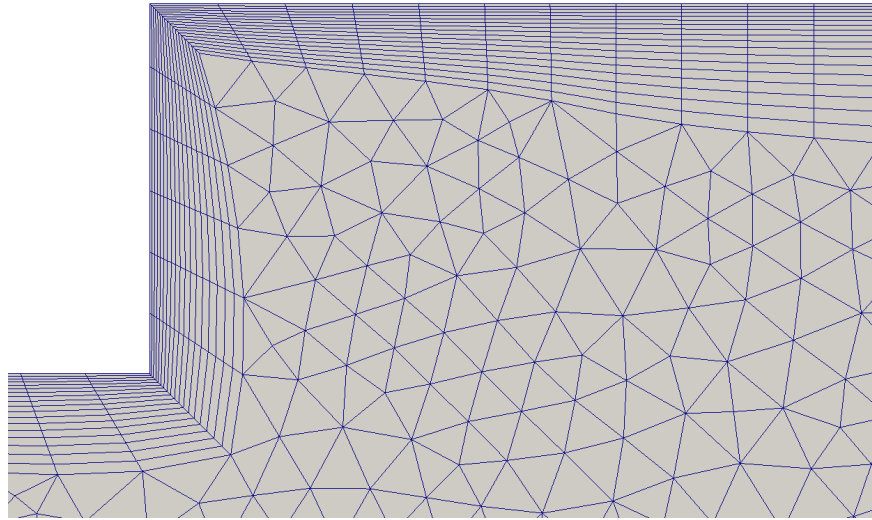


Figure 6.29: Sudden change of the cross-sectional area: refined cells in the OpenFOAM model

A function like this is very helpful, because it is not possible to see the quality of a non-structured mesh just by visual interpretation.

The necessary cell height of the cell next to the wall has been estimated in the range of y^+ 30 to 50. This has been proven to be the best magnitude in Chapter 6.1.6. Once the sudden change of the cross section has been created, the inlet and outlet surface of the grid is extruded to a length of 50 m. This action generates the lead-in and lead-out area of the geometry. Both, the lead-in and the lead-out area consist of a structured mesh, because the extrusion leads to this kind of mesh.

The used solver settings are:

- simpleFoam for a steady scheme
- RASModel $k-\omega$ -SST
- Gauss linear for gradSchemes
- Gauss linearUpwind Gauss for U divSchemes
- Gauss upwind for k divSchemes
- Gauss linear corrected for laplacianSchemes
- GAMG for the pressure
- smoothSolver for U, ω and k

The listed terms are no mathematical terms. They are predefined commands to select schemes in OpenFOAM. The meaning of each term can be read in the OpenFOAM User's Guide [20]. The detachment of the boundary layer is shown in Figure 6.30. As expected, zones with eddies and low velocities occur immediately after the cross-sectional change (compare with 2.7). The comparison between the flow profiles of the 1 m^2 and 4 m^2 cross-section is shown in Figure 6.31.

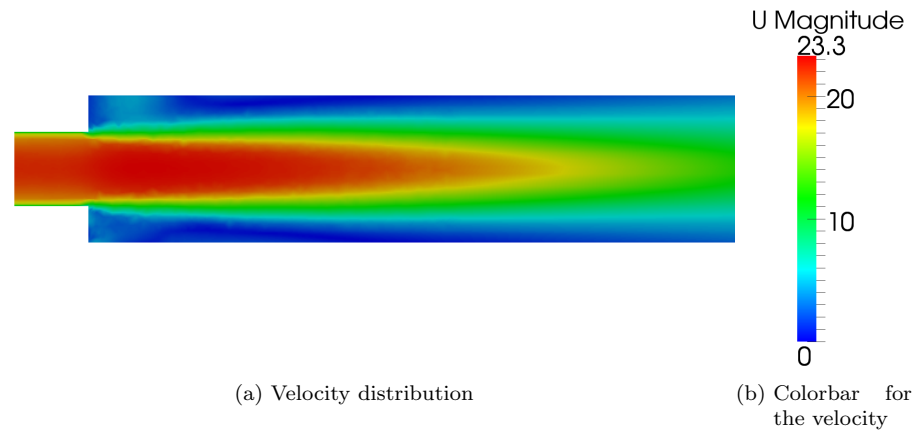


Figure 6.30: Sudden change of the cross-sectional area: Flow through the examined geometry for an inlet velocity of 20 m/s , OpenFOAM

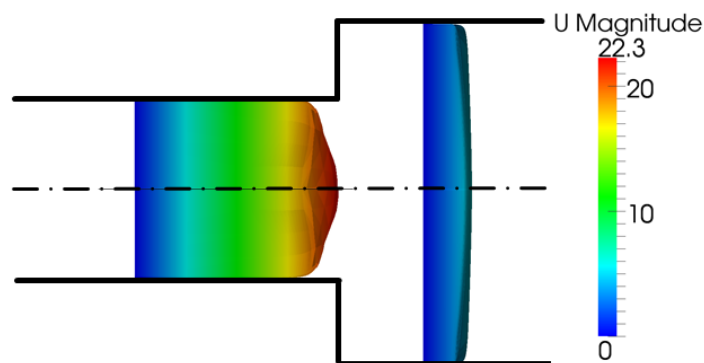


Figure 6.31: Sudden change of the cross-sectional area: Flow profile for an inlet velocity of 20 m/s , OpenFOAM

To determine the pressure loss of the channel, the distribution of the velocity and pressure along the longitudinal axis of the channel must be considered. The cross-

sectional change influences the flow over a certain length. After this length the flow is again nondisturbed. Both distributions are shown in Figure 6.32 and 6.33. These figures include the values of velocity and total pressure directly along the longitudinal axis of the geometry.

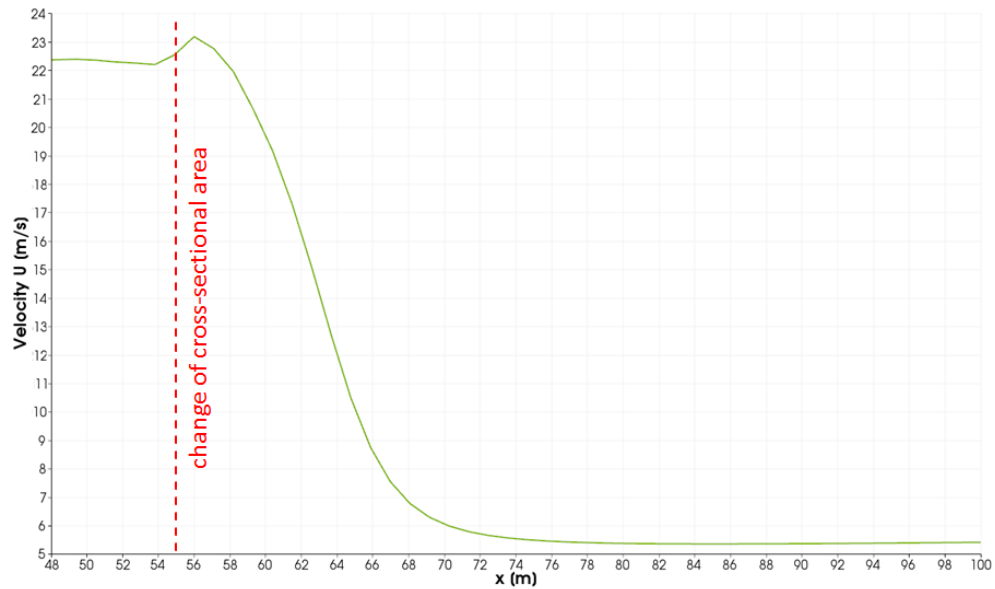


Figure 6.32: Sudden change of the cross-sectional area: Velocity trend for an inlet velocity of 20 m/s, OpenFOAM

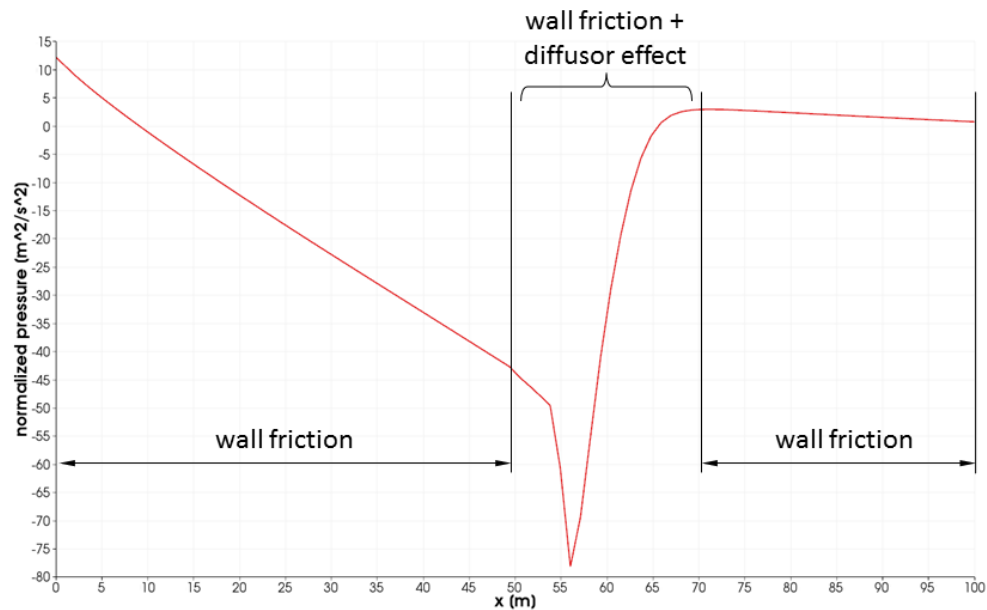


Figure 6.33: Sudden change of the cross-sectional area: Trend of static pressure for an inlet velocity of 20 m/s

All changes of velocity and pressure in the disturbed flow can be assigned to the cross-sectional change, with higher velocities occurring right after the cross-sectional change. These local high velocities are caused by the detachment of the boundary layer. The main flow is constricted by the boundary layer and forced to attain local over speed. An exact distribution of the disturbed and non-disturbed flow is not possible. The pressure along the first 50 m falls. This linear decrease is caused by the wall friction of the pipe. The same happens from about 70 m to the end of the geometry. All pressure losses between these two zones are caused by the cross-sectional change. The flow directly behind the cross-sectional change contains eddy zones, which prevent the contact of zones with high local velocities to the wall. Because of the low velocities next to the wall, the pressure loss caused by wall friction is low, but the mingling of the high velocity “jet” with the eddy zones and the circulation in the eddy zones also cause pressure losses. According to Idelchik [9], the pressure loss in the zone, where the flow is disturbed, can be treated like the pressure loss of a non-disturbed flow in a straight pipe.

The determination of the local pressure loss, caused by the sudden cross-sectional change, is difficult in OpenFOAM. Because of that, an empirical reference calculation is done. The reference calculation is similar to the calculation in Chapter 6.2.1. The only difference is that the pressure loss is calculated for the cross-sectional change plus two 20 m long pipes in front and behind the change (So it is a combination of the calculations in Chapter 6.1.1 and 6.2.1). The difference of the total pressures at

these locations is easier to compare with the results of OpenFOAM than the results for the local pressure losses. Unlike the total pressure, the static pressure is nearly constant over the entire cross-section. The total pressure varies over the cross-section, because the flow velocity has a certain profile. To get a representative value for the total pressure, the filter “**integrate variables**” is used in ParaView, which allows the calculation of area-averaged quantities:

$$\bar{p} = \frac{p_{int}}{F_{cross-section}} = \frac{\sum_{k=1}^N F_k p_{tot,k}}{F_{cross-section}} \text{ \textit{label eq : average}} \quad (6.7)$$

Tables 6.9 and shows the results of the empirical reference calculation. (The same simulation is done for NUMSTA3 to enable a direct comparison.)

Table 6.9: Sudden change of the cross-sectional area: Reference pressure losses, empirical

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 180.119 |
| 10 | 47.229 |
| 5 | 12.460 |

Table 6.10: Sudden change of the cross-sectional area: Reference pressure losses, NUMSTA3

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 178.680 |
| 10 | 46.800 |
| 5 | 12.381 |

These values for Δp_V can be compared to the total pressure differences in OpenFOAM. The results generated by OpenFOAM can be read in Table 6.11.

Table 6.11: Sudden change of the cross-sectional area: Pressure losses, OpenFOAM

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 188.657 |
| 10 | 49.016 |
| 5 | 12.773 |

6.2.4 Sudden Change of the Cross-Sectional Area - FDS

Blender is used again to create the geometry for FDS as shown in Figure 6.26 in Chapter 6.2.3. The cells of the examined area are 0.05 m long, 0.05 m high and 0.05 m

wide. The mesh is not refined near the walls because of the long computing duration of a simulation with a too fine mesh. Because of the unsteady simulation, a duration of 120 s is used to achieve a quasi-steady flow in the straight parts in front and behind the cross-sectional change. Splitting the computing domain in more than one mesh allows to define the geometry and enables a parallelization of the simulation. Vents are placed at the inlet and outlet surfaces to define boundary conditions. In Figure 6.35 the distribution of velocity (for an inlet velocity of 20 m/s) can be seen. One striking phenomenon is the occurrence of time dependent eddies. A command named VECTOR is set to .TRUE., to enable a vectorized depiction of eddies in the eddy zones. Several &DEVC blocks are used to define devices to record velocity and static pressure values over the cross section. Figure 6.36 shows the unsteady eddies at different time steps. Appendix A.2 includes the FDS input file for this simulation case. Figure 6.34 shows the grid of the FDS mesh. The total pressure is calculated in the cross sections

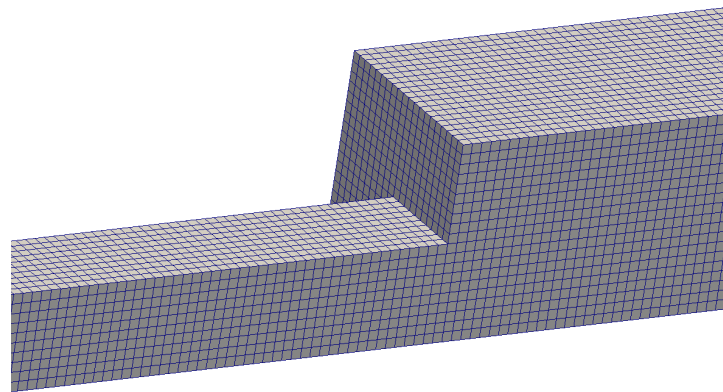


Figure 6.34: Sudden change of the cross-sectional area: FDS grid

20 m in front and behind the cross-sectional change and compared with the empirical results in Table 6.9. The reason for the selected location (20 m in front and behind the cross-sectional change) is the same as in Chapter 6.2.3. High fluctuations are occurring because of the unsteady behavior and the disturbed flow 20 m behind the cross-sectional change. The total pressure at each location a device has been placed and the temporal mean value over the last 20 s of the total pressure at each location are calculated (from second 100 to 120). After that, the temporal mean values are averaged over the cross section. The result is a value for the total pressure that is averaged in space and time. All pressure losses calculated in FDS can be read from Table 6.12.

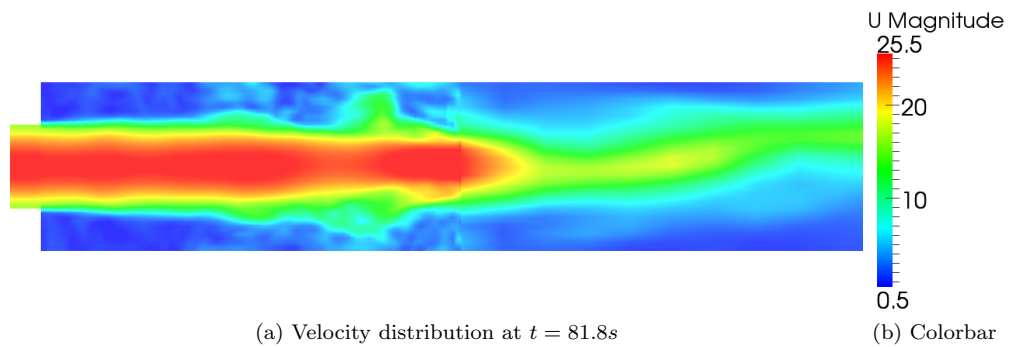


Figure 6.35: Sudden change of the cross-sectional area: Velocity distribution for an inlet velocity of 20 m/s , FDS

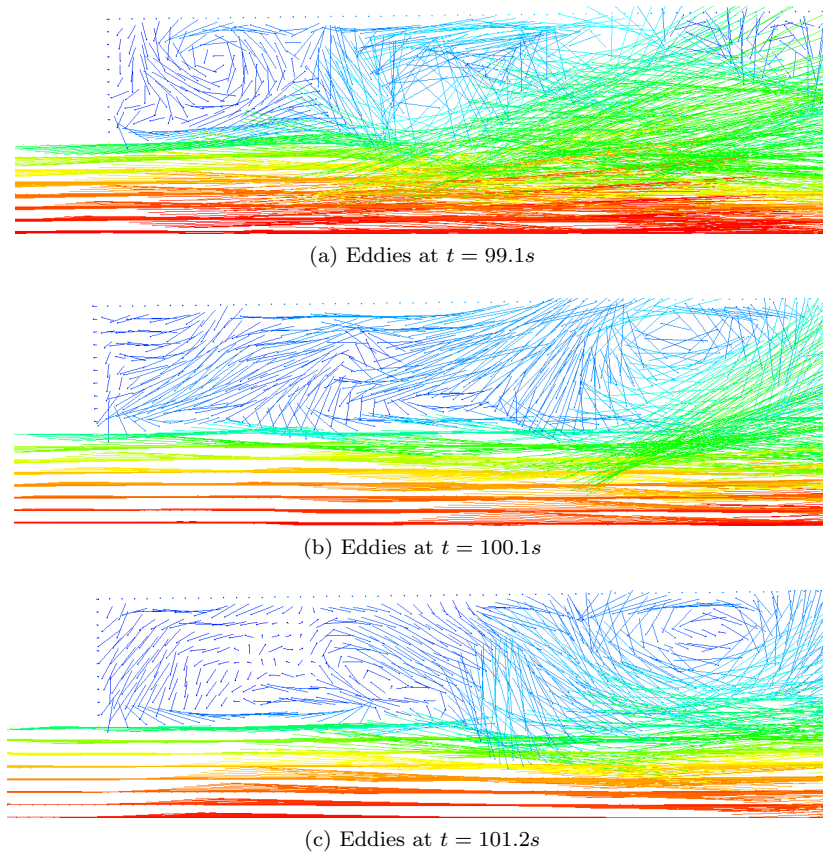


Figure 6.36: Sudden change of the cross-sectional area: Eddies for an inlet velocity of 20 m/s , FDS

Table 6.12: Sudden change of the cross-sectional area: Pressure losses, FDS

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 223.767 |
| 10 | 59.120 |
| 5 | 15.699 |

6.2.5 Sudden Change of the Cross-Sectional Area - Comparison of Results

The results of all used software are summarized in Table 6.13. The high deviation in the results provided from FDS is remarkable.

Table 6.13: Sudden change of the cross-sectional area: : summarized results

| U [m/s] | Δp [Pa] empirical | Error[%] NUMSTA3 | Error[%] OpenFOAM | Error[%] FDS |
|-----------|------------------------------|---------------------|----------------------|-----------------|
| 20 | 180.190 | -0.80 | 4.74 | 24.18 |
| 10 | 47.229 | -0.91 | 3.78 | 25.18 |
| 5 | 12.460 | -0.63 | 2.51 | 26.00 |

FDS has its strengths and weaknesses. It doesn't generate very accurate results, but it is dependable for the visualization of transport processes. When comparing Figure 6.30 and Figure 6.35, a big difference can be seen. The $k-\omega$ -SST model with a steady solver in OpenFOAM smooths the entire flow field, while the LES in FDS produces unsteady results and enables the visualization of transport processes. (An unsteady large eddy simulation is also possible in OpenFOAM, but requires more computing performance than RANS methods.)

6.2.6 Sudden Change of the Cross-Sectional Area - Reasons for Differing Results

NUMSTA3 provides results, which are close to the empirical solution. Therefore a detailed discussion of the reasons for these deviations is unnecessary.

OpenFOAM provides results with moderate deviations. The reason for the error is the constant cell height along the mesh. The height of the cell is constant, while the velocity in the geometry changes. This leads to different y^+ values, which are not all in the ideal range of 50 to 300, and thereby cause an error.

FDS The error caused by high velocities at walls does not occur in this case. The reason for this is the detachment of the boundary layer. The detachment of the boundary layer causes low velocities at the walls. The part of the pressure loss caused by wall

friction compared to the entire pressure loss is low and therefore the error caused by the “wrong” slip boundary condition is small.

6.3 Flow in an Elbow

Another simple case is a change of the flow direction. The physics behind the flow in a bow has already been described in Chapter 2.5.4. Once more the flow is simulated for three different velocities and a simple geometry. All calculations are done under the same initial conditions as in Chapter 6.1 ($T = 293.15\text{ K}$, $p_{amb} = 1013\text{ hPa}$)

6.3.1 Flow in an Elbow - Empirical Calculation

The same velocities as in Chapter 6.1 are used. A quadratic profile is necessary for the calculation in FDS. The angle of direction change is 90 degrees. Figure 6.37 shows the examined geometry. The pressure loss is calculated for a velocity of 20 m/s with Equation 2.49. ξ_{bow} is calculated with Equation 2.72, where A , C_1 and ξ_M can be read from tables in Idelchik I.E. Handbook of Hydraulic Resistance [9]. Two 20 m long pipes are used in front and behind the change of flow direction to enable a comparison between the results of all software. All results for further velocities can be read from Table 6.7.

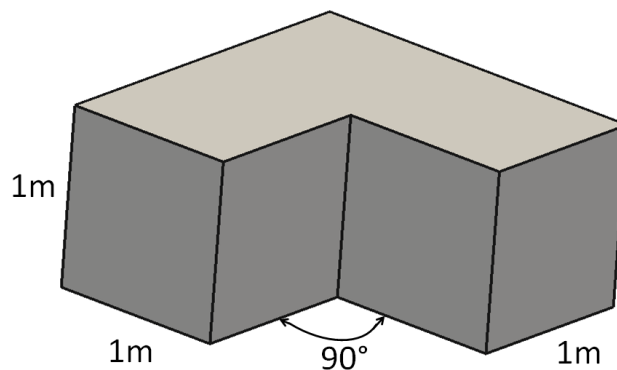


Figure 6.37: Flow in an Elbow: Examined geometry

Calculation for $U = 20 \text{ m/s}$:

$$\begin{aligned}\xi_{bow} &= C_1 A \xi_M \\ C_1 \left(\frac{a_0}{b_0} = 1 \right) &= 1 \\ A(\delta = 90^\circ) &= 1.2 \\ \xi_M(\delta = 90^\circ) &= 0.99 \\ \xi_{bow} &= 1 * 1.2 * 0.99 = 1.188 \\ D_{hydr} &= 4 * \frac{A}{U} = 4 * \frac{1 \text{ m} * 1 \text{ m}}{4 * 1 \text{ m}} = 1 \text{ m} \\ Re &= \frac{U D_{hydr}}{\nu} = \frac{20 \text{ m/s} * 1 \text{ m}}{15.35 * 10^{-6} \text{ m}^2/\text{s}} = 1,302,931 \\ \lambda &= 0.3164 * Re^{-\frac{1}{4}} = 0.3164 * 1,302,931^{-\frac{1}{4}} = 0.009365 \\ \xi &= \lambda \frac{L}{D_{hydr}} = 0.009365 \frac{20 \text{ m}}{1 \text{ m}} = 0.1873 \\ \xi_{cum} &= \xi_{bow} + 2\xi = 1.188 + 2 * 0.1873 = 1.5626 \\ \Delta p_V &= \xi_{cum} \frac{\rho}{2} U^2 = 1.5626 \frac{1.188 \text{ kg/m}^3}{2} * (20 \text{ m/s})^2 = 371.274 \text{ Pa}\end{aligned}$$

Because of the high Reynolds Number, the flow is certainly turbulent and the length of the lead-in area can be calculated with Equation 2.48:

$$\begin{aligned}10 \leq \frac{x_E}{D_{hydr}} \leq 60 \quad \text{with} \quad \frac{x_E}{D_{hydr}} &= 50 \\ x_E = 50 D_{hydr} &= 50 * 1 \text{ m} = 50 \text{ m}\end{aligned}$$

Results for all velocities :

Table 6.14: Flow in an Elbow: Pressure losses, empirical

| $U [\text{m/s}]$ | ξ_{cum} | $\Delta p_V [\text{Pa}]$ |
|------------------|-------------|--------------------------|
| 20 | 1.563 | 371.274 |
| 10 | 1.633 | 97.028 |
| 5 | 1.718 | 25.509 |

6.3.2 Flow in an Elbow - NUMSTA3

The model in NUMSTA3 is built up from simple blocks (pictured in Figure 6.38). An entrance (named inlet) is used to define the flow speed by defining the volume flux. Two tunnel elements (named tunnel1 and tunnel4) are used as a lead-in and lead-out length. Both tunnel elements are 50 m long. The 20 m long tunnel elements in the front end behind the bow are named tunnel2 and tunnel3. The quadratic cross-section

with a side length of 1 m is constant along the entire tunnel. All cells have a length of 0.5 m. A junction element named bow is used to define a change of flow direction. The angle of the junction element is defined by the ξ -value. ξ has the same value for all three velocities and has already been calculated in Chapter 6.3.1. The pipe drag coefficient λ is set to the same values as in Chapter 6.1.1. The second entrance block (named outlet) is used to enable outflow of the air. The velocity of the flow has the same value over the entire cross-section, because NUMSTA3 is a one-dimensional CFD program. The solver is set to solve the equations for a steady system. The heat model for adiabatic systems is used. An approximate Riemann solver of Roe is used to simulate the flow inside the tunnel. The convective terms are discretized by a first order upwind difference scheme.

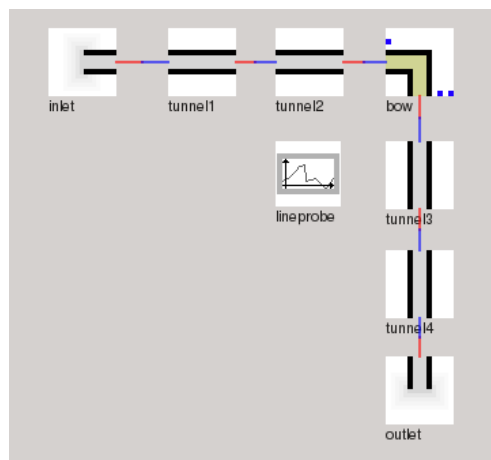


Figure 6.38: Flow in an Elbow: Model in NUMSTA3

The Lineprobe in the NUMSTA3 model writes the values of all physical quantities of each cell into a CSV file. This CSV file contains the values for p_{stat} and the velocity u . The values of the static pressures are read from the CSV file for the cells 20 m in front and behind the change of flow direction. Due to the constant cross section, Δp_{stat} is identical to Δp_{tot} and consequently identical to the pressure loss. The results of the NUMSTA3 simulation can be read from Table 6.15.

Table 6.15: Flow in an Elbow: Pressure losses, NUMSTA3

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 373.509 |
| 10 | 96.983 |
| 5 | 25.424 |

6.3.3 Flow in an Elbow - OpenFOAM

Geometries like a bow are too complex to be created quickly and efficiently in Gmesh. Because of that, Blender is used again to create the surface of the investigated geometry. All surfaces are triangulated (Figure 6.39) to enable the mesh generation in enGrid. Only one half of the geometry is simulated, and a symmetry plane is used to gain the same results as by a simulation of the complete geometry (Figure 6.40). The created mesh is unstructured with a maximum cell size length of 8 cm and is refined at the walls to enable the calculation of the boundary layer. OpenFOAM's function "checkMesh" is used to control if the generated mesh is "OK" (in terms of max. aspect ratio, max. skewness, etc.). The maximum cell height near walls has been estimated in a range of y^+ 30 to 50.

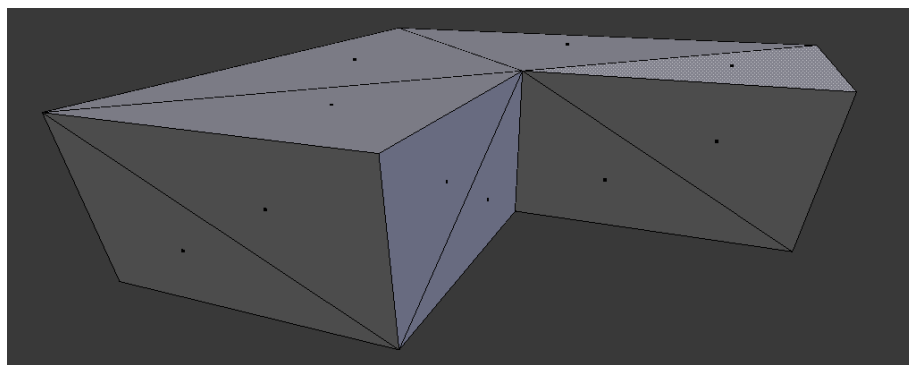


Figure 6.39: Flow in an Elbow: Triangulated surfaces in Blender

Once the generation of the examined geometry is finished, the lead-in and lead-out length can be created by extruding inlet and outlet surfaces to a length of 50 m.

The used solver settings are:

- simpleFoam for a steady scheme
- RASModel $k-\omega$ -SST
- Gauss linear for gradSchemes
- Gauss linearUpwind Gauss for U divSchemes
- Gauss upwind for k divSchemes
- Gauss linear corrected for laplacianSchemes
- GAMG for the pressure
- smoothSolver for U, ω and k

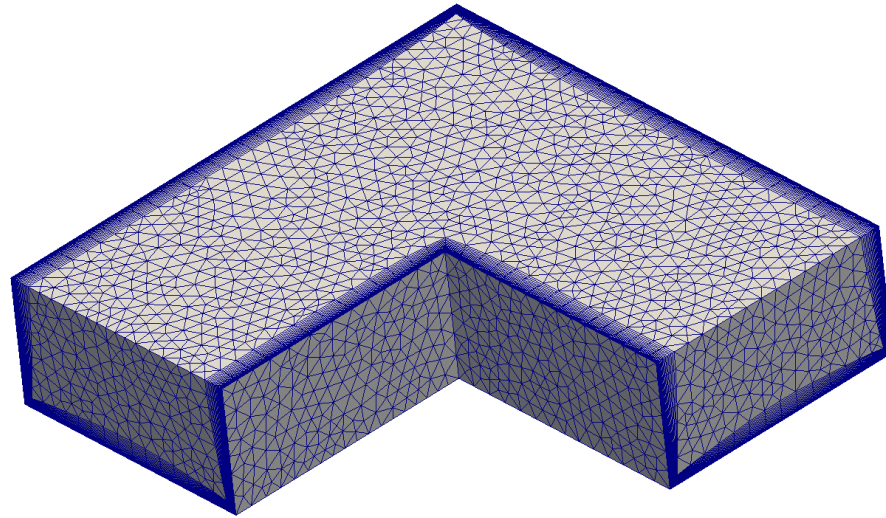


Figure 6.40: Flow in an Elbow: Unstructured mesh

The listed terms are no mathematical terms. They are predefined commands to select schemes in OpenFOAM. The meaning of each term can be read in the OpenFOAM User's Guide [20]. All changes of velocity and pressure in the non-developed flow are caused by change of flow direction and friction. An exact separation of the disturbed and non-disturbed flow is not possible. In the two straight pipes in front and behind the elbow, a linear pressure decrease is caused by the wall. All pressure losses between these two zones are caused by the change of flow direction and wall friction. Figure 6.41 shows the velocity distribution for the examined geometry calculated in OpenFOAM. Centrifugal forces appear, and the total pressure rises on the outer wall of the bow and decreases on the inner wall (look at Figure 6.42). The detachment of the flow blocks parts of the cross-section, causing a bell mouth effect and forces the flow to local over speed. After the bow passage the total pressure rises on the inner wall and sinks on the outer wall. The same process has been described by I.E. Idelchik and is summarized in Chapter 2.5.4. In a comparison of Figure 6.41 and 2.9, it can be seen that these two figures show the same flow. Table 6.16 shows the pressure losses between 20 m in front and behind the change of flow direction.

Table 6.16: Flow in an Elbow: Pressure losses, OpenFOAM

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 489.701 |
| 10 | 112.382 |
| 5 | 27.113 |

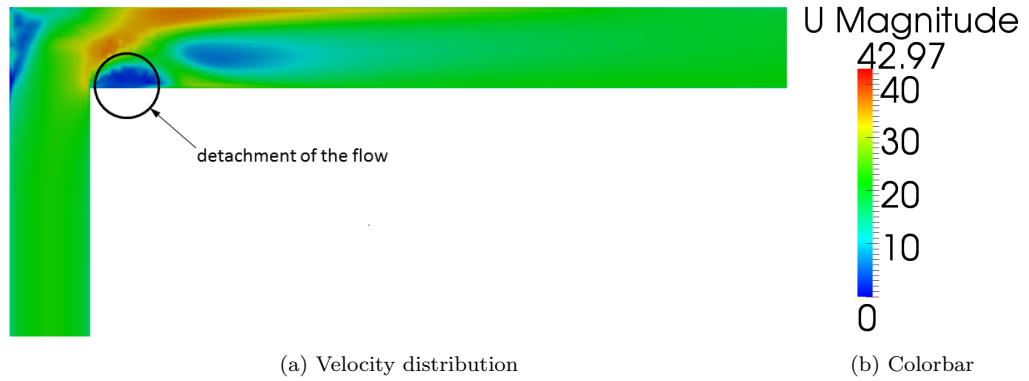


Figure 6.41: Flow in an Elbow: Velocity distribution for an inlet velocity of 20 m/s, OpenFOAM

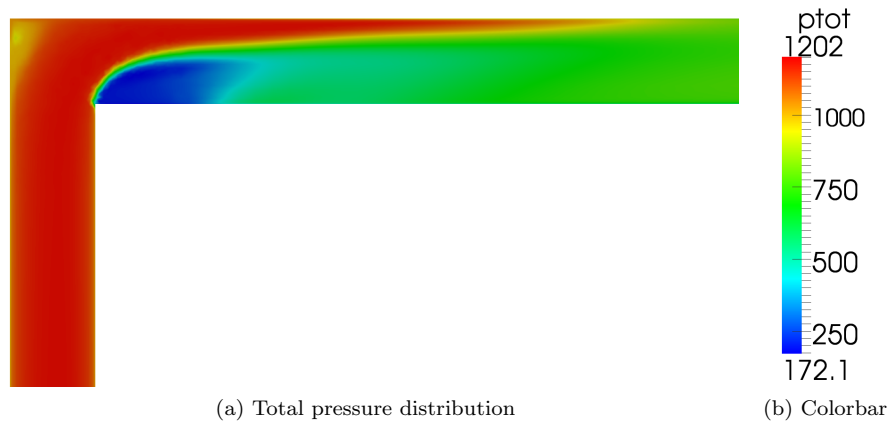


Figure 6.42: Flow in an Elbow: Total pressure distribution in OpenFOAM for an inlet velocity of 20 m/s

6.3.4 Flow in an Elbow - FDS

Again Blender is used to create the geometry for FDS. The geometry is identical to the geometries used in NUMSTA3 and OpenFOAM. The cross-section is quadratic and has a side length of 1 m. A 20 m long quadratic pipe is placed in front and behind the bow. A lead-in/lead-out length of 50 m is used. The geometry is pictured in Figure 6.43. Figure 6.44 shows the grid, which is used in FDS. The cells of the lead-in/lead-out area are 0.2 m long, 0.05 m high and 0.05 m wide. All cells in the “bow” are cubic and have a side length of 0.05 m. The mesh is not refined near the walls because of the long computing duration of a simulation with a mesh that is too fine. Because of the unsteady simulation, a duration of 120 s is used to create a quasi-steady flow in the straight parts in front and behind the change of flow direction. Splitting the computing domain into more than one mesh allows to define the geometry and enables a parallelization of the simulation. Vents are placed at the inlet and outlet surfaces to define boundary conditions. Appendix A.3 includes the FDS input file.

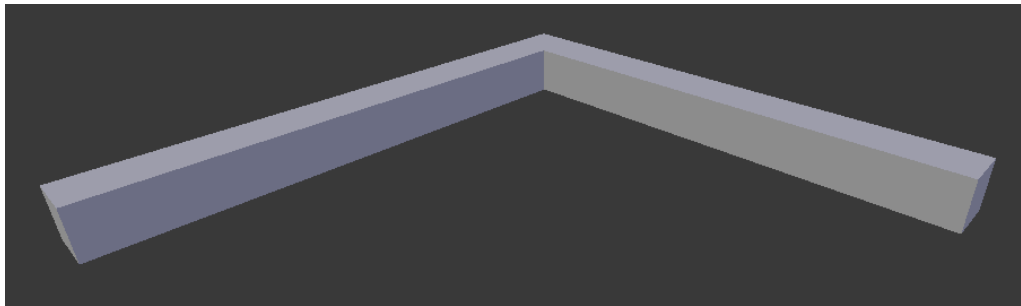


Figure 6.43: Flow in an Elbow: geometry in Blender

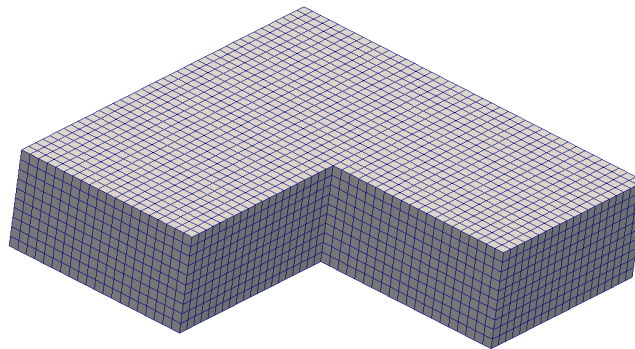


Figure 6.44: Flow in an Elbow: Grid in FDS

Table 6.17: Flow in an Elbow: Pressure losses, FDS

| U [m/s] | Δp_V [Pa] |
|-----------|-------------------|
| 20 | 626.380 |
| 10 | 166.730 |
| 5 | 44.898 |

6.3.5 Flow in an Elbow - Comparison of Results

The results of all used software are given in Table 6.18. Again, the deviation in the results provided from FDS is remarkable.

Table 6.18: Flow in an Elbow: Summarized results

| U [m/s] | Δp [Pa] empirical | Error[%] NUMSTA3 | Error[%] OpenFOAM | Error[%] FDS |
|-----------|------------------------------|---------------------|----------------------|-----------------|
| 20 | 371.274 | +0.60 | +31.90 | +68.71 |
| 10 | 97.028 | -0.05 | +15.82 | +71.84 |
| 5 | 25.509 | +0.33 | +6.37 | +76.00 |

6.3.6 Flow in an Elbow - Reasons for Differing Results

NUMSTA3 provides results, which are close to the empirical solution. Therefore, a detailed discussion of the reasons for these deviations is unnecessary.

OpenFOAM provides results with moderate deviations for low velocities. For the inlet velocity of 20 m/s the error is very high. The refinement of the boundary layer should be good enough, because the y^+ -values are about 55. The reason for the deviations at high velocities could be the relative coarse grid with a cell side length of 8 cm. Therefore, the mesh is refined and a side length of 4 cm is used. The simulation is redone for an inlet velocity of 20 m/s, because the highest error occurred at this input parameter. The result of the simulation with the refined grid can be read from Table 6.19:

Table 6.19: Flow in an Elbow: Results for refined mesh, OpenFOAM

| U [m/s] | Δp [Pa] empirical | Δp [Pa] OpenFOAM | Error[%] OpenFOAM |
|-----------|------------------------------|-----------------------------|----------------------|
| 20 | 371.274 | 439.879 | +18.48 |

The magnitude of the error became smaller, but is still too high to be tolerated. A possibility to reduce the error even more would be a further refinement of the cell size. This would cause a high amount of cells and in the case of complex geometries very

high computing performance, which is not very practical in industry. Hence, another solution must be found. A second approach is based on a thread in “CFD Online” [28]. OpenFOAM often has problems to generate accurate and convergent solutions, if a tetrahedral mesh is used in areas where a detachment of the flow takes place. enGrid offers the option to export a polyhedral mesh. In this process enGrid replaces all tetrahedral cells by polyhedral cells. The generation of a polyhedral mesh can be found in literature under the name **dual mesh**. Basis is a tetrahedral mesh as shown in Figure 6.45a. In a first step points are created in the center of gravity of all tetrahedrons (Figure 6.45b). In a second step points are created in the center of each cell edge (Figure 6.45c). In a last step the new points are connected like shown in Figure 6.45d and the old points are deleted. The result is a polyhedral mesh like shown in Figure 6.45e.

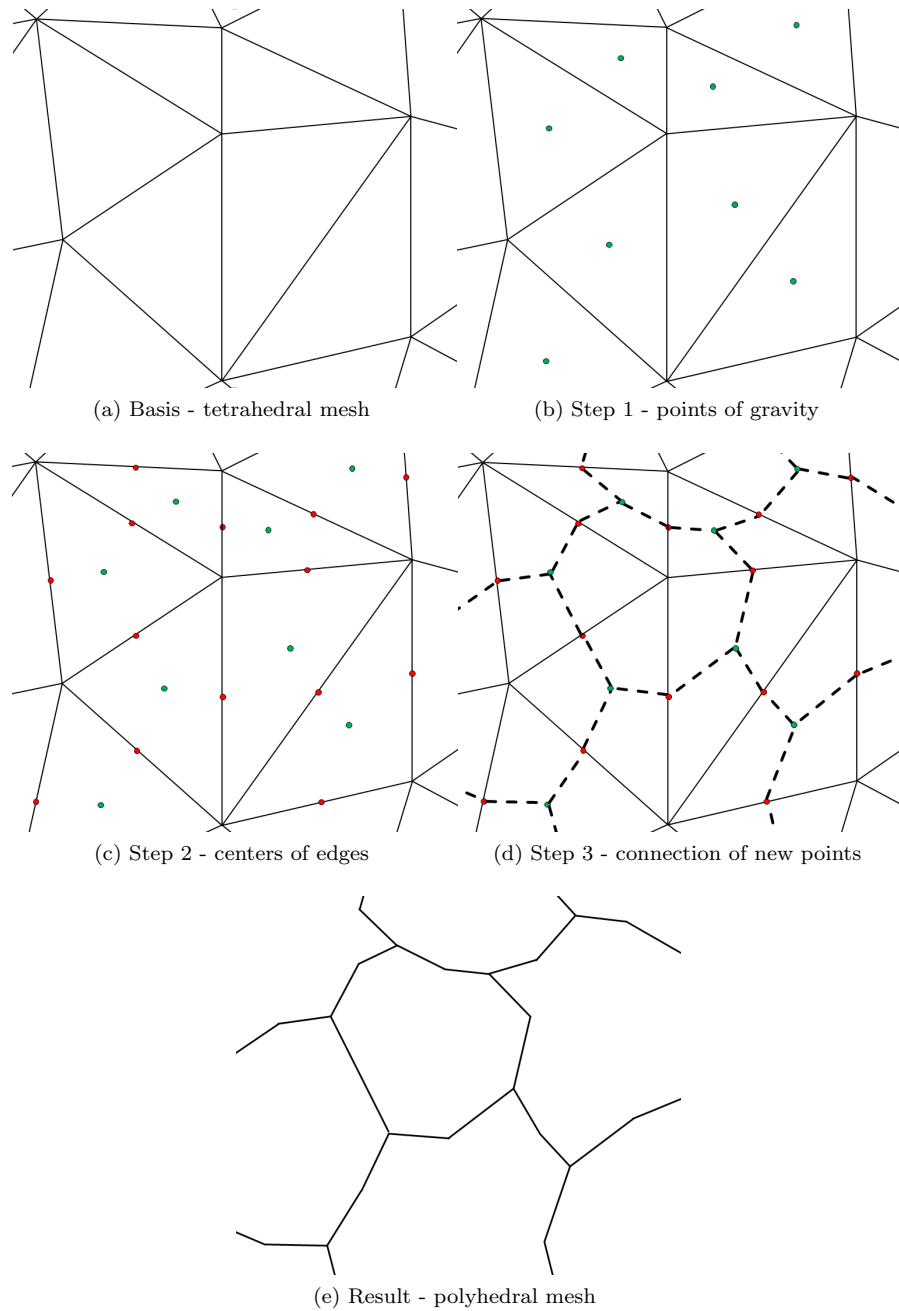


Figure 6.45: Generation of a polyhedral mesh

In another simulation the generated tetrahedral cells still have a side length of 4 cm, but they are replaced by polyhedral cells. This process creates cells with more cell surfaces, but reduces the amount of cells outside the boundary layer to about a third of the amount of cells in the tetrahedral grid. Figures 6.45a and 6.45e show that the grid does not get much coarser, although the amount of cells has been reduced. Thus, the necessary computing performance is reduced and OpenFOAM should not have any problems with the detachment of the boundary layer. The result of the simulation with the polyhedral grid can be read from Table 6.20:

Table 6.20: Flow in an Elbow: Results for polyhedral mesh, OpenFOAM

| U [m/s] | Δp [Pa] empirical | Δp [Pa] OpenFOAM | Error [%] OpenFOAM |
|-----------|------------------------------|-----------------------------|-----------------------|
| 20 | 371.274 | 371.063 | -0.06 |

A Reason for the less accurate results when using a tetrahedral mesh is the use of a upwind scheme. This will be explained with the help of a short two-dimensional example. Figure 6.46 shows two tetrahedral cells and isolines for a scalar named φ . The figure shows that the values for φ in blue the point and in the red point are the same. The velocity u is assumed to be constant.

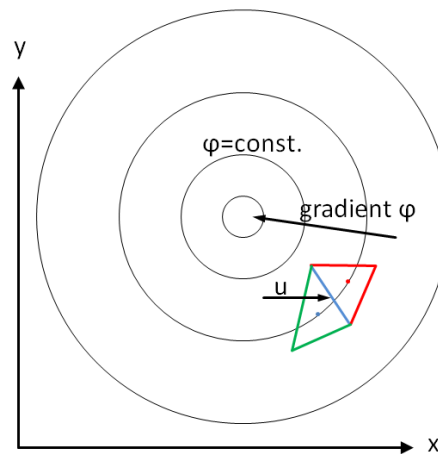


Figure 6.46: Flow in an Elbow: Sketch of a tetrahedral grid

Figure 6.47 shows a more detailed sketch of the red cell. A general formulation for the transport of φ can be written:

$$\int_V \frac{\partial \varphi}{\partial t} dV + u * \int_O \varphi \vec{n} * dO = 0 \tag{6.8}$$

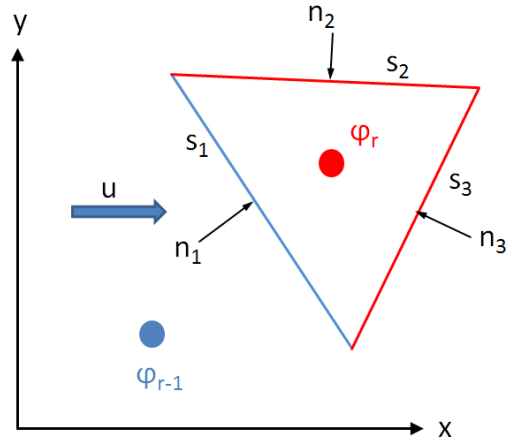


Figure 6.47: Flow in an Elbow: Sketch of a upwind scheme

The second term in Equation 6.8 can be re-arranged with the help of the Gauss's theorem [4]:

$$\int_O \varphi * dO = \int_V \vec{\nabla} \varphi * dV \quad (6.9)$$

Since the cell pictured in Figure 6.47 has a finite number of edges, the left side of Equation 6.9 can be written as a sum, where k is the number of cell edges:

$$\int_O \varphi * dO = \sum_{i=1}^k \varphi_i * \vec{n}_i * s_i \quad (6.10)$$

The value for φ at the boundaries of the cell are according to a first order upwind scheme φ_{r-1} on the blue border and φ_r on the red borders. Therefore Equation 6.10 can also be written as:

$$\int_O \varphi * dO = \varphi_{r-1} * \vec{n}_1 * s_1 + \varphi_r * \vec{n}_2 * s_2 + \varphi_r * \vec{n}_3 * s_3 \quad (6.11)$$

The value φ is constant over the volume of a single cell. Therefore the right side of Equation 6.9 can be written as:

$$\int_V \vec{\nabla} \varphi * dV = \vec{\nabla} \varphi * V \quad (6.12)$$

After all these formulations Equation 6.9 can be re-arranged to:

$$\vec{\nabla} \varphi = \frac{\varphi_{r-1} * \vec{n}_1 * s_1 + \varphi_r * \vec{n}_2 * s_2 + \varphi_r * \vec{n}_3 * s_3}{V} \quad (6.13)$$

Equation 6.13 is an equation in vector form and in this example it can be split up in two equations:

$$\frac{\partial \varphi}{\partial x} = \frac{\varphi_{r-1} * n_{x1} * s_1 - \varphi_r * n_{x2} * s_2 - \varphi_r * n_{x3} * s_3}{V}$$

$$\frac{\partial \varphi}{\partial y} = \frac{\varphi_{r-1} * n_{y1} * s_1 - \varphi_r * n_{y2} * s_2 - \varphi_r * n_{y3} * s_3}{V}$$

According to Figure 6.46 $\varphi_{r-1} = \varphi_r$ and the x-component of Equation 6.13 can be written as:

$$\frac{\partial \varphi}{\partial x} = \frac{\varphi_r * (n_{x1} * s_1 - n_{x2} * s_2 - n_{x3} * s_3)}{V} \quad (6.14)$$

For all closed line loops is:

$$\sum \vec{n}_i * s_i = 0$$

Therefore, the gradient of φ in x-direction would be zero. But Figure 6.46 shows that the gradient at this point is not zero. This problem occurs, because the tetrahedral cell (Figure 6.48a) gets its information only over one edge. If the flow would enter the cells at two different cell edges, this error would not occur and the calculated gradient would not be zero. A polyhedral cell (Figure 6.48b) has more cell edges and therefore is less sensible to errors caused by a unfavorable inflow direction.

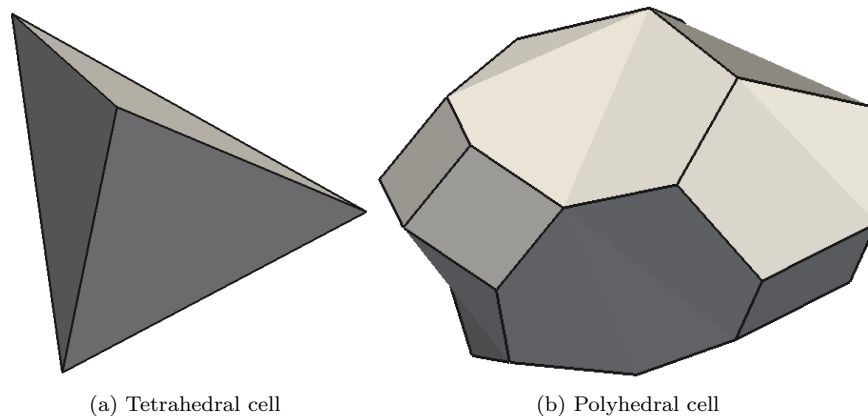


Figure 6.48: Cells of a numeric grid

FDS doesn't generate reliable results. The reason for that is the high velocities at the walls. The bend of the flow causes a detachment of the boundary layer on one side of the duct. The air is forced to flow faster on the other side to fulfill the conservation of mass. The high velocities at the wall in combination with the "wrong" slip boundary condition leads to wrong velocity gradients at the wall and therefore to a wrong pressure loss caused by wall friction. In addition to that, the cells at the walls are too coarse to simulate the detachment of the boundary layer accurately, which leads to wrong velocity profiles and again to wrong pressure losses.

An additional simulation with a refined mesh is used to determine, if the relatively high errors are caused by the coarse grid. The same bow as in Chapter 6.3.4 for an inlet velocity of 20 m/s is simulated, but all cell lengths are divided by two. Therefore, the amount of cells rises by a factor of 8 and the necessary computer performance is higher. Table 6.21 gives the results of the simulation:

Table 6.21: Flow in an Elbow: Results for polyhedral mesh, OpenFOAM

| U [m/s] | Δp [Pa] empirical | Δp [Pa] FDS | Error [%] FDS |
|-----------|------------------------------|------------------------|------------------|
| 20 | 371.274 | 613.738 | 65.31 |

The simulation shows that the simulated pressure loss got more accurate, but still is not very reliable. In an other simulation all cell lengths are divided by two again. This should show, if the simulation could get more accurate for a grid that is just fine enough. The amount of cells rises again by a factor of 8 and the necessary computer performance gets even higher. After a few seconds the simulation stopped, because of insufficient computer performance. To determine effects caused by the boundary layer, the resolution of the grid near walls must be very high. FDS provides no function to refine the grid for the boundary layer. The only possibility to resolve the boundary layer is to refine the entire mesh. This causes many cells and also a high computer performance. Hence, FDS is not very practical to determine pressure losses.

6.4 Heat Source

The next test case is a tunnel flow with a heat source. An analytical or empirical calculation of a three-dimensional geometry including a heat source is not possible. Therefore, no analytical or empirical calculation will be done. The data of a fire test is used for comparison with the simulation results.

6.4.1 Memorial Tunnel Fire Ventilation Test Program [30]

The Memorial Tunnel is located in the USA near Charleston, West Virginia. This is a two-lane 854 m long, mountain tunnel having a 3.2% incline from south to north. The construction of the Memorial Tunnel started in 1952, and the tunnel officially opened to traffic on November 8, 1954. 30 years later the Memorial Tunnel was abandoned. Between 1992 and 1995, the tunnel was utilized by the Department of Transportation for smoke, fire and ventilation experiments. The cross-section of the tunnel is pictured in Figure 6.49.

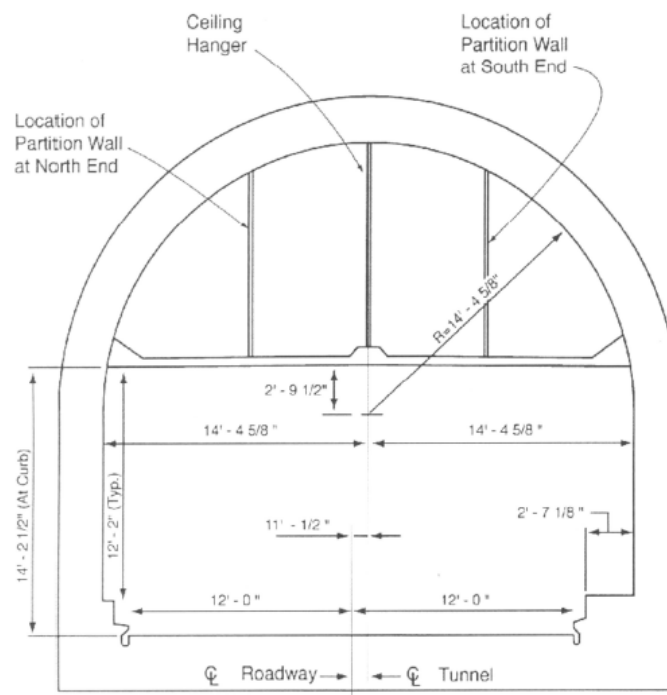


Figure 6.49: Heat Source: Cross-section of Memorial Tunnel [30]

The Memorial Tunnel Fire Ventilation Test Program (MTFVTP) consisted of a se-

ries of full-scale fire tests. Various tunnel ventilation systems and configurations of such systems were operated to evaluate their respective smoke and temperature management capabilities. In preparation for the MTFVTP, the tunnel was modified and instrumented to allow operation and evaluation with the following ventilation system configurations:

- Full Transverse Ventilation
- Partial Transverse Ventilation
- Partial Transverse Ventilation with Single Point Extraction
- Partial Transverse Ventilation with Oversized Exhaust Ports
- Point Supply and Point Exhaust Operation
- Natural Ventilation
- Longitudinal Ventilation with Jet Fans

Full Transverse Ventilation:

At the first 150 m on both sides of the tunnel all openings in the exhaust and supply ducts were closed. In the middle part of the tunnel the openings of the supply and exhaust duct were balanced to enable uniform exhaust/supply of the tunnel. A fire was started in the middle of the tunnel and the exhaust/supply fans started to operate. This configuration caused a flow in the middle of the tunnel, which was normal to the tunnel longitudinal axis.

Partial Transverse Ventilation:

The separating wall between the supply and exhaust duct (6.49) was removed so that the both ducts can operate as one single exhaust ducts. All openings of this exhaust duct were balanced to provide uniform exhaust of the entire tunnel. A fire was started in the middle of the tunnel and the exhaust fans started to operate. The supply duct entered the tunnel at the north and south portal.

Partial Transverse Ventilation with Single Point Extraction:

The same settings as for the Partial Ventilation tests were used for this sequence. In addition, large normally closed openings in the ceiling were utilized. One single point extra opening was placed south of the fire and three were placed in the north of it.

Partial Transverse Ventilation with Oversized Exhaust Ports:

For these sequences, the tunnel was configured as for partial transverse exhaust ventilation. In addition, oversized exhaust ports in the ceiling were utilized.

Point Supply and Point Exhaust Operation:

The same settings as for the Partial Transverse Ventilation tests. All of the standard openings for the supply and exhaust of the tunnel closed. Large, normally-closed openings in the ceiling were used as discrete points to supply or exhaust air.

Natural Ventilation:

The false ceiling of the tunnel was removed so that a bigger cross section of the tunnel was available. A fire was started in the middle of the tunnel and the tunnel was just ventilated by the effect of buoyancy.

Longitudinal Ventilation with Jet Fans:

Longitudinal Ventilation Tests are similar to the simulations that have been done in the course of this diploma thesis. Therefore a Longitudinal Ventilation Test will be simulated. The original tunnel ventilation system was a full transverse type, consisting of a supply fan chamber at the south portal and an exhaust fan chamber at the north portal. In order to prepare the Memorial Tunnel for longitudinal ventilation using jet fans, major modifications to the facility were necessary. The false ceiling of the tunnel was removed so that the jet fans could be installed in the crown of the tunnel. Five groups of jet fans were installed for the conduct of 12 tests. The spacing for the groups was approximately 95 m, but the exact position of all fans is not recorded in the documentation of the MTFVTP. However, the position of the fans can be estimated with the help of sketches like shown in Figure 6.52. Three jet fans in a group were installed so that the center of the fans was 2.3 m from the tunnel ceiling and 2.4 m horizontally from each other (pictured in Figure 6.50). Each jet fan was equipped with a 75 PS motor rated to deliver $43 \text{ m}^3/\text{s}$ of air at an exit velocity of 34 m/s , and designed to withstand air temperatures of 300°C .

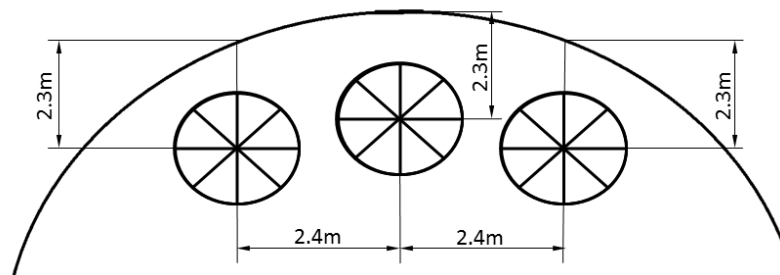


Figure 6.50: Heat Source: Fan group

The tunnel was equipped with instrumentation and recording equipment for data acquisition. Figure 6.51 shows a measurement tree which was used for measuring air velocity, temperature, carbon monoxide (CO) and carbon dioxide (CO_2) concentration.



Figure 6.51: Heat Source: Measurement Tree [30]

Several sensors were situated at different elevations. Table 6.22 shows the elevations of measurement trees sensors.

Table 6.22: Heat Release: Elevation Levels of sensors

| Elevation Level | Height [m] |
|-----------------|------------|
| Ground | 0.00 |
| A | 0.30 |
| B | 1.20 |
| C | 2.15 |
| D | 3.35 |
| E | 4.50 |
| F | 5.45 |
| G | 6.20 |
| H | 7.10 |

The positions of the measurement trees in longitudinal direction of the tunnel are shown in Table 6.23. Not all locations of measurement trees that were used for the fire tests are listed in the table. But since the part of the tunnel with the fire in it is of interest, only this part will be simulated, and only the relevant measurement trees are mentioned in the table below.

Table 6.23: Heat Release: Position of the measurement trees in longitudinal direction

| Measurement Tree | Position in longitudinal direction [m] |
|------------------|--|
| North Portal | -509 |
| Loop 207 | 0 |
| Loop 307 | 45.1 |
| Loop 306 | 77.7 |
| Loop 305 | 95.7 |
| Loop 304 | 119.2 |
| Loop 303 | 133.5 |
| Loop 302 | 173.1 |
| Loop 301 | 214.9 |
| South Portal | 345 |

The Longitudinal Ventilation Test Series included several tests with different test parameters. The fire is situated 107 m behind Loop 207 (in direction of the south portal) for all tests. The basic setup of the tests is shown in Figure 6.52.

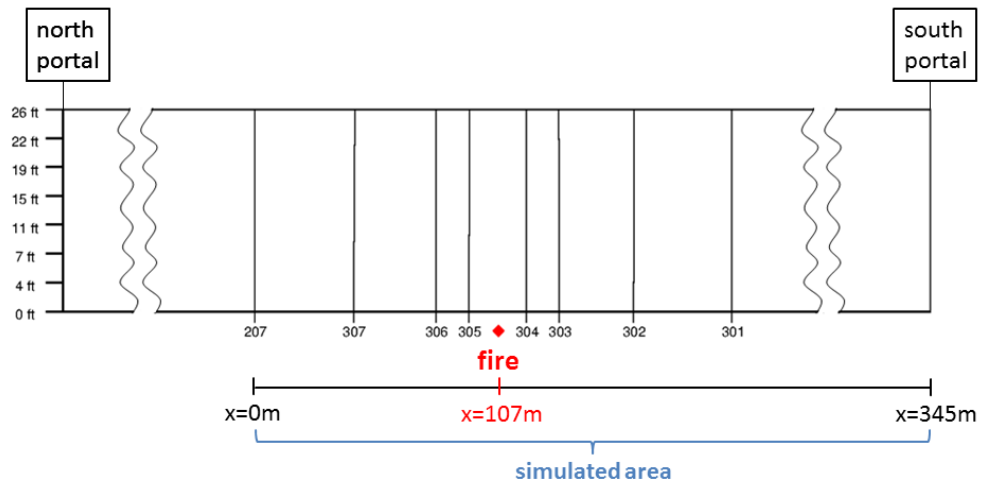


Figure 6.52: Heat Source: Basic setup for Longitudinal Ventilation Tests

Fans are not operating constantly over the whole test period. Some fans are switched on and off during the tests. It is difficult to implement the process of running up and shutting down the fans in a simulation. Test 610 includes a period of 320 s in which no fan is switched on or off. Starting point of the simulations will be the moment in which the fans 7,9,13 and 15 are running at the same time and won't be shut down for at least 300 s. That is at elapsed test time 20 : 18 – 25 : 38 (look at Table 6.24). The location of each fan is shown in Figure 6.52.

Table 6.24: Heat Release: Fan Operating Time

| Jet Fans Running | Elapsed Test Time |
|------------------|-------------------|
| 2,8,11,14 | 0:00-5:31 |
| 2,5,8,11 | 5:31-9:52 |
| 2,5,8,11,14 | 9:52-14:56 |
| 5,8,11,13,14 | 14:56-15:24 |
| 8,11,13,14,15 | 15:24-16:15 |
| 7,8,13,14,15 | 16:15-16:39 |
| 7,8,9,13,14,15 | 16:39-18:19 |
| 7,8,9,13,15 | 18:19-20:18 |
| 7,9,13,15 | 20:18-25:38 |
| 7,8,9,13,15 | 25:38-End of Test |

The measurement data at test time 20 min are used to define the initial conditions for the simulation. Cross section and perimeter can be calculated with the dimensions

shown in Figure 6.49. The cross-sectional area is 61.2 m^2 and the perimeter is 29.6 m . The simulated tunnel length is 345 m . The flow rate at the position of all measurement trees is shown in Figure 6.53. The flow rate of $128.4 \text{ m}^3/\text{s}$ (equal to a average flow velocity of 2.1 m/s) in front of Loop 214, cannot be defined like the flow profile shown in Figure 6.52. Therefore, a lead-in length of 300 m is used in front of Loop 207. The length of the lead-in area has been estimated with Equation 2.48. A rectangular flow profile is defined at the inlet of the lead-in area. After the length of 300 m the flow should be a non-disturbed flow and the flow profile should approximately look like the flow profile shown in Figure 6.52. The active jet fans are also shown in Figure 6.53. Exact positions of the fans in longitudinal direction of the tunnel are not mentioned in the documentation of the MTFVTP, but the approximate position of the active fans can be read from plans [30]. Fan 13 and 15 are about 150 m upstream of the fire and the fans 7 and 9 are about 350 m upstream. The heat release rate of the fire is 50 MW . The measurement data at test time 20 min are pictured in Figure 6.54 and are used as comparative data for the simulation results. The exact data at each sensor of the measurement trees are saved in the form of Excel tables.

Note: The geometry in the MTFVTP report is not described in all details. Pictures from this report show vehicles, pallets, equipment and other things in the tunnel, which cannot be taken into account in the simulations. Furthermore, the distance between the fan groups is known, but the exact position of the fan groups is unknown. Therefore the position of the fans is estimated with the help of Figure 6.53. Another important influence factor is the incline of the tunnel. A incline of 3.2% is the average incline from south to north, but the slope of the tunnel at every position cannot be read from the report. All this unknown data causes simulation results, which can never be as accurate as the results of the experiment.

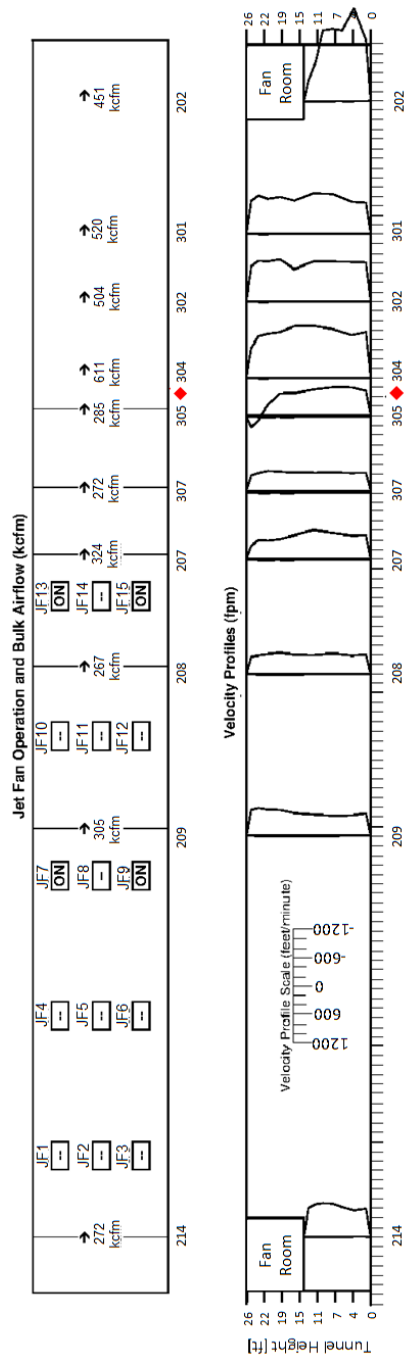


Figure 6.53: Heat Source: Initial conditions for the simulation [30]

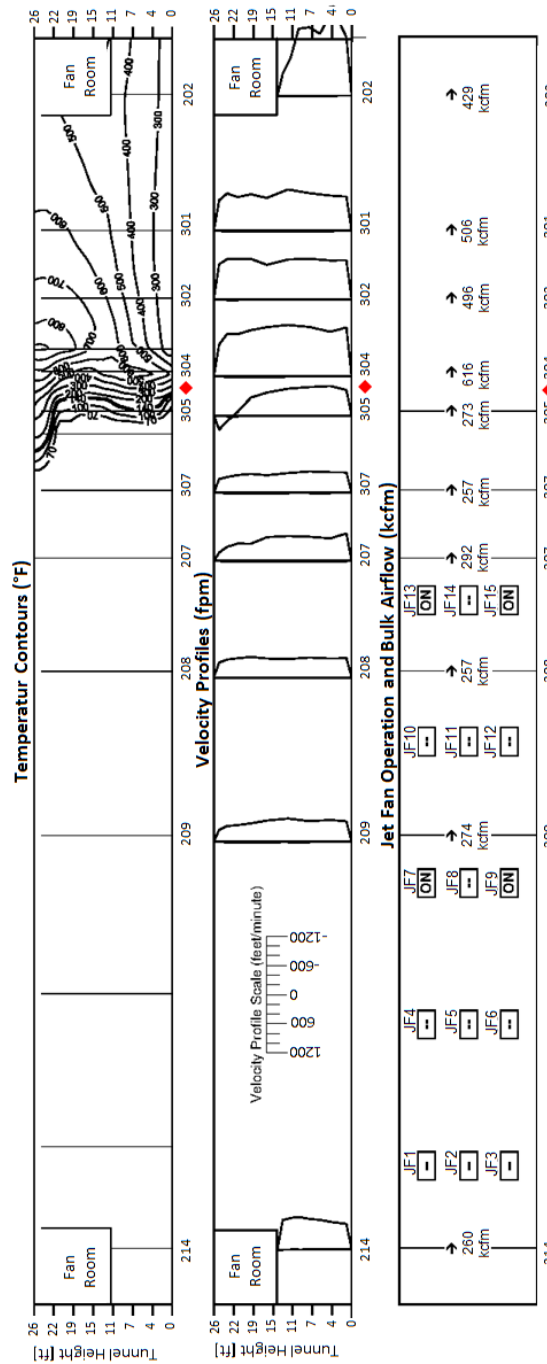


Figure 6.54: Heat Source: Comparative data for the simulations [30]

6.4.2 Heat Source - NUMSTA3

An illustration of velocity profiles or temperature fields is not possible in NUMSTA3. Only averaged values over the cross section can be calculated by NUMSTA3. The measurement data gives information about the temperature and velocity in individual points, but no information about the temperature profile over a certain cross section. For this reason a graph will be created that shows the average temperature over the tunnel axis. Because the average temperature in each cross section cannot be read out of the memorial test data, the NUMSTA3 results will be compared to a simulation in OpenFOAM. This comparison is only acceptable, if the OpenFOAM simulation is close to the measurement data.

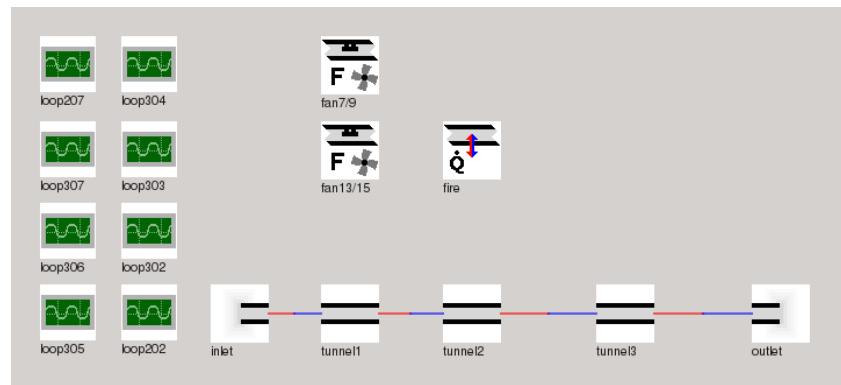


Figure 6.55: Heat Source: NUMSTA3 model

The NUMSTA3 model is pictured in Figure 6.55. The element named “inlet” is used to define the flow rate of $128.4 \text{ m}^3/\text{s}$ at the inlet of the tunnel. The lead-in area is represented by the element named “tunnel1”, which has a length of 300 m and a cell length of 5 m. The examined length of 345 m is created with the element named “tunnel2” and has a resolution of 0.5 m. A lead-out area is used to keep the simulation as similar as possible to the previous simulations. The element “tunnel3” is also 300 m long and has a resolution of 5 m. All tunnel elements are created with a 3.2% incline from south to north. Fans which are in the same fan group are summarized by one fan element, because their three-dimensional arrangement has no effect on the NUMSTA3 simulation anyway. In this element the jet velocity of 34 m/s and a thrust must be defined. The thrust can be calculated with the information in Chapter 6.4.1 and the following equations:

$$F_S = \dot{m}c$$

$$F_S = \rho \dot{V}c$$

$$F_S = 1.188 \text{ kg/m}^3 * 43 \text{ m}^3/\text{s} * 34 \text{ m/s}$$

$$F_S = 1,737 \text{ N}$$

The calculated thrust is multiplied by two to get a thrust of 3,474 N for the summarized jet fan group. These fan elements are located 150 m and 350 m upstream of the element named “fire”, which is used to create a heat source with a heat release rate of 50 MW. Probe elements (Elements in NUMSTA3, which enable recording quantities over time.) are used to define the measurement trees. Different elevations of the loops can’t be defined because of the one-dimensionality of NUMSTA3. The pipe drag coefficient λ for a hydraulic smooth wall is calculated and has a value of 0.009. The simulation starts at minute 20 of the memorial fire tests and ends at minute 24. Therefore, the maximal simulation time has been set to 240 s. The solver is set to solve the equations for an unsteady system. The heat model for a radial heat transfer is used. This heat model determines the heat transfer coefficient with Equation 2.81. An approximate Riemann solver of Roe is used to simulate the flow inside the tunnel. The convective terms are discretized by a first order upwind difference scheme.

The probes in the NUMSTA3 model write the values of all physical quantities at the defined location into a CSV file. This CSV file contains the values for the temperature T and the velocity u .

To estimate the value of the maximum temperature occurring in the tunnel, the following equation can be used:

$$\dot{Q} = \dot{m}c_p\Delta T = \rho\dot{V}c_p\Delta T \quad (6.15)$$

A heat capacity of 1.004 kJ/kgK is assumed, and the heat source is multiplied by the fire efficiency of 75 %. The fire efficiency parameter has already been explained in Chapter 5.4 and according to the RVS [2] the fire efficiency for a HRR of 50 MW is 75 %. This leads to the following mean temperature increase:

$$\begin{aligned} \Delta T &= \frac{50 \text{ MW} * 0.75}{1.188 \text{ kg/s} * 128.4 \text{ m}^3/\text{s} * 1004 \text{ J/kgK}} \\ \Delta T &= 244.9 \text{ K} \\ t &= t_{\text{ambient}} + \Delta T \\ t &= 20^\circ\text{C} + 244.9^\circ\text{C} = 264.9^\circ\text{C} \end{aligned}$$

Figure 6.56 shows the temperature trend of the NUMSTA3 simulation. NUMSTA3 has a problem with the fans. Because of the one-dimensional characteristics of NUMSTA3, no recirculation can occur near the fans. So if the flow rate at the inlet and the thrust of the fans do not fit exactly together, oscillations occur downstream of the fire. Additionally, the maximum temperature is much higher than the result of the estimation with Equation 6.15.

One way to avoid this problem is the use of a steady solver. A steady solver smooths the oscillations and prevents the overshoot of the temperature trend. The desired solution is steady anyway, and the use of a steady solver doesn’t cause any uncertainties. The temperature trend gained by a steady simulation is shown in Figure 6.57. The

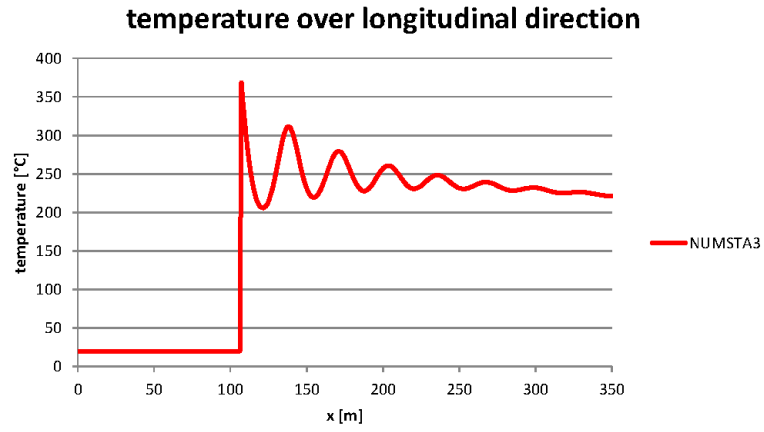


Figure 6.56: Heat Source: Temperature trend of the transient NUMSTA3 simulation

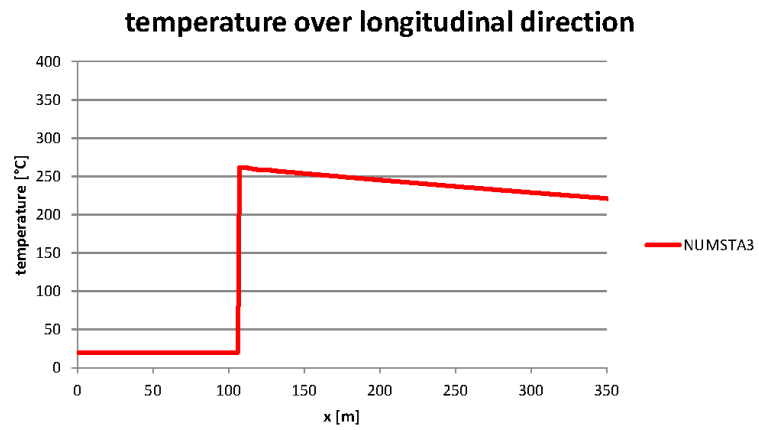


Figure 6.57: Heat Source: Temperature trend of the steady NUMSTA3 simulation

Simulations for Comparing the Programs

maximum temperature in the results of the steady NUMSTA3 simulation is 286.3 °C, which is close to the estimation of Equation 6.15.

6.4.3 Heat Source - OpenFOAM

The geometry in OpenFOAM is created with the help of **Gmsh** and **enGrid**. In order to keep the simulation time as short as possible, only a half of the tunnel is simulated and a symmetry plane is used. Figure 6.58 shows the Gmesh geometry of the Memorial Tunnel. The maximum edge length of the cells in the Gmesh grid is approximately 0.2 m.

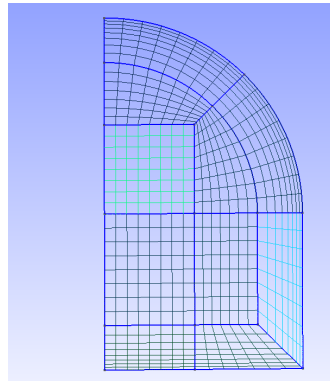


Figure 6.58: Heat Source: Geometry in Gmesh

The Gmesh geometry is imported in enGrid and extruded to a length of 345 m with a cell length of 0.5 m. The “extrude” function is used to create a 300 m long lead-in and lead-out area. The inlet velocity of 2.1 m/s is defined as a rectangular flow profile at the inlet of the lead-in area. To take the heat transfer from the fluid to the wall into account, the newest version of enGrid (enGrid-1.4x) is necessary. enGrid-1.4x enables the creation of a mesh, which includes two different volumes. In this case, a volume named “air” is used to create the flow area, and a volume named “solid” is used to create the solid walls of the tunnel. Figure 6.59 shows the enGrid model of the Memorial Tunnel. The fire with a heat release rate of 50 MW is defined in a file named heatDict. The symmetry requires to define a 25 MW heat source to generate the same results like the complete geometry would create. Loops don’t have to be defined in the OpenFOAM input files because Paraview enables the viewing of all physical quantities of each cell. A file named fanDict allows to create momentum sources. Location, size and exit velocity must be defined to create the fans. The inner diameter d_i (see below) can be calculated with the data in Chapter 6.4.1. A gravity vector can be defined in a file named “g” and consequently the grade of the tunnel (Figure 6.60).

$$\begin{aligned} \dot{V} &= c * A \\ A &= \frac{d_i^2 * \pi}{4} \\ \dot{V} &= c \frac{d_i^2 * \pi}{4} \\ d_i &= \sqrt{\frac{4\dot{V}}{c\pi}} \\ d_i &= \sqrt{\frac{4 * 43 \text{ m}^3/\text{s}}{34 \text{ m}/\text{s}\pi}} \\ d_i &= 1.25 \text{ m} \end{aligned}$$

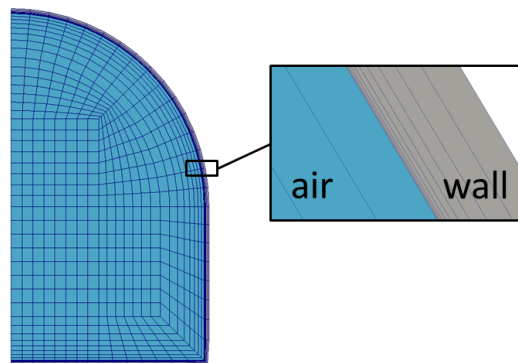


Figure 6.59: Heat Source: Geometry in enGrid

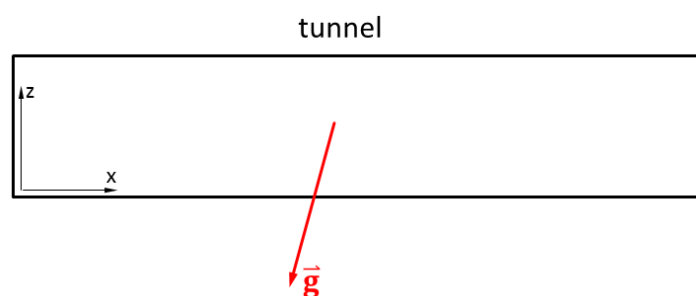


Figure 6.60: Heat Source: Orientation of the gravity vector

The used solver settings are:

- chtMultiRegionFoam for an unsteady scheme, which is able to simulate two phases (air and solid)
- RASModel $k-\omega-SSTF$, which switches between the $k-\omega-SST$ near walls and LES simulation in the center of the tunnel
- Gauss linear for gradSchemes
- Gauss linearUpwind Gauss for U divSchemes
- Gauss upwind for k divSchemes
- Gauss linear corrected for laplacianSchemes
- DICPCG for the pressure
- DILUPBiCG for U, ω and k

The listed terms are no mathematical terms. They are predefined commands to select schemes in OpenFOAM. The meaning of each term can be read in the OpenFOAM User's Guide [20]. Figure 6.61 shows the results after a simulation time of 240 s. The simulated temperature profile is similar to the temperature measurements. One significant phenomenon is the appearance of backlayering. Backlayering is the reversion of movement of smoke and hot gases against the direction of the ventilation airflow. The backlayering in Figure 6.61 is greater than the backlayering in Figure 6.54. The flow profiles will be described in Chapter 6.4.5.

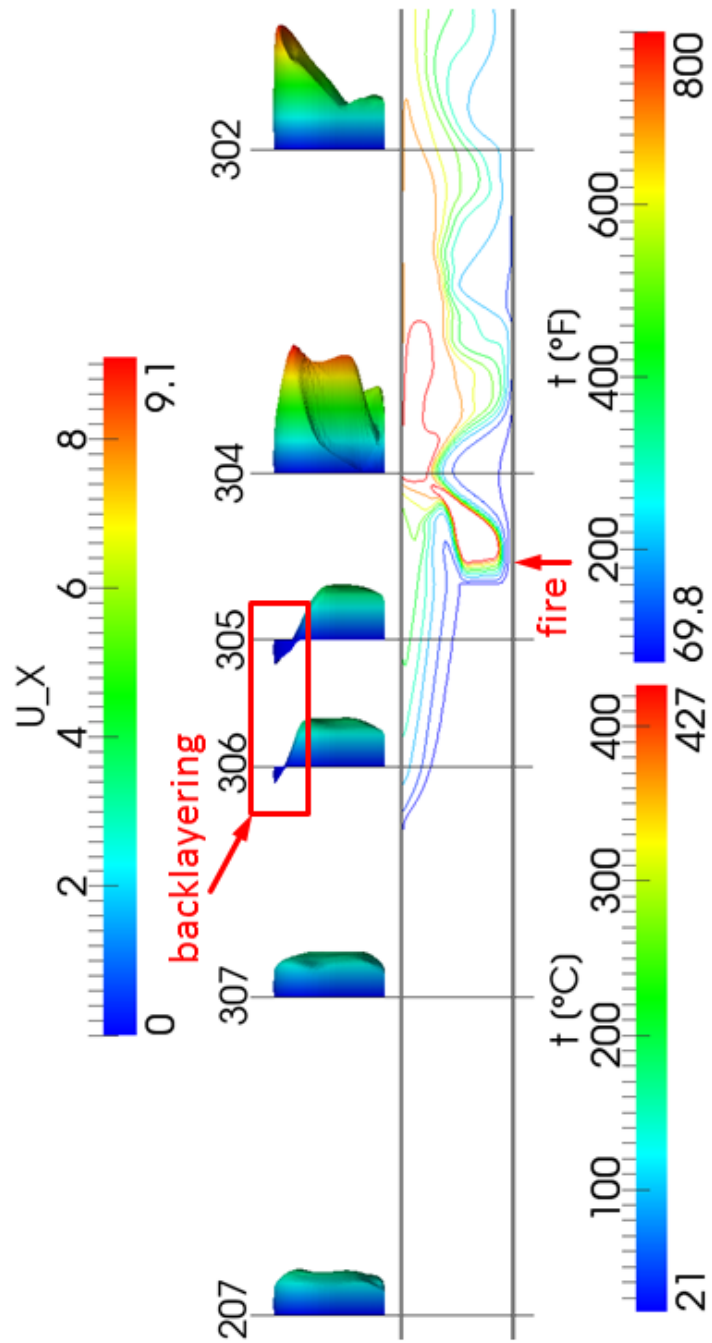


Figure 6.61: Heat Source: Results of the simulation in OpenFOAM

6.4.4 Heat Source - FDS

The Memorial Tunnel geometry for the FDS simulation is created in Blender. The tunnel has a length of 945 m. The horseshoe cross section of the tunnel can be created in Blender, but FDS will change the round geometry to a sawtooth profile. The computing domain is separated in 3 meshes. Two of the meshes are used to define the coarse resolution of $0.4\text{ m} \times 0.4\text{ m} \times 5\text{ m}$ for the lead-in and lead-out area. In a third &MESH block, the resolution of $0.4\text{ m} \times 0.4\text{ m} \times 0.5\text{ m}$ for the examined geometry is defined. Also, the definition of a round inlet &VENT block and circular cross sections of the fans are not possible in FDS. Therefore, rectangular surfaces are defined in Blender. A symmetry plane is used to reduce simulation duration. A surface with a heat release rate of 25 MW represents the fire. Devices are placed at each location of the measurement tree sensors. A surface with SURF_ID='OPEN' enables air and smoke to exit the tunnel at the south portal. A gravity vector is defined to create the incline of 3.2%. Appendix A.4 includes the FDS input file. Figures 6.62 to 6.64 show the Blender and FDS geometry.

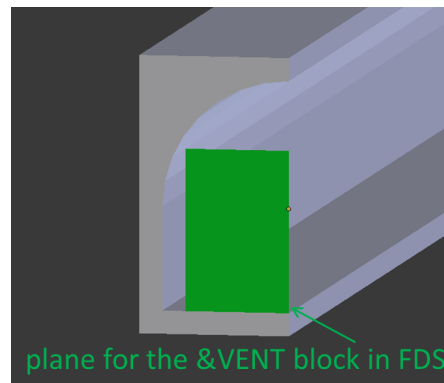


Figure 6.62: Heat Source: Geometry in Blender

Figure 6.65 shows the results of the simulation in FDS. An illustration with isolines is not possible in Smokeview. One advantage of FDS is that it is easy to understand unsteady processes. Figure 6.65 includes only screen shots of a video which is provided by Smokeview. The video can be used to picture the behavior of the flow easily. Furthermore, influence factors like the position of the fans can be understood and realized fast.

The results of FDS show a highly developed backlayering compared to the measurement results, which could be caused by the angular shaped geometry in the FDS model. The hot gases at the ceiling of the tunnel have approximately the same temperature as in the OpenFOAM simulation. Figure 6.66 shows the visualization of smoke in FDS. The other software used in this diploma thesis are not able to visualize processes like smoke spreading as quickly and easily as FDS.

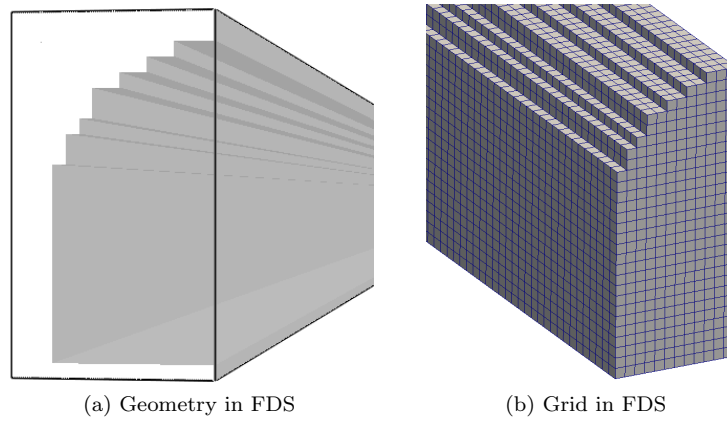


Figure 6.63: Heat Source: FDS model

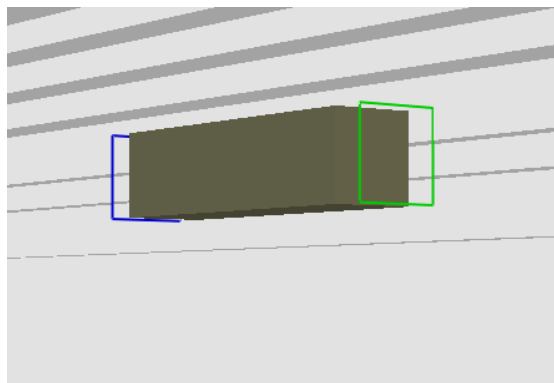


Figure 6.64: Heat Source: Fans in FDS

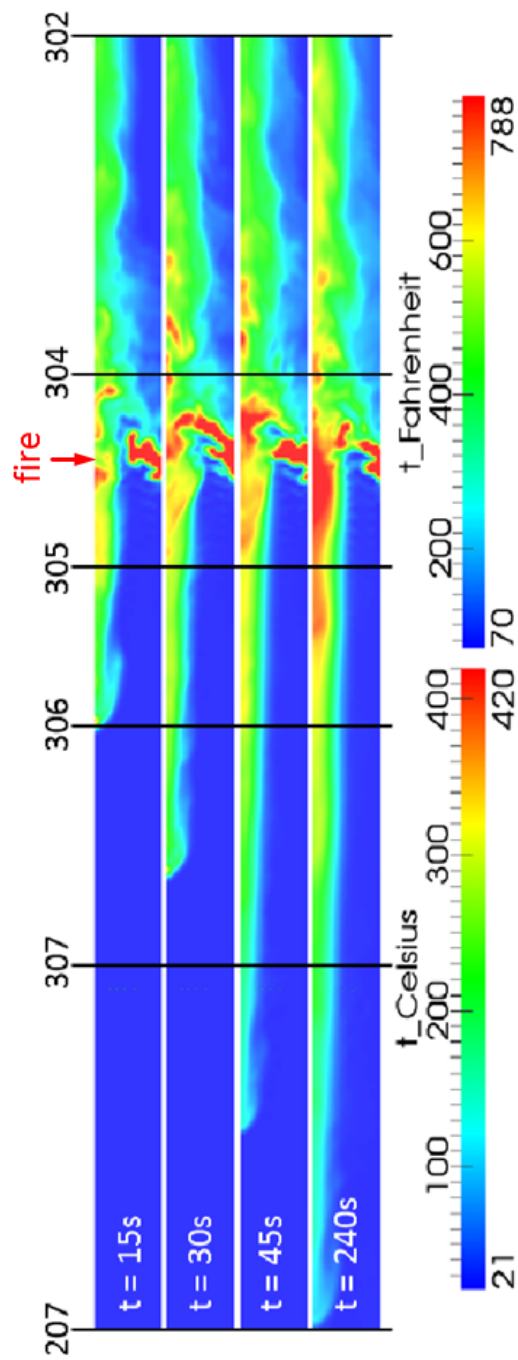


Figure 6.65: Heat Source: Results of the simulation in FDS

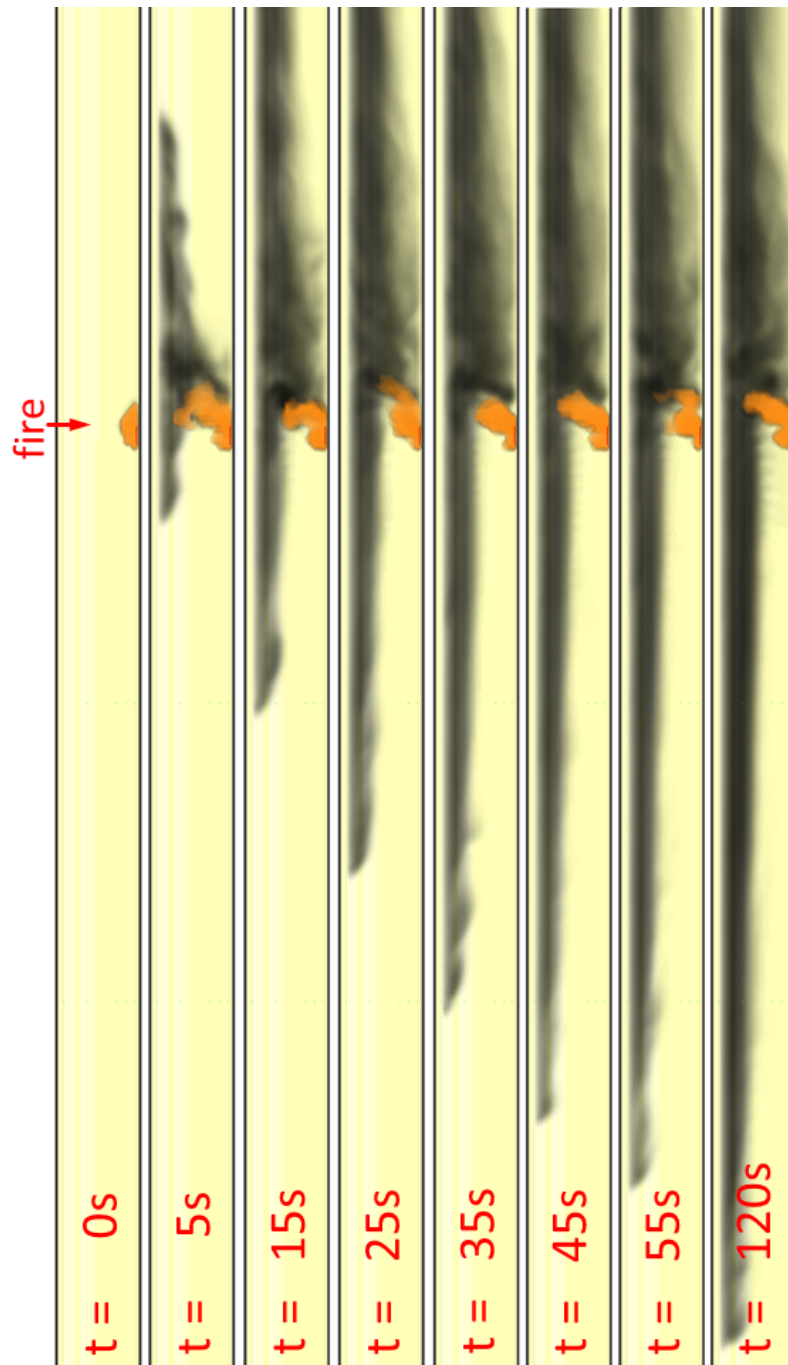


Figure 6.66: Spreading of smoke in FDS

6.4.5 Heat Source - Comparison of the Results

Figures 6.67a to 6.67e show the comparison of the velocity profiles at all loops with existing measurement data. A remarkable deficiency in all figures is the tangential velocity at walls in all FDS velocity profiles. In Figure 6.67a the velocity profiles of the two simulations are similar, but they differ from the measurements. The reasons for this are the above mentioned missing information (no information about the local incline and geometry details) and details which cannot be simulated even if the information would be available. In Figure 6.67b the flow profile in OpenFOAM does not have the same values as the measurement at each elevation of the measurement tree, but the flow profiles look similar. FDS overestimates the backlayering process and results in a flow profile at Loop 307, which contains a highly developed backlayering. Figure 6.67c shows the backlayering at Loop 305. The measurements show that the hot gases at the ceiling of the tunnel flow against the flow direction. OpenFOAM describes this process quite well and the velocity of the layer at the ceiling is very close to that in the measurements. However, FDS results in a high velocity at the tunnel ceiling and therefore in a strong backlayering. In the Figures 6.67d and 6.67e the flow profiles of FDS and OpenFOAM show similar profiles, where the flow velocity at the ceiling is high. This effect is caused by the buoyancy of the hot gases. The temperature trend over longitudinal direction in NUMSTA3 is compared to OpenFOAM. The reason for this is the good agreement of OpenFOAM's results with the measurement data. In order to enable a comparison between NUMSTA3 and OpenFOAM, the average value of the temperature every 25 m is calculated in OpenFOAM and plotted in Figure 6.68. The maximum temperature of both simulations are close to each other, but the temperatures downstream of the fire are different.

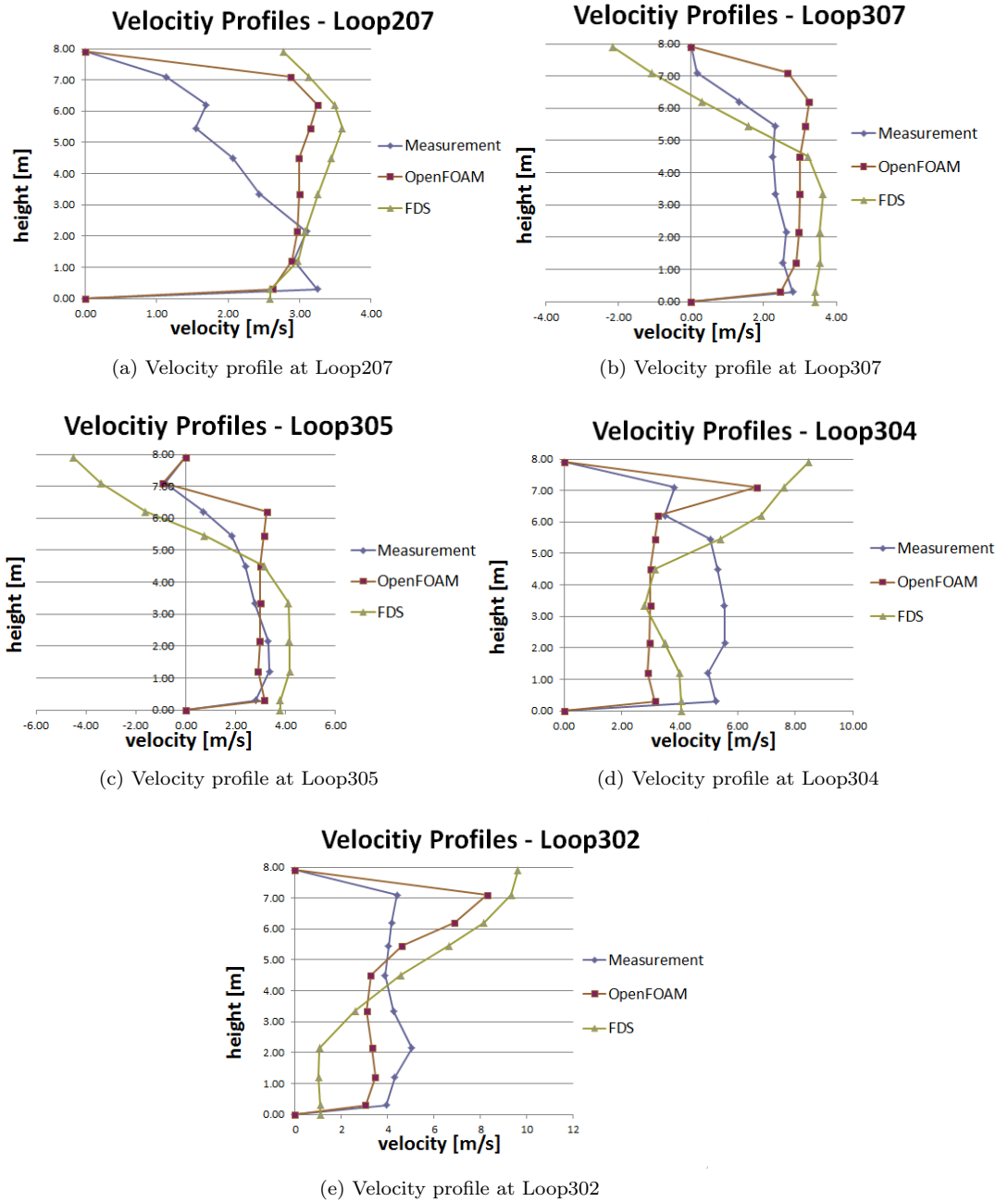


Figure 6.67: Velocity profiles at several Loops

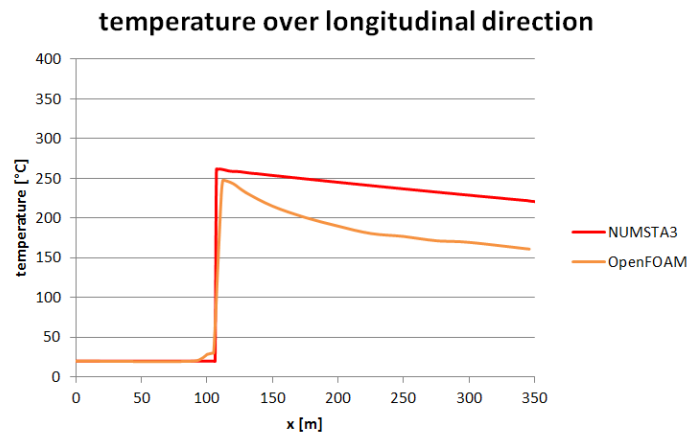


Figure 6.68: Heat Source: Comparison between the temperature trend of NUMSTA3 and OpenFOAM

6.4.6 Heat Source - Reasons for Deviating Results

Difference between simulation and measurement:

The biggest influence factor that causes differing results is the geometry. Missing information about the exact geometry of the tunnel (including lay-bys, vehicles and equipment in the tunnel, etc.) makes it impossible to generate results in simulations that are equal to the measurement data. The local incline at each location inside the tunnel is also unknown and causes deviating flow profiles caused by buoyancy. This is a reason for the difference between the simulations and the measurement in Figure 6.67e. The missing information about the exact position of the fans can be mentioned as a further reason for differences between the simulations and the measurements, but it is not a cause for differences between FDS and OpenFOAM.

Difference between FDS and OpenFOAM:

Fans at the ceiling of a tunnel are used to ventilate the tunnel and prevent back-layering. Different positions of the fans lead to different smoke spreading. The fan groups are situated at the same position and are running with the same performance in every simulation. Therefore they cannot be the reason for different predicted flow profiles. A reason for the difference between FDS and OpenFOAM is the geometry of the ceiling. The angular shaped geometry blocks parts of the flow domain and provides less space for the hot gases. The hot gases are forced to flow with a higher velocity and to spread more in a longitudinal direction. The effect of the blocked flow domain is sketched in Figure 6.69, where the red areas are “lost” cross sections and the blue areas are “gained” cross sections.

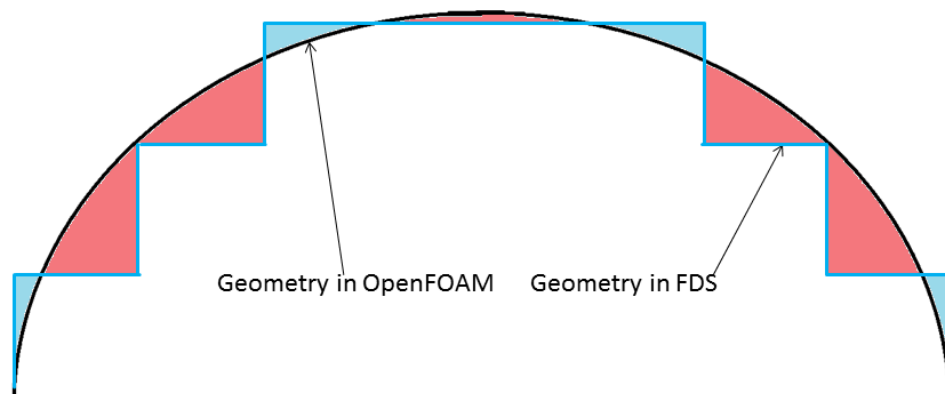


Figure 6.69: Heat Source: Different geometries in OpenFOAM and FDS

NUMSTA3: The temperature level near the fire is close to the temperature level of the OpenFOAM simulation. The reason for the similar temperature levels is the fire efficiency which is used in NUMSTA3. Without the reduced heat release rate, the air temperature in the NUMSTA3 simulation would be very high because of the missing heat radiation model. The maximum temperature on the NUMSTA3 simulation is not very accurate because the fire efficiency is only estimated and it is not a real physical quantity that can be determined. Upstream of the fire the temperature level in OpenFOAM is higher than the ambient temperature. This higher temperature niveau is caused by backlayering. In a one-dimensional simulation, backlayering effects cannot be simulated and therefore the temperature niveau in NUMSTA3 is at ambient level. The difference downstream of the fire is caused by the “wrong” heat model (only convection is simulated and the HRR is multiplied by the fire efficiency) and the one-dimensionality in NUMSTA3. In the OpenFOAM simulation hot gases flow at the ceiling of the tunnel. The high temperature difference between these gases and the wall causes a high heat transfer to the wall and the air flow is cooled down. Furthermore, the heat radiation from the gas to the wall is taken into account in the OpenFOAM simulation but not in the NUMSTA3 simulation. All these effects cause the air in the OpenFOAM simulation to cool down faster than in the NUMSTA3 simulation.

7 Summary

This chapter will summarize the information of the software used throughout this diploma thesis. Additionally, some personal impressions will be mentioned.

NUMSTA3:

- + Simple and quick generation of geometries
- + No additional software for the generation of the geometry and the grid is necessary
- + Very accurate for simple geometries
- + Few cells compared to three-dimensional software and therefore short simulation times and less computing resources required
- + No additional software for the visualization is necessary
- + Offers the opportunity to implement own solvers and models
- Only usable for geometries with one-dimensional characteristics
- It is difficult to take three-dimensional effects into account (e.g. if small obstructions are attached to the wall, the pipe drag coefficient λ can be increased but no obstructions are modeled)
- No heat radiation model is available
- No opportunity for parallelized simulations

OpenFOAM:

- + Very powerful and flexible tool
- + Offers the opportunity to implement individual solvers and models
- + Geometry and grid can be created with and without additional software
- + Complex three-dimensional geometries can be generated and meshed
- + Refinement of the boundary layer is possible
- + Meshing process offers many possibilities, but requires special know-how about the different turbulence models (estimate the correct cell height for a y^+ -value that is suitable for the turbulence model)

Summary

- + Parallelized simulations are possible
- Requires a relatively long time for the meshing process
- requires basic knowledge about operating on Linux Systems for pre-processing, simulating and post-processing
- More complicated to use than the other software
- Additional software for the visualization is necessary

FDS:

- + Geometry can be created with and without external software
- + Limited to “simple” geometries
- + Fast meshing process
- + Allows simple and fast visualization of unsteady processes like smoke spreading
- No refinement for the boundary layer is possible
- LES requires long simulation times compared to other software
- Additional software for the visualization is necessary
- Locations in the geometry that will be analyzed later, must be defined before the simulation is started
- Uses a “wrong” boundary condition for the velocity at walls

Table 7.1 shows subjective strengths and weaknesses of the used software.

| | NUMSTA3 | OpenFOAM | FDS |
|-----------------------------------|---------|----------|-----|
| Simulation of complex geometries | o | ++ | o |
| Refinement for the boundary layer | - | ++ | - |
| Fast meshing process | ++ | - | ++ |
| Fast simulation | ++ | o | - |
| Field of application | + | ++ | + |
| Accuracy for investigated cases | ++ | + | - |

Table 7.1: Strengths/Weaknesses of the used software

8 Conclusion

The objective of this diploma thesis was to determine the accuracy of NUMSTA3, OpenFOAM and FDS in standard flow problems and provide relevant ambit for the electrical and mechanical department of ILF consulting engineers. In order to achieve these goals, different simulations have been performed. The simulations have shown that each program is only suitable for specific applications.

NUMSTA3 is best suitable for flows without three-dimensional effects, because it is a one-dimensional CFD software. This software enables the fastest, simplest and most accurate simulation of standard flow problems. The pressure losses and average velocities can be calculated fast and accurate. Heat sources and averaged temperatures in cross sections can be simulated, but if the heat transferred by radiation is too large the result is not accurate enough. The reason for this handicap is the non-existent heat radiation model. To solve this deficiency, the heat transfer coefficient must be adjusted manually or a heat radiation model must be implemented. As mentioned above, three-dimensional effects as backlayering and the detachment of the boundary layer cannot be simulated and are therefore out of NUMSTA3's field of application.

OpenFOAM is the most flexible and versatile of the used software. It allows to choose between different solvers and turbulence models. Also individual solvers can be written and implemented in this software. On one hand, the generation of a geometry and of the numerical grid is much more time consuming than in NUMSTA3. On the other hand, the meshing process offers many possibilities like a refinement of the boundary layer, which is required for achieving accurate and reliable results. The relatively huge amount of cells, caused by the necessary refined mesh for the boundary layer, additionally induces a longer computing time than NUMSTA3. Because of these facts, NUMSTA3 should be preferred to OpenFOAM in the case of simple geometries without any three-dimensional effects. Effects which require a highly resolved numerical grid like the detachment of the boundary layer or backlayering processes can only be simulated in OpenFOAM. Also heat radiation can be considered, which leads to more accurate temperature profiles when heat radiation is a significant part of the heat transfer. On the basis of all these attributes, OpenFOAM should be used for three-dimensional flows, which require highly resolved numeric grids and if a accurate results are required.

FDS should not be used for simulations in which the pressure loss caused by wall friction has great influence on the results. The slip boundary condition leads to tangential velocities at walls and therefore to incorrect velocity profiles. The Large Eddy Simulation requires long simulation times, even if the amount of cells is not very high.

Conclusion

Also a refinement of the grid in near-wall regions is not possible and disqualifies FDS for the simulation of processes which include a detachment of the boundary layer. Besides all the mentioned disadvantages, FDS allows in combination with Smokeview to visualize unsteady processes in a simple and descriptive way. Even if the results are not very accurate, the trend of the results is correct. So if the trend of a process is enough (Like when the smoke spreading in the smoke extraction in a building should be simulated.) and the required accuracy is not very high, FDS is a good choice.

Bibliography

- [1] Rudolf A. (1998). Simulation of Compressible Flow in Tunnel Systems Induced by Trains Traveling at High Speed. Eidgenössische Technische Hochschule Lausanne, Fakultät für Maschinenbau
- [2] Richtlinien und Vorschriften für das Straßenwesen (RVS), RVS 09.02.31, Ausgabe 1. August 2008
- [3] Richtlinien für die Ausstattung und den Betrieb von Straßentunneln (RABT), Ausgabe 2006
- [4] Brenn G., Meile W. (2010). Vorlesungsskriptum Strömungslehre und Wärmeübertragung I. Technische Universität Graz, Institut für Strömungs und Wärmeübertragung.
- [5] Steiner H. (2011). Vorlesungsskriptum Höhere Strömungslehre und Wärmeübertragung I. Technische Universität Graz, Institut für Strömungs und Wärmeübertragung.
- [6] Hochenauer C. (2012). Vorlesungsskriptum Wärmetechnik 2. Technische Universität Graz, Institut für Wärmetechnik.
- [7] http://de.wikipedia.org/wiki/Dimensionslose_Kennzahl [11.02.2013]
- [8] <http://www.dept.aoe.vt.edu/~jschetz/fluidnature/unit07/unit7f.html> [13.02.2013]
- [9] Idelchik I.E. Handbook of Hydraulic Resistance 3rd Edition, CRC Press
- [10] Offner G. (2012). Numerische Methoden in der angewandten Thermodynamik, Numerische Grundlagen. Technische Universität Graz, Institut für Verbrennungskraftmaschinen und Thermodynamik
- [11] Sanz W. (2011). Numerische Verfahren in der Energietechnik, Kapitel: Computational Fluid Dynamics. Technische Universität Graz, Institut für Thermische Turbomaschinen und Maschinendynamik
- [12] Sanz W. (2011). Numerische Verfahren in der Energietechnik, Kapitel: Finite Elemente in der Wärmeübertragung. Technische Universität Graz, Institut für Thermische Turbomaschinen und Maschinendynamik
- [13] <http://www.exp.univie.ac.at/cp1/cp1-6/node52.html> [11.03.2013]

- [14] Wilcox D.C. (1988), Turbulence Modeling for CFD (Third Edition), California: DCW Industries.
- [15] Menter F.R.(1994), Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications, AIAA Journal, Vol. 32, No. 8, August 1994
- [16] Almbauer R. (2012). Numerische Methoden in der angewandten Thermodynamic. Technische Universität Graz, Institut für Verbrennungskraftmaschinen und Thermodynamik
- [17] Moisi H., Schranzhofer H. (2011). Vorlesungsskriptum Numerische Verfahren in der Energietechnik, Partielle Differentialgleichungen am Beispiel der Wärmeleitung, Technische Universität Graz, Institut für Wärmetechnik
- [18] Tunnel Simulation Tool NUMSTA 3.0 Documentation
- [19] Rudolf A., Simulation of Compressible Flow in Tunnel Systems Induced by Trains Traveling at High Speed. Thèse No 1806. Lausanne: EPFL, 1998.
- [20] OpenFOAM User's Guide. <http://www.openfoam.com/docs/user/> [06.02.2013]
- [21] OpenFoam User's Guide, Version2.1.1, OpenFOAM Foundation
- [22] <http://www.openfoam.org/docs/user/fvSchemes.php>
- [23] Fire Dynamics Simulator User's Guide, Version 5, NIST Special Publication 1019-5
- [24] Fire Dynamics Simulator (Version 5) Technical Reference Guide, NIST Special Publication 1018-5
- [25] VDI - Wärmeetlas, 10. Auflage 2010
- [26] <http://geuz.org/gmsh/> [13.02.2013]
- [27] <http://engits.eu/en/engrid> [13.02.2013]
- [28] <http://www.cfd-online.com> [13.03.2013]
- [29] FLUENT 6.3 User's Guide. http://cdlab2.fluid.tuwien.ac.at/LEHRE/TURB/Fluent.Inc/fluent6.3.26/help/html/ug/main_pre.htm [13.03.2013]
- [30] Data CDs Memorial Tunnel Fire Ventilation Test Program, 1996
- [31] NIST Technical Note 1640, FDS Wall Flows Part I: Straight Channels, July 2009

Appendix

A.1 FDS: Straight Pipe Flow

```
&HEAD CHID='pipe_flow', TITLE='pipe_flow', /

!!! External config file

&TIME T_BEGIN = 0., T_END = 120/

&MISC TMPA = 20./

!!! Boundary condition defs

&SURF ID='SUPPLY', RGB=204,204,204, VOLUME_FLUX=20.0,
      COLOR='GREEN',TMP_FRONT=20., /

!!! Computational domain

&MESH ID='MESH1', IJK=250,20,20,
      XB=0.0,50.0,-0.5,0.5,-0.5,0.5, /

&MESH ID='MESH2', IJK=1000,20,20,
      XB=50.0,100.0,-0.5,0.5,-0.5,0.5, /

&MESH ID='MESH3', IJK=250,20,20,
      XB=100.0,150.0,-0.5,0.5,-0.5,0.5, /

!!! Geometry

&VENT ID='INLET', SURF_ID='SUPPLY',
      XB=0.0,0.0,-0.5,0.5,-0.5,0.5, /
&VENT ID='OUTLET', SURF_ID='OPEN',
      XB=150.0,150.0,-0.5,0.5,-0.5,0.5, /

!!! Evacuation

!!! Control logic and output
```



```
! PRESS.0
&DEVC ID='PRESS.0', QUANTITY='PRESSURE', XYZ=50.0,0.0,0.0, /
! PRESS.1
&DEVC ID='PRESS.1', QUANTITY='PRESSURE', XYZ=55.0,0.0,0.0, /
! PRESS.2
&DEVC ID='PRESS.2', QUANTITY='PRESSURE', XYZ=60.0,0.0,0.0, /
! PRESS.3
&DEVC ID='PRESS.3', QUANTITY='PRESSURE', XYZ=65.0,0.0,0.0, /
! PRESS.4
&DEVC ID='PRESS.4', QUANTITY='PRESSURE', XYZ=70.0,0.0,0.0, /
! PRESS.5
&DEVC ID='PRESS.5', QUANTITY='PRESSURE', XYZ=75.0,0.0,0.0, /
! PRESS.6
&DEVC ID='PRESS.6', QUANTITY='PRESSURE', XYZ=80.0,0.0,0.0, /
! PRESS.7
&DEVC ID='PRESS.7', QUANTITY='PRESSURE', XYZ=85.0,0.0,0.0, /
! PRESS.8
&DEVC ID='PRESS.8', QUANTITY='PRESSURE', XYZ=90.0,0.0,0.0, /
! PRESS.9
&DEVC ID='PRESS.9', QUANTITY='PRESSURE', XYZ=95.0,0.0,0.0, /
! VEL.0
&DEVC ID='VEL.0', QUANTITY='VELOCITY', XYZ=75.0,0.0,0.0, /
! VEL.1
&DEVC ID='VEL.1', QUANTITY='VELOCITY', XYZ=75.0,0.050,0.0, /
! VEL.2
&DEVC ID='VEL.2', QUANTITY='VELOCITY', XYZ=75.0,0.1,0.0, /
! VEL.3
&DEVC ID='VEL.3', QUANTITY='VELOCITY', XYZ=75.0,0.150,0.0, /
! VEL.4
&DEVC ID='VEL.4', QUANTITY='VELOCITY', XYZ=75.0,0.2,0.0, /
! VEL.5
&DEVC ID='VEL.5', QUANTITY='VELOCITY', XYZ=75.0,0.250,0.0, /
! VEL.6
&DEVC ID='VEL.6', QUANTITY='VELOCITY', XYZ=75.0,0.3,0.0, /
! VEL.7
&DEVC ID='VEL.7', QUANTITY='VELOCITY', XYZ=75.0,0.350,0.0, /
! VEL.8
&DEVC ID='VEL.8', QUANTITY='VELOCITY', XYZ=75.0,0.4,0.0, /
! VEL.9
&DEVC ID='VEL.9', QUANTITY='VELOCITY', XYZ=75.0,0.450,0.0, /

&SLCF PBX=0., QUANTITY='VELOCITY' /
&SLCF PBZ=0., QUANTITY='VELOCITY' /
```



```
&SLCF PBX=0., QUANTITY='PRESSURE' /  
&SLCF PBZ=0., QUANTITY='PRESSURE' /  
  
&TAIL /
```




A.2 FDS: Sudden Change of the Cross-Sectional Area

```
&HEAD CHID='diffuser', TITLE='diffuser', /

!!! External config file

&TIME T_BEGIN = 0., T_END = 120/

&MISC TMPA = 20./

!!! Boundary condition defs

&SURF ID='SUPPLY', RGB=204,204,204, VOLUME_FLUX=20.0,
      COLOR='GREEN',TMP_FRONT=20., /

!!! Computational domain

&MESH ID='MESH', IJK=1000,20,20,
      XB=-0.0,50.0,-0.5,0.5,-0.5,0.5, /
&MESH ID='MESH.1', IJK=1000,40,40,
      XB=50.0,1.0,-1.0,1.0,-1.0,1.0, /

!!! Geometry

&VENT ID='INLET', SURF_ID='SUPPLY',
      XB=0.0,0.0,-0.5,0.5,-0.5,0.5, /
&VENT ID='OUTLET', SURF_ID='OPEN',
      XB=1.0,1.0,-1.0,1.0,-1.0,1.0, /

!!! Evacuation

!!! Control logic and output

! PRESS.0
&DEVC ID='PRESS.0', QUANTITY='PRESSURE', XYZ=50.0,0.0,0.0, /
! PRESS.1
&DEVC ID='PRESS.1', QUANTITY='PRESSURE', XYZ=55.0,0.0,0.0, /
! PRESS.2
&DEVC ID='PRESS.2', QUANTITY='PRESSURE', XYZ=60.0,0.0,0.0, /
! PRESS.3
&DEVC ID='PRESS.3', QUANTITY='PRESSURE', XYZ=65.0,0.0,0.0, /
! PRESS.4
&DEVC ID='PRESS.4', QUANTITY='PRESSURE', XYZ=65.0,0.0,0.0, /
! PRESS.5
```



```
&DEVC ID='PRESS.5', QUANTITY='PRESSURE', XYZ=30.0,0.0,0.0, /
! VEL.0
&DEVC ID='VEL.0', QUANTITY='VELOCITY', XYZ=30.0,0.0,0.0, /
! VEL.1
&DEVC ID='VEL.1', QUANTITY='VELOCITY', XYZ=30.0,0.050,0.0, /
! VEL.2
&DEVC ID='VEL.2', QUANTITY='VELOCITY', XYZ=30.0,0.1,0.0, /
! VEL.3
&DEVC ID='VEL.3', QUANTITY='VELOCITY', XYZ=30.0,0.150,0.0, /
! VEL.4
&DEVC ID='VEL.4', QUANTITY='VELOCITY', XYZ=30.0,0.2,0.0, /
! VEL.5
&DEVC ID='VEL.5', QUANTITY='VELOCITY', XYZ=30.0,0.250,0.0, /
! VEL.6
&DEVC ID='VEL.6', QUANTITY='VELOCITY', XYZ=30.0,0.3,0.0, /
! VEL.7
&DEVC ID='VEL.7', QUANTITY='VELOCITY', XYZ=30.0,0.350,0.0, /
! VEL.8
&DEVC ID='VEL.8', QUANTITY='VELOCITY', XYZ=30.0,0.4,0.0, /
! VEL.9
&DEVC ID='VEL.9', QUANTITY='VELOCITY', XYZ=30.0,0.450,0.0, /
! VEL.010
&DEVC ID='VEL.010', QUANTITY='VELOCITY', XYZ=70.0,0.450,0.0, /
! VEL.011
&DEVC ID='VEL.011', QUANTITY='VELOCITY', XYZ=70.0,0.4,0.0, /
! VEL.012
&DEVC ID='VEL.012', QUANTITY='VELOCITY', XYZ=70.0,0.350,0.0, /
! VEL.013
&DEVC ID='VEL.013', QUANTITY='VELOCITY', XYZ=70.0,0.3,0.0, /
! VEL.014
&DEVC ID='VEL.014', QUANTITY='VELOCITY', XYZ=70.0,0.250,0.0, /
! VEL.015
&DEVC ID='VEL.015', QUANTITY='VELOCITY', XYZ=70.0,0.2,0.0, /
! VEL.016
&DEVC ID='VEL.016', QUANTITY='VELOCITY', XYZ=70.0,0.150,0.0, /
! VEL.017
&DEVC ID='VEL.017', QUANTITY='VELOCITY', XYZ=70.0,0.1,0.0, /
! VEL.018
&DEVC ID='VEL.018', QUANTITY='VELOCITY', XYZ=70.0,0.050,0.0, /
! VEL.019
&DEVC ID='VEL.019', QUANTITY='VELOCITY', XYZ=70.0,0.0,0.0, /
! VEL.020
&DEVC ID='VEL.020', QUANTITY='VELOCITY', XYZ=70.0,0.5,0.0, /
! VEL.021
&DEVC ID='VEL.021', QUANTITY='VELOCITY', XYZ=70.0,0.550,0.0, /
```



```
! VEL.022
&DEVC ID='VEL.022', QUANTITY='VELOCITY', XYZ=70.0,0.6,0.0, /
! VEL.023
&DEVC ID='VEL.023', QUANTITY='VELOCITY', XYZ=70.0,0.650,0.0, /
! VEL.024
&DEVC ID='VEL.024', QUANTITY='VELOCITY', XYZ=70.0,0.7,0.0, /
! VEL.025
&DEVC ID='VEL.025', QUANTITY='VELOCITY', XYZ=70.0,0.750,0.0, /
! VEL.026
&DEVC ID='VEL.026', QUANTITY='VELOCITY', XYZ=70.0,0.8,0.0, /
! VEL.027
&DEVC ID='VEL.027', QUANTITY='VELOCITY', XYZ=70.0,0.850,0.0, /
! VEL.028
&DEVC ID='VEL.028', QUANTITY='VELOCITY', XYZ=70.0,0.9,0.0, /
! VEL.029
&DEVC ID='VEL.029', QUANTITY='VELOCITY', XYZ=70.0,0.950,0.0, /

&SLCF PBX=0., QUANTITY='VELOCITY', VECTOR=.TRUE. /
&SLCF PBZ=0., QUANTITY='VELOCITY', VECTOR=.TRUE. /

&SLCF PBX=0., QUANTITY='PRESSURE', VECTOR=.TRUE. /
&SLCF PBZ=0., QUANTITY='PRESSURE', VECTOR=.TRUE. /

&TAIL /
```

A.3 FDS: Change of Flow Direction

```
&HEAD CHID='bow', TITLE='bow', /

!!! External config file

&TIME T_BEGIN = 0., T_END = 120/

&MISC TMPA = 20./

!!! Boundary condition defs

&SURF ID='Material', RGB=204,204,204, /
&SURF ID='SUPPLY', RGB=204,204,204, VOLUME_FLUX=20.0,
      COLOR='GREEN',TMP_FRONT=20., /

!!! Computational domain

&MESH ID='MESH.0', IJK=250,20,20,
      XB=-0.0,50.0,-0.5,0.5,-0.5,0.5, /
&MESH ID='MESH.1', IJK=20,20,20,
      XB=50.0,51.0,-0.5,0.5,-0.5,0.5, /
&MESH ID='MESH.2', IJK=20,250,20,
      XB=50.0,51.0,-50.5,-0.5,-0.5,0.5, /

!!! Geometry

&VENT ID='INLET', SURF_ID='SUPPLY',
      XB=0.0,0.0,-0.5,0.5,-0.5,0.5, /
&VENT ID='OUTLET', SURF_ID='OPEN',
      XB=50.0,51.0,-50.5,-50.5,-0.5,0.5, /

!!! Evacuation

!!! Control logic and output

! PRESS.0
&DEVC ID='PRESS.0', QUANTITY='PRESSURE', XYZ=50.5,-5.5,0.0, /
! PRESS.1
&DEVC ID='PRESS.1', QUANTITY='PRESSURE', XYZ=50.5,-10.5,0.0, /
! PRESS.2
&DEVC ID='PRESS.2', QUANTITY='PRESSURE', XYZ=50.5,-15.5,0.0, /
! PRESS.3
```



```
&DEVC ID='PRESS.3', QUANTITY='PRESSURE', XYZ=50.5,-20.5,0.0, /
! PRESS.4
&DEVC ID='PRESS.4', QUANTITY='PRESSURE', XYZ=30.0,0.0,0.0, /
! VEL.0
&DEVC ID='VEL.0', QUANTITY='VELOCITY', XYZ=30.0,0.0,0.0, /
! VEL.1
&DEVC ID='VEL.1', QUANTITY='VELOCITY', XYZ=30.0,0.050,0.0, /
! VEL.2
&DEVC ID='VEL.2', QUANTITY='VELOCITY', XYZ=30.0,0.1,0.0, /
! VEL.3
&DEVC ID='VEL.3', QUANTITY='VELOCITY', XYZ=30.0,0.150,0.0, /
! VEL.4
&DEVC ID='VEL.4', QUANTITY='VELOCITY', XYZ=30.0,0.2,0.0, /
! VEL.5
&DEVC ID='VEL.5', QUANTITY='VELOCITY', XYZ=30.0,0.250,0.0, /
! VEL.6
&DEVC ID='VEL.6', QUANTITY='VELOCITY', XYZ=30.0,0.3,0.0, /
! VEL.7
&DEVC ID='VEL.7', QUANTITY='VELOCITY', XYZ=30.0,0.350,0.0, /
! VEL.8
&DEVC ID='VEL.8', QUANTITY='VELOCITY', XYZ=30.0,0.4,0.0, /
! VEL.9
&DEVC ID='VEL.9', QUANTITY='VELOCITY', XYZ=30.0,0.450,-0.0, /
! VEL.010
&DEVC ID='VEL.010', QUANTITY='VELOCITY', XYZ=50.0,-5.5,0.0, /
! VEL.011
&DEVC ID='VEL.011', QUANTITY='VELOCITY', XYZ=50.050,-5.5,0.0, /
! VEL.012
&DEVC ID='VEL.012', QUANTITY='VELOCITY', XYZ=50.1,-5.5,0.0, /
! VEL.013
&DEVC ID='VEL.013', QUANTITY='VELOCITY', XYZ=50.150,-5.5,0.0, /
! VEL.014
&DEVC ID='VEL.014', QUANTITY='VELOCITY', XYZ=50.2,-5.5,0.0, /
! VEL.015
&DEVC ID='VEL.015', QUANTITY='VELOCITY', XYZ=50.250,-5.5,0.0, /
! VEL.016
&DEVC ID='VEL.016', QUANTITY='VELOCITY', XYZ=50.3,-5.5,0.0, /
! VEL.017
&DEVC ID='VEL.017', QUANTITY='VELOCITY', XYZ=50.350,-5.5,0.0, /
! VEL.018
&DEVC ID='VEL.018', QUANTITY='VELOCITY', XYZ=50.4,-5.5,0.0, /
! VEL.019
&DEVC ID='VEL.019', QUANTITY='VELOCITY', XYZ=50.450,-5.5,0.0, /
! VEL.020
&DEVC ID='VEL.020', QUANTITY='VELOCITY', XYZ=50.5,-5.5,0.0, /
```



```
! VEL.021
&DEVC ID='VEL.021', QUANTITY='VELOCITY', XYZ=50.550,-5.5,0.0, /
! VEL.022
&DEVC ID='VEL.022', QUANTITY='VELOCITY', XYZ=50.6,-5.5,0.0, /
! VEL.023
&DEVC ID='VEL.023', QUANTITY='VELOCITY', XYZ=50.650,-5.5,0.0, /
! VEL.024
&DEVC ID='VEL.024', QUANTITY='VELOCITY', XYZ=50.7,-5.5,0.0, /
! VEL.025
&DEVC ID='VEL.025', QUANTITY='VELOCITY', XYZ=50.750,-5.5,0.0, /
! VEL.026
&DEVC ID='VEL.026', QUANTITY='VELOCITY', XYZ=50.8,-5.5,0.0, /
! VEL.027
&DEVC ID='VEL.027', QUANTITY='VELOCITY', XYZ=50.850,-5.5,0.0, /
! VEL.028
&DEVC ID='VEL.028', QUANTITY='VELOCITY', XYZ=50.9,-5.5,0.0, /
! VEL.029
&DEVC ID='VEL.029', QUANTITY='VELOCITY', XYZ=50.950,-5.5,0.0, /
! VEL.030
&DEVC ID='VEL.030', QUANTITY='VELOCITY', XYZ=51.0,-5.5,0.0, /
! VEL.031
&DEVC ID='VEL.031', QUANTITY='VELOCITY', XYZ=30.0,50.0,-0.0, /

&SLCF PBX=0., QUANTITY='VELOCITY', VECTOR=.TRUE. /
&SLCF PBZ=0., QUANTITY='VELOCITY', VECTOR=.TRUE. /

&SLCF PBX=0., QUANTITY='PRESSURE', VECTOR=.TRUE. /
&SLCF PBZ=0., QUANTITY='PRESSURE', VECTOR=.TRUE. /

&TAIL /
```



A.4 FDS: Heat Source

```
&HEAD CHID='memorial', TITLE='memorial', /

!!! External config file

&TIME T_BEGIN = 0., T_END = 240/

&MISC TMPA = 20.
GVEC = -0.313787, 0.0, -9.804980
U0=2.1/

&DUMP DT_SLCF = 1.
DT_RESTART = 50. /

&MATL ID = 'CONCRETE',
CONDUCTIVITY = 1.8,
SPECIFIC_HEAT = 1.04,
DENSITY = 1440.
EMISSIVITY = 0.95/

&MATL ID = 'ALU'
CONDUCTIVITY = 235.
SPECIFIC_HEAT = 0.897
DENSITY = 27.
EMISSIVITY = 0.95/

&REAC ID = 'FIRE'
C = 16
H = 34
HEAT_OF_COMBUSTION = 41000
SOOT_YIELD=0.05/

!!! Boundary condition defs

&SURF ID='ALU', RGB=85,84,62, MATL_ID='ALU',THICKNESS=0.4,, /
&SURF ID='FAN_IN', RGB=0,204,0, VEL=-34. /
&SURF ID='FAN_OUT', RGB=0,1,204, VEL=+34, /
&SURF ID='FIRE', RGB=255,0,0, HRRPUA=6250, /
&SURF ID='IN', RGB=0,204,4, VOLUME_FLUX=-64.2, /
&SURF ID='WALL', RGB=204,204,204, TRANSPARENCY=0.311, MATL_ID='CONCRETE',THICKNESS=0.4,

!!! Computational domain
```

```

&MESH IJK=60,15,24,
      XB=-3.0,0.0,-0.0,5.2,-4.0,5.2, /
&MESH IJK=345,15,24,
      XB=0.0,345.0,-0.0,5.2,-4.0,5.2, /
&MESH IJK=60,15,24,
      XB=345.0,645.0,-0.0,5.2,-4.0,5.2, /

!!! Geometry

&VENT SURF_ID='FAN_IN', IOR=1, OUTLINE=.TRUE.,
      XB=-45.0,-45.0,2.0,3.2,1.6,2.8, /
&VENT SURF_ID='FAN_IN', IOR=1, OUTLINE=.TRUE.,
      XB=-145.0,-145.0,2.0,3.2,1.6,2.8, /
&VENT SURF_ID='FAN_OUT', IOR=-1, OUTLINE=.TRUE.,
      XB=-50.0,-50.0,2.0,3.2,1.6,2.8, /
&VENT SURF_ID='FAN_OUT', IOR=-1, OUTLINE=.TRUE.,
      XB=-150.0,-150.0,2.0,3.2,1.6,2.8, /
&VENT SURF_ID='FIRE', XB=106.0,108.0,0.0,2.0,-3.6,-3.6, /
&VENT SURF_ID='IN', IOR=1, OUTLINE=.TRUE.,
      XB=-3.0,-3.0,0.0,3.6,-3.6,2.0, /
&OBST SURF_ID='ALU', XB=-50.0,-45.0,2.0,3.2,1.6,2.8, /
&OBST SURF_ID='ALU', XB=-150.0,-145.0,2.0,3.2,1.6,2.8, /
&VENT SURF_ID='OPEN', XB=645.0,645.0,0.0,5.2,-4.0,5.2, /
&VENT SURF_ID='MIRROR', XB=-3.0,645.0,0.0,0.0,-4.0,5.2, /
! Tunnel
&OBST SURF_ID='WALL', XB=-3.1,645.1,-0.1,5.3,-4.5,-3.5, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,-0.1,5.3,4.3,5.3, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,0.9,1.9,4.1,4.3, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,1.5,1.9,3.9,4.1, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,1.9,3.1,3.7,4.3, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,2.3,3.1,3.5,3.7, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,2.5,2.7,3.3,3.5, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,2.7,3.1,3.1,3.5, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,3.1,5.3,2.7,4.3, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,3.3,3.5,2.5,2.7, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,3.5,3.7,2.3,2.7, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,3.7,4.3,1.9,2.7, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,3.9,4.3,1.5,1.9, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,4.1,4.3,0.9,1.5, /
&OBST SURF_ID='WALL', XB=-3.1,645.1,4.3,5.3,-3.5,2.7, /
&OBST SURF_ID='WALL', XB=107.1,645.1,4.1,4.3,0.7,0.9, /

```

```

!!! Evacuation

```




!!! Control logic and output

```
&DEVC ID='Loop207_GROUND', QUANTITY='VELOCITY', XYZ=0.0,0.0,-3.5, /
&DEVC ID='Loop207_GROUND.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,-3.5, /
&DEVC ID='Loop207_A', QUANTITY='VELOCITY', XYZ=0.0,0.0,-3.3, /
&DEVC ID='Loop207_A.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,-3.3, /
&DEVC ID='Loop207_B', QUANTITY='VELOCITY', XYZ=0.0,0.0,-2.4, /
&DEVC ID='Loop207_B.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,-2.4, /
&DEVC ID='Loop207_C', QUANTITY='VELOCITY', XYZ=0.0,0.0,-1.450, /
&DEVC ID='Loop207_C.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,-1.450, /
&DEVC ID='Loop207_D', QUANTITY='VELOCITY', XYZ=0.0,0.0,-0.250, /
&DEVC ID='Loop207_D.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,-0.250, /
&DEVC ID='Loop207_E', QUANTITY='VELOCITY', XYZ=0.0,0.0,0.9, /
&DEVC ID='Loop207_E.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,0.9, /
&DEVC ID='Loop207_F', QUANTITY='VELOCITY', XYZ=0.0,0.0,1.850, /
&DEVC ID='Loop207_F.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,1.850, /
&DEVC ID='Loop207_G', QUANTITY='VELOCITY', XYZ=0.0,0.0,2.6, /
&DEVC ID='Loop207_G.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,2.6, /
&DEVC ID='Loop207_H', QUANTITY='VELOCITY', XYZ=0.0,0.0,3.5, /
&DEVC ID='Loop207_H.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,3.5, /
&DEVC ID='Loop207_CEILING', QUANTITY='VELOCITY', XYZ=0.0,0.0,4.4, /
&DEVC ID='Loop207_CEILING.1', QUANTITY='TEMPERATURE', XYZ=0.0,0.0,4.4, /
&DEVC ID='Loop301_GROUND', QUANTITY='VELOCITY', XYZ=214.9,0.0,-3.5, /
&DEVC ID='Loop301_GROUND.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,-3.5, /
&DEVC ID='Loop301_A', QUANTITY='VELOCITY', XYZ=214.9,0.0,-3.3, /
&DEVC ID='Loop301_A.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,-3.3, /
&DEVC ID='Loop301_B', QUANTITY='VELOCITY', XYZ=214.9,0.0,-2.4, /
&DEVC ID='Loop301_B.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,-2.4, /
&DEVC ID='Loop301_C', QUANTITY='VELOCITY', XYZ=214.9,0.0,-1.450, /
&DEVC ID='Loop301_C.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,-1.450, /
&DEVC ID='Loop301_D', QUANTITY='VELOCITY', XYZ=214.9,0.0,-0.250, /
&DEVC ID='Loop301_D.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,-0.250, /
&DEVC ID='Loop301_E', QUANTITY='VELOCITY', XYZ=214.9,0.0,0.9, /
&DEVC ID='Loop301_E.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,0.9, /
&DEVC ID='Loop301_F', QUANTITY='VELOCITY', XYZ=214.9,0.0,1.850, /
&DEVC ID='Loop301_F.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,1.850, /
&DEVC ID='Loop301_G', QUANTITY='VELOCITY', XYZ=214.9,0.0,2.6, /
&DEVC ID='Loop301_G.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,2.6, /
&DEVC ID='Loop301_H', QUANTITY='VELOCITY', XYZ=214.9,0.0,3.5, /
&DEVC ID='Loop301_H.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,3.5, /
&DEVC ID='Loop301_CEILING', QUANTITY='VELOCITY', XYZ=214.9,0.0,4.4, /
&DEVC ID='Loop301_CEILING.1', QUANTITY='TEMPERATURE', XYZ=214.9,0.0,4.4, /
&DEVC ID='Loop302_GROUND', QUANTITY='VELOCITY', XYZ=173.1,0.0,-3.5, /
&DEVC ID='Loop302_GROUND.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,-3.5, /
&DEVC ID='Loop302_A', QUANTITY='VELOCITY', XYZ=173.1,0.0,-3.3, /
```



```
&DEVC ID='Loop302_A.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,-3.3, /
&DEVC ID='Loop302_B', QUANTITY='VELOCITY', XYZ=173.1,0.0,-2.4, /
&DEVC ID='Loop302_B.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,-2.4, /
&DEVC ID='Loop302_C', QUANTITY='VELOCITY', XYZ=173.1,0.0,-1.450, /
&DEVC ID='Loop302_C.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,-1.450, /
&DEVC ID='Loop302_D', QUANTITY='VELOCITY', XYZ=173.1,0.0,-0.250, /
&DEVC ID='Loop302_D.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,-0.250, /
&DEVC ID='Loop302_E', QUANTITY='VELOCITY', XYZ=173.1,0.0,0.9, /
&DEVC ID='Loop302_E.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,0.9, /
&DEVC ID='Loop302_F', QUANTITY='VELOCITY', XYZ=173.1,0.0,1.850, /
&DEVC ID='Loop302_F.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,1.850, /
&DEVC ID='Loop302_G', QUANTITY='VELOCITY', XYZ=173.1,0.0,2.6, /
&DEVC ID='Loop302_G.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,2.6, /
&DEVC ID='Loop302_H', QUANTITY='VELOCITY', XYZ=173.1,0.0,3.5, /
&DEVC ID='Loop302_H.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,3.5, /
&DEVC ID='Loop302_CEILING', QUANTITY='VELOCITY', XYZ=173.1,0.0,4.4, /
&DEVC ID='Loop302_CEILING.1', QUANTITY='TEMPERATURE', XYZ=173.1,0.0,4.4, /
&DEVC ID='Loop303_GROUND', QUANTITY='VELOCITY', XYZ=133.5,0.0,-3.5, /
&DEVC ID='Loop303_GROUND.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,-3.5, /
&DEVC ID='Loop303_A', QUANTITY='VELOCITY', XYZ=133.5,0.0,-3.3, /
&DEVC ID='Loop303_A.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,-3.3, /
&DEVC ID='Loop303_B', QUANTITY='VELOCITY', XYZ=133.5,0.0,-2.4, /
&DEVC ID='Loop303_B.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,-2.4, /
&DEVC ID='Loop303_C', QUANTITY='VELOCITY', XYZ=133.5,0.0,-1.450, /
&DEVC ID='Loop303_C.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,-1.450, /
&DEVC ID='Loop303_D', QUANTITY='VELOCITY', XYZ=133.5,0.0,-0.250, /
&DEVC ID='Loop303_D.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,-0.250, /
&DEVC ID='Loop303_E', QUANTITY='VELOCITY', XYZ=133.5,0.0,0.9, /
&DEVC ID='Loop303_E.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,0.9, /
&DEVC ID='Loop303_F', QUANTITY='VELOCITY', XYZ=133.5,0.0,1.850, /
&DEVC ID='Loop303_F.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,1.850, /
&DEVC ID='Loop303_G', QUANTITY='VELOCITY', XYZ=133.5,0.0,2.6, /
&DEVC ID='Loop303_G.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,2.6, /
&DEVC ID='Loop303_H', QUANTITY='VELOCITY', XYZ=133.5,0.0,3.5, /
&DEVC ID='Loop303_H.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,3.5, /
&DEVC ID='Loop303_CEILING', QUANTITY='VELOCITY', XYZ=133.5,0.0,4.4, /
&DEVC ID='Loop303_CEILING.1', QUANTITY='TEMPERATURE', XYZ=133.5,0.0,4.4, /
&DEVC ID='Loop304_GROUND', QUANTITY='VELOCITY', XYZ=119.2,0.0,-3.5, /
&DEVC ID='Loop304_GROUND.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,-3.5, /
&DEVC ID='Loop304_A', QUANTITY='VELOCITY', XYZ=119.2,0.0,-3.3, /
&DEVC ID='Loop304_A.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,-3.3, /
&DEVC ID='Loop304_B', QUANTITY='VELOCITY', XYZ=119.2,0.0,-2.4, /
&DEVC ID='Loop304_B.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,-2.4, /
&DEVC ID='Loop304_C', QUANTITY='VELOCITY', XYZ=119.2,0.0,-1.450, /
&DEVC ID='Loop304_C.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,-1.450, /
```



```
&DEVC ID='Loop304_D', QUANTITY='VELOCITY', XYZ=119.2,0.0,-0.250, /
&DEVC ID='Loop304_D.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,-0.250, /
&DEVC ID='Loop304_E', QUANTITY='VELOCITY', XYZ=119.2,0.0,0.9, /
&DEVC ID='Loop304_E.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,0.9, /
&DEVC ID='Loop304_F', QUANTITY='VELOCITY', XYZ=119.2,0.0,1.850, /
&DEVC ID='Loop304_F.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,1.850, /
&DEVC ID='Loop304_G', QUANTITY='VELOCITY', XYZ=119.2,0.0,2.6, /
&DEVC ID='Loop304_G.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,2.6, /
&DEVC ID='Loop304_H', QUANTITY='VELOCITY', XYZ=119.2,0.0,3.5, /
&DEVC ID='Loop304_H.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,3.5, /
&DEVC ID='Loop304_CEILING', QUANTITY='VELOCITY', XYZ=119.2,0.0,4.4, /
&DEVC ID='Loop304_CEILING.1', QUANTITY='TEMPERATURE', XYZ=119.2,0.0,4.4, /
&DEVC ID='Loop305_GROUND', QUANTITY='VELOCITY', XYZ=95.7,0.0,-3.5, /
&DEVC ID='Loop305_GROUND.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,-3.5, /
&DEVC ID='Loop305_A', QUANTITY='VELOCITY', XYZ=95.7,0.0,-3.3, /
&DEVC ID='Loop305_A.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,-3.3, /
&DEVC ID='Loop305_B', QUANTITY='VELOCITY', XYZ=95.7,0.0,-2.4, /
&DEVC ID='Loop305_B.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,-2.4, /
&DEVC ID='Loop305_C', QUANTITY='VELOCITY', XYZ=95.7,0.0,-1.450, /
&DEVC ID='Loop305_C.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,-1.450, /
&DEVC ID='Loop305_D', QUANTITY='VELOCITY', XYZ=95.7,0.0,-0.250, /
&DEVC ID='Loop305_D.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,-0.250, /
&DEVC ID='Loop305_E', QUANTITY='VELOCITY', XYZ=95.7,0.0,0.9, /
&DEVC ID='Loop305_E.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,0.9, /
&DEVC ID='Loop305_F', QUANTITY='VELOCITY', XYZ=95.7,0.0,1.850, /
&DEVC ID='Loop305_F.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,1.850, /
&DEVC ID='Loop305_G', QUANTITY='VELOCITY', XYZ=95.7,0.0,2.6, /
&DEVC ID='Loop305_G.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,2.6, /
&DEVC ID='Loop305_H', QUANTITY='VELOCITY', XYZ=95.7,0.0,3.5, /
&DEVC ID='Loop305_H.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,3.5, /
&DEVC ID='Loop305_CEILING', QUANTITY='VELOCITY', XYZ=95.7,0.0,4.4, /
&DEVC ID='Loop305_CEILING.1', QUANTITY='TEMPERATURE', XYZ=95.7,0.0,4.4, /
&DEVC ID='Loop306_GROUND', QUANTITY='VELOCITY', XYZ=77.7,0.0,-3.5, /
&DEVC ID='Loop306_GROUND.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,-3.5, /
&DEVC ID='Loop306_A', QUANTITY='VELOCITY', XYZ=77.7,0.0,-3.3, /
&DEVC ID='Loop306_A.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,-3.3, /
&DEVC ID='Loop306_B', QUANTITY='VELOCITY', XYZ=77.7,0.0,-2.4, /
&DEVC ID='Loop306_B.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,-2.4, /
&DEVC ID='Loop306_C', QUANTITY='VELOCITY', XYZ=77.7,0.0,-1.450, /
&DEVC ID='Loop306_C.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,-1.450, /
&DEVC ID='Loop306_D', QUANTITY='VELOCITY', XYZ=77.7,0.0,-0.250, /
&DEVC ID='Loop306_D.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,-0.250, /
&DEVC ID='Loop306_E', QUANTITY='VELOCITY', XYZ=77.7,0.0,0.9, /
&DEVC ID='Loop306_E.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,0.9, /
&DEVC ID='Loop306_F', QUANTITY='VELOCITY', XYZ=77.7,0.0,1.850, /
```



```
&DEVC ID='Loop306_F.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,1.850, /
&DEVC ID='Loop306_G', QUANTITY='VELOCITY', XYZ=77.7,0.0,2.6, /
&DEVC ID='Loop306_G.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,2.6, /
&DEVC ID='Loop306_H', QUANTITY='VELOCITY', XYZ=77.7,0.0,3.5, /
&DEVC ID='Loop306_H.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,3.5, /
&DEVC ID='Loop306_CEILING', QUANTITY='VELOCITY', XYZ=77.7,0.0,4.4, /
&DEVC ID='Loop306_CEILING.1', QUANTITY='TEMPERATURE', XYZ=77.7,0.0,4.4, /
&DEVC ID='Loop307_GROUND', QUANTITY='VELOCITY', XYZ=45.1,0.0,-3.5, /
&DEVC ID='Loop307_GROUND.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,-3.5, /
&DEVC ID='Loop307_A', QUANTITY='VELOCITY', XYZ=45.1,0.0,-3.3, /
&DEVC ID='Loop307_A.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,-3.3, /
&DEVC ID='Loop307_B', QUANTITY='VELOCITY', XYZ=45.1,0.0,-2.4, /
&DEVC ID='Loop307_B.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,-2.4, /
&DEVC ID='Loop307_C', QUANTITY='VELOCITY', XYZ=45.1,0.0,-1.450, /
&DEVC ID='Loop307_C.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,-1.450, /
&DEVC ID='Loop307_D', QUANTITY='VELOCITY', XYZ=45.1,0.0,-0.250, /
&DEVC ID='Loop307_D.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,-0.250, /
&DEVC ID='Loop307_E', QUANTITY='VELOCITY', XYZ=45.1,0.0,0.9, /
&DEVC ID='Loop307_E.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,0.9, /
&DEVC ID='Loop307_F', QUANTITY='VELOCITY', XYZ=45.1,0.0,1.850, /
&DEVC ID='Loop307_F.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,1.850, /
&DEVC ID='Loop307_G', QUANTITY='VELOCITY', XYZ=45.1,0.0,2.6, /
&DEVC ID='Loop307_G.1', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,2.6, /
&DEVC ID='Loop307_H', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,3.5, /
&DEVC ID='Loop307_H.1', QUANTITY='VELOCITY', XYZ=45.1,0.0,3.5, /
&DEVC ID='Loop307_CEILING', QUANTITY='TEMPERATURE', XYZ=45.1,0.0,4.4, /
&DEVC ID='Loop307_CEILING.1', QUANTITY='VELOCITY', XYZ=45.1,0.0,4.4, /
&SLCF FYI='(SLCF_E)', QUANTITY='extinction coefficient', PBX=0. /
&SLCF FYI='(SLCF_T)', QUANTITY='TEMPERATURE', PBX=0. /
&SLCF FYI='(SLCF_V)', QUANTITY='VELOCITY', VECTOR=.TRUE., PBX=0. /
```