BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF PROCESS ENGINEERING
ÚSTAV PROCESNÍHO INŽENÝRSTVÍ

SIMPLIFIED FLOW DISTRIBUTION MODELLING
ZJEDNODUŠENÉ MODELOVÁNÍ DISTRIBUCE TOKU

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. Miroslav Rebej**
AUTOR PRÁCE

SUPERVISOR — **Ing. Vojtěch Turek, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2019

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

Specification Master's Thesis

Department:	Institute of Process Engineering
Student:	**Bc. Miroslav Rebej**
Study programme:	Mechanical Engineering
Study field:	Process Engineering
Supervisor:	**Ing. Vojtěch Turek, Ph.D.**
Academic year:	2018/19

Pursuant to Act no. 111/1998 concerning universities and the BUT study and examination rules, you have been assigned the following topic by the institute director Master's Thesis:

Simplified flow distribution modelling

Concise characteristic of the task:

Heat transfer equipment often contains tube bundles typically consisting of hundreds to thousands of tubes. Still, for the purposes of mathematical modelling of fluid flow distribution in the bundle, the actual geometry can be somewhat simplified. The extent of employed simplification measures depends on the arrangement of the bundle, operating conditions, and so on. The resulting model, however, must be adequate for the intended application and the required data accuracy.

Goals Master's Thesis:

(1) Acquaint oneself with the issues of fluid flow distribution in tube bundles and simplified modelling of fluid flow in parallel distribution systems.
(2) Develop a computer code which will, for a given set of key flow distribution system parameters, evaluate via a chosen simplified model the resulting fluid flow distribution.
(3) Compare for several cases the obtained results with data yielded by detailed CFD models (these data will be provided) and assess to what extent the use of the chosen simplified flow distribution modelling approach is adequate in these particular cases.

Recommended bibliography:

ANSYS Inc. ANSYS Fluent Theory Guide, Release 19.1. Canonsburg, PA, USA: ANSYS Inc., 2018.

ANSYS Inc. ANSYS Fluent User’s Guide, Release 19.1. Canonsburg, PA, USA: ANSYS Inc., 2018.

PATANKAR, Suhas V. Numerical Heat Transfer and Fluid Flow. New York, NY, USA: McGraw-Hill, 1980.

Deadline for submission Master's Thesis is given by the Schedule of the Academic year 2018/19

In Brno,

L. S.

prof. Ing. Petr Stehlík, CSc., dr. h. c. Director of the Institute	doc. Ing. Jaroslav Katolický, Ph.D. FME dean

Abstract

The focus of this master's thesis is modelling of fluid flow in parallel distribution systems where the uniformity of fluid distribution plays an important role. For this purpose, a custom CFD code is presented. The code is written in Java programming language and uses third-party low-level libraries for improved performance. The code is also characterised by certain simplifications that are expected to reduce computational times. The effect of employed simplification measures is evaluated by comparing the results of flow simulations on several geometries with the data yielded by detailed CFD models. The geometries of tube bundles are distinguished by different flow and tube arrangements and also by a different number of tubes.

Keywords

flow distribution, tube bundle, flow arrangement, computational fluid dynamics, CFD, simplified CFD model

Abstrakt

Tato diplomová práce se zaměřuje na modelování proudění tekutiny v paralelních distribučních systémech, kde hraje důležitou roli rovnoměrnost distribuce tekutin. Pro tento účel je vytvořen vlastní CFD kód. Kód je napsán v programovacím jazyce Java a používá ke zlepšení výkonu knihovny třetích stran, které se vyznačují přímým přístupem ke hardwarovým a systémovým prostředkům. Kód se také vyznačuje určitými zjednodušeními, u nichž se očekává, že sníží výpočetní časy. Vliv použitých zjednodušujících opatření je vyhodnocen porovnáním výsledků simulací proudění na několika geometriích s údaji získanými z podrobných modelů CFD. Geometrie použitých svazků trubek se odlišují různými uspořádáními toku a trubek a také různým počtem trubek.

Klíčová slova

distribuce toku, trubkový svazek, uspořádání toku, výpočtová dynamika tekutin, CFD, zjednodušený CFD model

REBEJ, Miroslav. *Simplified flow distribution modelling*. Brno, 2019. Available from: https://www.vutbr.cz/studenti/zav-prace/detail/116103. Master's Thesis. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Process Engineering. Supervisor Vojtěch Turek.

I declare that I have personally elaborated the thesis " Simplified flow distribution modelling" independently, under the supervision of my master's thesis supervisor Ing. Vojtěch Turek, Ph.D., and with the use of the sources which are all listed in the bibliography section.

In Brno 24.5.2019

_____

Bc. Miroslav Rebej

I would like to thank my master's thesis supervisor Ing. Vojtěch Turek, Ph.D. for his guidance and his willingness to help at any time. I have found his expertise, advice, and friendly attitude truly helpful.

Also, great gratitude goes to my parents for giving me the opportunity to study and for their support.

Contents

Introduction

Using software tools designed for flow modelling is a common thing in many engineering fields these days. Such tools can be a great addition or even a substitute for real experiments and prototypes as they can save a lot of financial resources and time. Designing an equipment in chemical and process engineering where a fluid is engaged in almost all applications means that understanding and modelling of the fluid flow plays a very important role. Good examples of such equipment are heat exchangers consisting of many tubes. Their proper design is essential for reliability and safe operation. If the fluid flow inside the tube bundle is not as required, issues regarding, e.g., thermal expansion, overheating or fouling may occur.

Over the years, various authors came up with methods on how to analyse the distribution of fluid in such equipment. First methods were based mostly on one-dimensional equations with coefficients tailored to a specific application. The availability of computational power in recent years has opened the doors for more complex computational methods, such as the computational fluid dynamics (CFD).

CFD analyses offer some advantages compared to other computational methods being it the accuracy or applicability to a wider range of cases. The market with the CFD software has grown notably in recent years, as well. These days, there are available tools that are suitable for almost any application and each one has its pros and cons. Probably the simplest CFD tools are the ones integrated into a 3D CAD tool, e.g., the SolidWorks Flow Simulation (Dassault Systèmes, 2019) or the Autodesk CFD (Autodesk Inc., 2019). These tools are easy to use and often do not require a deep knowledge of, for example, numerical methods or discretisation schemes. Their main limitation is that they are often suitable for solving simple laminar or turbulent fluid flow problems.

Opposite of CAD-integrated tools are comprehensive and general-purpose packages, such as ANSYS Fluent (ANSYS Inc., 2018) or Star-CCM+ (Siemens PLM Software, 2019). Since these tools are specifically designed to be used for almost any application, they provide a lot of room to tweak the setup of the solver and their models. Though the user must have a certain knowledge of it to use it properly.

Another comprehensive and open-source CFD software tool that is widely used is the OpenFOAM (The OpenFOAM Foundation Ltd, 2019).

In applications where CAD-integrated tools are not suitable for the job and a comprehensive package is out of the scope due to its complexity or cost, the third group of the CFD software may find its use: special CFD tools targeted to niche markets. These tools are designed and highly optimized for specific applications. For example, the NUMECA Turbo (NUMECA International, 2019) that has its market in the turbomachinery.

The code developed in this master's thesis forms a foundation for such niche CFD tool. This model follows previous development at the Institute of Process Engineering at the Brno University of Technology where tools for evaluation of the flow uniformity in dense tube bundles were built. It uses the same approach that is common for a CFD tool, but it also incorporates certain simplifications that were found during the aforementioned development process to be acceptable for the initial design stage. The code will be presented and described with its details and limitations in this thesis. Next, tube bundles with various sizes and arrangements were selected and simulated for validation purposes. Results of those simulations are analysed and compared with simulations performed in a commercial CFD tool later in the text. Lastly, its other features regarding data visualisation and post-processing are presented and possible future work is discussed.

1 Flow analysis

In the field of chemical and process engineering, there are many applications where splitting of the fluid flow is needed. The main objective of it is to provide an increase in the fluid's surface area, which brings some benefits. For instance, improvement of the progress of chemical reactions or of the heat transfer efficiency. A common apparatus for a heat transfer process is a heat exchanger consisting of a number of tubes. Illustrative examples of such equipment where tube bundles form important parts of the construction are in Figure 1.1 and 1.2.

Figure 1.1 U-tube bundle of a box cooler for marine application (Kelvion, 2019)

Figure 1.2 Industrial condenser for refrigeration and air conditioning (Kelvion, 2019)

Proper design of such device is often essential as there are many subjects that can have a negative impact on its function, efficiency, and lifespan. One of them is the uniformity of flow distribution. This factor plays a substantial role in a reliable design. Ensuring that there is appropriate flow rate through each parallel channel is necessary in order to avoid some structural issues caused by thermal expansion or contraction. Another issue that can occur is, e.g., fouling.

1.1 Models for flow analysis

The problem of the uniformity of fluid flow has been in the interest of researchers and designers for some time. Some of the first pioneers that explored this problem were Acrivos et al. (1959). They have performed the calculation using one-dimensional flow equations for manifolds where the fluid flow is being divided or combined. The predicted distributions were also compared with observations in real experiments. Nevertheless, their approach was dealing with the phenomena of only the division or only the combination of the fluid flow. More complex systems, where both processes occur, were not considered. Similarly, Bailey (1975) dealt with the fluid outflow from a channel. In his work, the increase of static pressure for air flow through a pipe wall with holes was explored. Based on that, the static discharge coefficient for the prediction of pressure variations and air discharge is presented.

Bajura & H. Jones Jr (1976), on the other hand, studied the flow distribution in manifolds by solving differential equations. They have identified dimensionless parameters that have an effect on the flow distribution and have presented a method for evaluation of flow distribution systems.

Finally, Pigford et al. (1983) studied the complete system where both headers, the entrance header and the discharge header, are present and connected with parallel streams, i.e., tubes. They considered two flow arrangements, the parallel flow Z-manifold and the reverse flow U-manifold. Both flow arrangements will be described later in this thesis. Conclusion of their work is that, in case of the fluid being air, the distribution is more uniform in a U-manifold. Also, they present a concept for the distribution of fluid in such manifolds based on experiments.

The third possible flow arrangement using a T-manifold (again, will be described later) is explored in (Lalot et al., 1999). The importance of good flow distribution in electrical heaters is discussed there, as well. It was found that the flow distribution in T-manifolds can be very non-uniform and designing a manifold with such flow arrangement needs special attention.

Many of those presented authors used some sort of simplification in dealing with the problem of their interest that often resulted in rather limited usability and validity. Following authors used the CFD approach that allows for more detailed investigations in the field of fluid flows.

For instance, the work (Gandhi et al., 2012) focuses on the flow and pressure distribution in piping networks. CFD simulations are used to explore the distribution of flow and pressure of pure steam in assemblies composed of a header and tubes. The effect of a header design on a flow maldistribution was studied in (Kumaran et al., 2013). In that work, real experiments along with CFD simulations were performed. Yaïci et al. (2014) combined the thermal and hydraulic CFD study and examined the effect of air flow maldistribution in plate fin-and-tube heat exchangers. Other authors that used CFD for optimal flow distribution and a header design in plate-fin heat exchangers are, e.g., Zhang & Li (2003) and Wen & Li (2004). Tubular heat exchangers, on the other hand, are in the focus of (Labbadlia et al., 2017).

Flow distribution analyses were also in the interest at the Institute of Process Engineering at the Brno University of Technology. So far, some simplified quasi-1D or 2D (quasi-3D) models have been developed there as some sort of computational tools for the niche market of flow distribution analyses.

The quasi-1D model in (Chýlek, 2018) incorporates a significant extent of simplifications. This model is based on findings presented in (Bailey, 1975). The key point of this model is that the actual tube arrangement is replaced with one virtual row of tubes. This model also does not take turbulence into consideration. However, its advantage is the computational time.

Turek et al. (2011) used CFD simulations to provide a formula for the coefficient of static regain in parallel systems connected with U-tubes. A simple computational tool for shape optimization of manifolds is presented there, as well.

The continuation of the development of such tool led to a 2D model of flow distribution in dense tube bundles presented in (Turek et al., 2015). This model promised a good balance between the accuracy and the calculation times of detailed CFD simulations.

Next step was extending the model into the 3D and so the simplified 3D CFD model in (Turek et al., 2016) was developed called the Dense Tube Bundle Flow Modeller (abbreviated as DTBFMM). This model has grown into a software tool with a graphical user interface that uses the approach similar to other CFD tools. However, the current version of the in-built automatic mesher can only process certain types of tube bundles and flow arrangements. The simplified CFD code that was developed for this thesis and will be described in the following chapters uses many findings from the development of the DTBFMM. What is more, the new code is built in a way that it is able to process almost any type of flow and tube arrangement.

1.2 Methods for evaluation of flow distribution

It is obvious that the methods for flow analyses have seen significant development over the years. The information about the state of the fluid flow within the domain, i.e., knowing the pressure and velocity fields, is necessary but it may not be enough. For practical purposes, an engineer must be able to evaluate and post-process such data using specific methods or criterions.

There are various criterions to evaluate how uniformly is the fluid distributed in the tube bundle. The main objective in deciding what criterion to choose is to define whether it is better to classify the flow uniformity for each individual tube resulting with a set of values that describe the fluid distribution in it, or to have a criterion that renders a single value for the whole tube bundle. Next, it is necessary to define based on what flow property the uniformity will be evaluated. The most common properties are mass flow rates, volume flow rates, and fluid flow velocities in tubes.

When comparing the same tube bundle configuration, but with different flow properties, e.g., total mass flow rate, having a number of values for each tube can be beneficial as it can better illustrate the difference between all compared cases. This approach is often straightforward and was used in (Gandhi et al., 2012). The percentual nonuniformity, E_t, is calculated using equation (1.1).

$$E_t = \frac{\dot{m}_i - \dot{m}_{avg}}{\dot{m}_{avg}} \cdot 100 \tag{1.1}$$

Where $\dot{m}_i$ is the mass flow rate in the *i*-th tube, and $\dot{m}_{avg}$ is the average mass flow rate in tubes.

When it is necessary to compare distribution systems with a different number of tubes, having a single value that describes the uniformity of the fluid distribution is better. However, there are numerous methods on how to evaluate such case in the literature.

To name a few, there is the division criterion, DC, described in (Bava & Furbo, 2016) that uses dimensionless flow rates, $\dot{V}_i'$. The division criterion is in equation (1.2) and the dimensionless flow rates are in equation (1.3). Zhang & Li (2003) used in their work a criterion of flow nonuniformity based on the deviation of fluid velocities in individual tubes from average velocity, S, equation (1.4). Another criterion of flow nonuniformity in equation (1.5) is based on velocities, too. It is used in (Jiao et al., 2003). On the other hand, the criterion that was used in (Wei et al., 2016), called the deviation factor, DF, used a deviation of a tube mass flow rate from a value of mass flow rate at ideal distribution, equation (1.6).

$$\mathrm{DC} = \frac{\max_i\{\dot{V}_i'\}}{\min_i\{\dot{V}_i'\}} \tag{1.2}$$

$$\dot{V}_i' = \frac{\dot{V}_i \cdot N}{\sum_{i=1}^{N} \dot{V}_i} \tag{1.3}$$

$$S = \sum_{i=1}^{N} \left| \frac{u_i - u_{avg}}{u_{avg}} \right| \tag{1.4}$$

$$S' = \left[\frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{u_i - u_{avg}}{u_{avg}} \right)^2 \right]^{1/2} \tag{1.5}$$

$$\mathrm{DF} = \left[\frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{\dot{m}_i - \dot{m}_{id}}{\dot{m}_{id}} \right)^2 \right]^{1/2} \tag{1.6}$$

Where $\dot{m}_i$ is the mass flow rate in the *i*-th tube, $\dot{m}_{id}$ is a mass flow rate at the uniform flow distribution, $\dot{V}_i$ is the volume flow rate in the *i*-th tube, u_i is the velocity of fluid in the *i*-th branch (tube), u_{avg} is the average velocity of fluid in a branch (tube), and N is the total number of tubes.

Next criterion in equation (1.7), called the relative standard deviation, δ (later denoted as RSD), describes the flow nonuniformity with a variation coefficient in the percentual form. The concept of this criterion is based on the comparison of mass flow rates in individual tubes with a mass flow rate at ideal distribution for each tube. This criterion was used in (Anbumeenakshi & Thansekhar, 2016) as well as in (Fialová, 2017). To better compare the results of this work with the work of Fialová (2017), the RSD will be used later in this thesis.

$$\delta = \frac{100}{\dot{m}_{id}} \left[\frac{1}{N} \sum_{i=1}^{N} (\dot{m}_i - \dot{m}_{id})^2 \right]^{1/2} \tag{1.7}$$

Where $\dot{m}_i$ is the mass flow rate in the i-th tube, $\dot{m}_{id}$, is mass flow rate in a tube at ideal distribution of the fluid, and N is the number of tubes.

2 The CFD code

The proposed computer code is written using the Java SE Development Kit 11 (Oracle, 2019). The main reason for selecting this programming platform was based on the previous work in the development of software for simulations of flow distribution systems containing dense tube bundles at the Institute of Process Engineering at Faculty of Mechanical Engineering in Brno. Next, the availability of additional libraries such as Parallel Colt (Wendykier & Nagy, 2010) and Apache Commons Math (The Apache Software Foundation, 2016) was found very useful. They provide a very fast and efficient way of performing all numerical operations by directly accessing the system and hardware resources and for this reason are extensively used in the code. Another reason for selecting Java was also the fact that it is an object-oriented programming language. This plays an important role in the code's architecture.

The fundamental part of a Java program is a class. It is where all program activity occurs and where data and code are defined (Schildt, 2014). Therefore, the code for this diploma thesis is split into two Java classes, the `Main` class and the `Mesh` class.

The `Main` class is the core of this code and it is the one that is executed as the code is run. It contains commands that are relevant for problem initialization, building and solving discretized equations, and exporting results. It is where modelling of physical phenomena takes place. In contrast to that, the `Mesh` class is used as a template for constructing components for a required tube bundle configuration. The `Mesh` class is an extension of the `Main` class and it is where modelling of a physical domain takes place. This class is called from the `Main` class each time constructing of a new mesh object is needed.

Although Java has tools and components to build a user interface, the code has not got any, yet. The main goal was to develop and test the numerical model. Thus, if the user is interested in using the code, the knowledge of the Java programming language is necessary as all input is provided by modifying the code itself. Likewise, all necessary outputs are written to the command window in the Integrated Development Environment or to text files. Currently, residuals are written to both but calculated mass flow rates in tubes are written only into a text file.

2.1 Modelling in relation to the physical domain

In order to carry out a simulation of a physical phenomenon, a representation of the physical space where such phenomenon occurs must be prepared. The physical domain is represented by a specified number of mesh cells, i.e., discrete elements, that together form a mesh, on which discretized equations are later solved. Fundamentally, these mesh cells are arrays of data that store the information about the fluid in the specific location within the domain. When needed, the data is accessed through a local or global cell identification index.

At the beginning of constructing a virtual model in the proposed code, the geometry must be broken down into parts of a separate and simplified shape. In the current stage of the code, it is up to the user to identify parts of the apparatus where the simplification is possible and to modify the code accordingly so that the geometry is virtually broken down into those parts. These separate and simplified parts of the actual physical domain are constructed as instances of the `Mesh` class and further subdivided into mesh.

The process of breaking down the computational domain into separate parts makes generating and handling mesh cells later straightforward in relation to the topology of the object, i.e., local indexes are used in the discretization process to determine object's boundaries, geometric information of the cell elements, or element to element relations. Global indexes, on the other hand, are used to build sets of discretized equations over the computational domain. These steps are based on the approach described in (Moukalled et al., 2016) and provide a simplification in iterating through the physical domain in stages described later.

All problems that are in the interest of this diploma thesis are composed of a number of elementary objects (Figure 2.1) that are always constructed in the following order:

- Distributor `Object id = 0`
- Collector `Object id = 1`
- Inlet region `Object id = 2`
- Outlet region `Object id = 3`
- Tube(s) `Object id = 4`
  (incremented for each tube in case of multiple tubes)

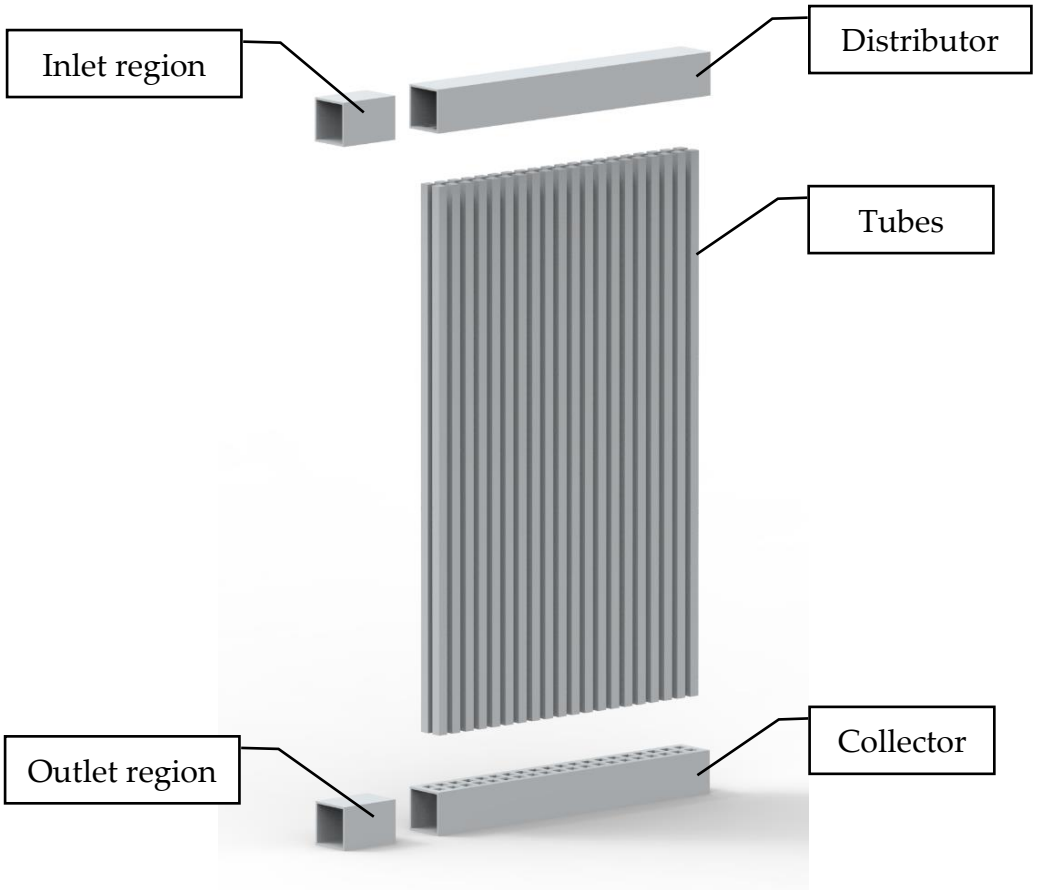

Figure 2.1 Exploded view of one of the considered geometries

Each object is constructed with a set of the initial information that defines its shape (x, y, and z dimension), spatial discretization (x, y, and z divisions), and `Object id`. The `Object id` is a unique value that is passed into the class's constructor. Its purpose is to help identify objects in subsequent operations. Values for dimensions and spatial discretization are set by the user to describe the actual part of the tube bundle configuration and to meet the desired mesh resolution. Example values for one of the considered geometries are in Table 2.1.

The constructor of an object in the `Mesh` class then iterates through the defined boundaries and places new items in arrays where data for control volumes are stored, e.g., a cell's x, y, and z coordinate, or local and global position indexes. These and following steps are considered as generating of control volumes. Based on the number of generated cells, additional arrays are created for storing the type of a boundary condition, information about neighbouring cells, or for storing values of physical properties of the fluid. These arrays are filled during initialization with data using specific methods in the code after all geometry objects are constructed.

Table 2.1 Example of initial information for the `Mesh` class for one of the considered geometries

Object	`Object id`	Direction	Dimension, mm	Number of divisions
Distributor	0	`X`	320	104
		`Y`	40	16
		`Z`	40	14
Collector	1	`X`	320	104
		`Y`	40	16
		`Z`	40	14
Inlet region	2	`X`	60	30
		`Y`	40	16
		`Z`	40	14
Outlet region	3	`X`	913.6	65
		`Y`	40	16
		`Z`	40	14
Tube	4+	`X`	8.862	3
		`Y`	2,000	36
		`Z`	8.862	3

One of the simplifications of this code is that control volumes are always cuboids with 6 orthogonal faces designated as S (south), N (north), W (west), E (east), B (bottom), and T (top). The illustration of a control volume with face labels is in Figure 2.2.

Orthogonality of faces simplifies algorithms where discretized equations are built and filled into matrices as the number of conditions on cell faces that needs to be handled is always the same. Besides, this renders another simplification in the calculation of gradients of fluxes since they are always perpendicular to faces of a control volume. Therefore, there are no complex and computationally expensive calculations needed.

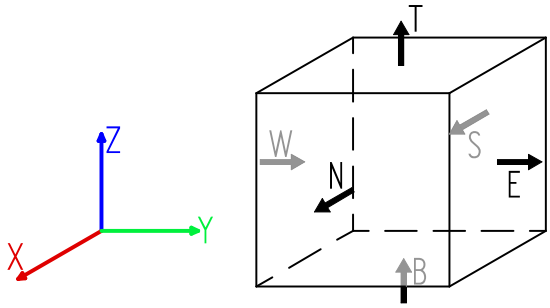

Figure 2.2 Control volume and orientation of its faces

Tubes with a circular cross-sectional are constructed in the same manner, i.e., by cuboids. But, in order to simulate the phenomena inside tubes, the meshing process of tubes is performed in a way that the cross-sectional area is the same. Therefore, the edge length of a tube is adjusted based on the equation (2.1):

$$a = \sqrt{\frac{\pi}{4} d^2} \tag{2.1}$$

In this equation, a is the edge size of the rectangular tube representation and d is the inner diameter of the tube. Tubes in all considered cases have a diameter of 10 mm and consist of 3 × 3 cells in cross-section. The tube of 10 mm in diameter is, therefore, modelled by a square cross-section area with the edge length of 8,862 mm. The third dimension of a cell in the first layer of a tube in the longitudinal direction is set to be the same as in a neighbouring cell in a manifold. For cells in the next layers, a growth factor is used to elongate cells towards the middle of the tube. Values for this growth factor are used in the range from 1.39 up to 1.405, to keep the number of cells in the longitudinal direction between 20 to 25. The spatial discretisation of a tube is illustrated in Figure 2.3.

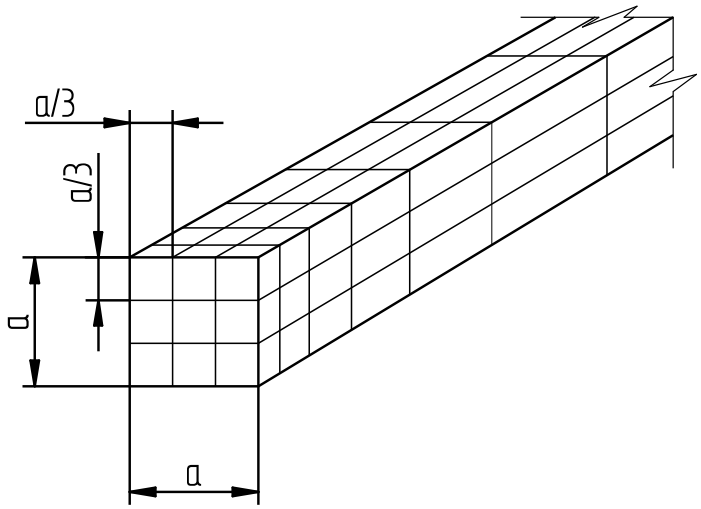

Figure 2.3 Spatial discretization of a tube

In order to avoid numerical instability, sizes of cells in other, adjacent, mesh objects must be similar to these cells. This way, the mesh conformity is preserved, and no interpolation of values is needed. To illustrate how this affects the meshing process, figures 2.4 and 2.5 show the distributor (or collector) of cases with 90° inline and 45° staggered tube arrangement, respectively. In both figures, grey areas mark cell faces to which the tubes are connected to.

The type and position of boundary conditions must be defined inside the family of `set_BC` methods within the `Mesh` class. However, those methods are called from the `Main` class, immediately after constructing of all objects. The code allows for following boundary conditions:

- Wall (stationary, no slip)
- Mass flow inlet
- Pressure outlet
- Internal (domain)

In addition to that, there is also a temporary boundary condition called the domain connection. This boundary condition is used to pair a face from one object of the tube bundle to a face belonging to another object by setting a unique negative value to the necessary pair of faces. Once all matching pairs are specified, this condition is replaced with the Internal (domain) boundary condition with respect to assigned values.

After all boundary conditions are set, local lists with a local control volume data from individual objects are concatenated into global lists. This is considered as a transition from local coordinate systems into one global coordinate system where the global index of a control volume is the only coordinate. As mentioned before, this step simplifies and speeds up algorithms where discretized equations are built and filled into matrices as it allows to iterate through a single one-dimensional array of global indexes ranging from 0 to the value equal to the number of cells in the domain.

All of this means that the mesh resolution in the distributor and collector is given mainly by the number of cells in the cross-section of tubes. Mesh sizes in reviewed cases are ranging from around 15 000 up to almost 70 000 cells. Since the 3 × 3 resolution of tubes was seen as the reasonable number of cells for the initial stage of the design process, no higher (nor lower) resolutions were considered. Any higher resolution would likely result in better accuracy, but this resolution was mainly selected based on the results presented in (Turek et al., 2016). The 3 × 3 resolution was found to offer the best balance between the accuracy and total mesh size. Thus, there was no grid independence study performed.

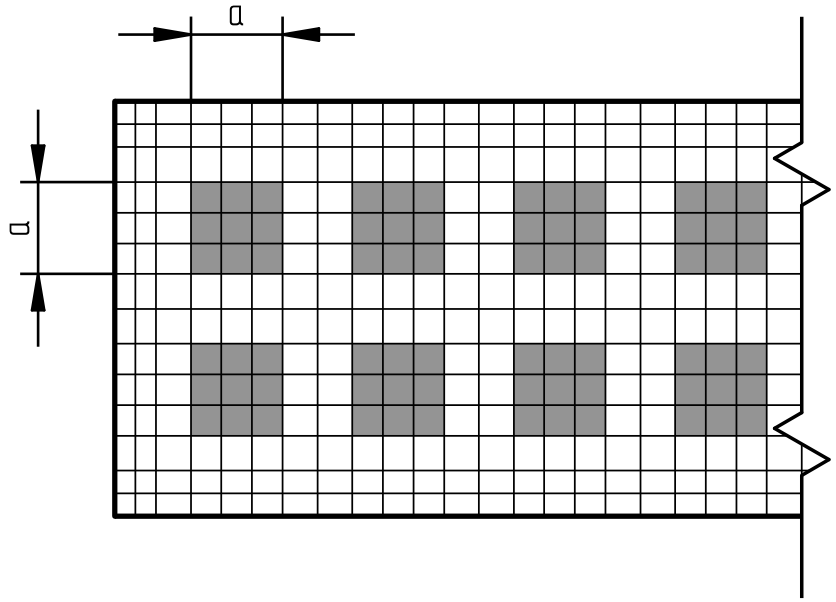

Figure 2.4 Spatial discretization of a tubesheet with 90° inline tube arrangement

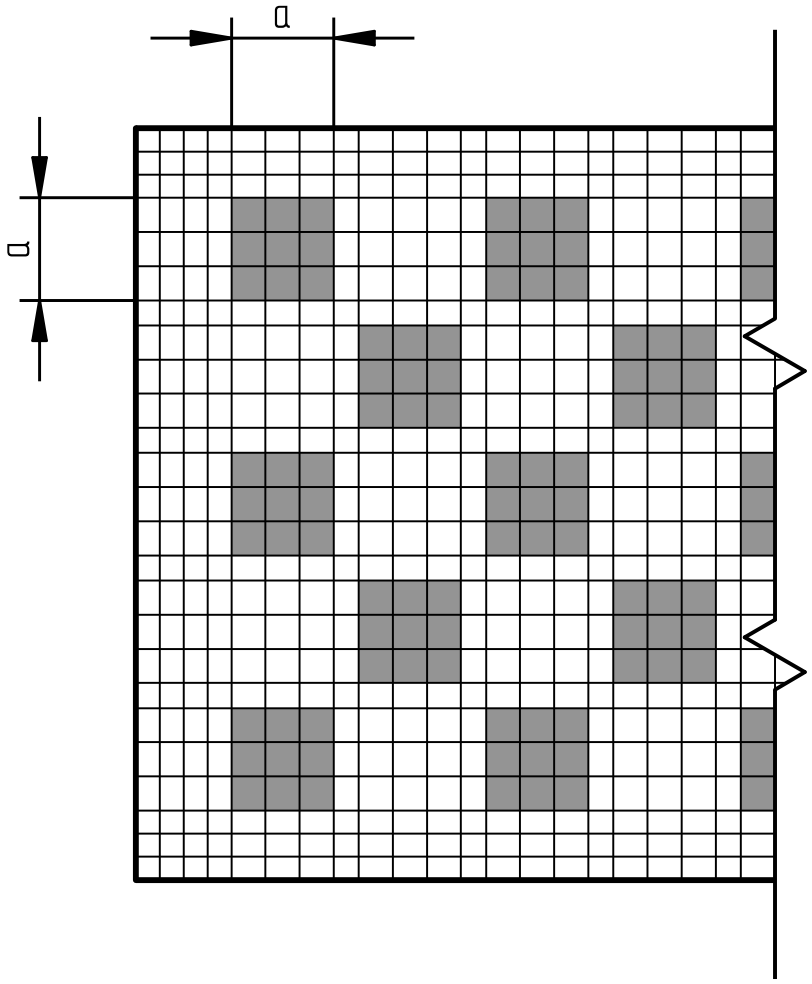

Figure 2.5 Spatial discretization of a tubesheet with 45° staggered tube arrangement

At last, the `Mesh` class also contains some methods for printing information about neighbouring cells and boundary conditions. These methods can be helpful in inspecting the geometry and visualising the position of boundary conditions in case of some problems. In addition to that, there are also methods for exporting results for post-processing. These functionalities will be briefly presented at the end of this thesis.

2.2 Modelling in relation to the physical phenomena

Equations (2.2) to (2.8) in Table 2.2 govern the flow of an incompressible Newtonian fluid.

Table 2.2 Governing equations (Versteeg & Malalasekera, 1995)

Mass	$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0$	(2.2)
Momentum	$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho u \boldsymbol{u}) = -\frac{\partial p}{\partial x} + \nabla \cdot (\mu \, \nabla u) + S_{Mx}$	(2.3)
	$\frac{\partial (\rho v)}{\partial t} + \nabla \cdot (\rho v \boldsymbol{u}) = -\frac{\partial p}{\partial y} + \nabla \cdot (\mu \, \nabla v) + S_{My}$	(2.4)
	$\frac{\partial (\rho w)}{\partial t} + \nabla \cdot (\rho w \boldsymbol{u}) = -\frac{\partial p}{\partial z} + \nabla \cdot (\mu \, \nabla w) + S_{Mz}$	(2.5)
Energy	$\frac{\partial (\rho T)}{\partial t} + \nabla \cdot (\rho T \boldsymbol{u}) = \nabla \cdot \left(\frac{\lambda}{c_P} \, \nabla T \right) + \frac{S_{\mathrm{En}}}{c_P}$	(2.6)
Turbulent kinetic energy	$\frac{\partial (\rho k)}{\partial t} + \nabla \cdot (\rho k \boldsymbol{u}) = \nabla \cdot \left(\mu_{\mathrm{eff},k} \, \nabla k \right) + S_k$	(2.7)
Turbulent energy dissipation rate	$\frac{\partial (\rho \varepsilon)}{\partial t} + \nabla \cdot (\rho \varepsilon \boldsymbol{u}) = \nabla \cdot \left(\mu_{\mathrm{eff},\varepsilon} \, \nabla \varepsilon \right) + S_\varepsilon$	(2.8)

Where the physical properties of the fluid are given by the density ρ, the dynamic viscosity μ, the specific heat capacity c_P, the thermal conductivity λ, the temperature T, and pressure p. The $\mu_{eff,k}$ and $\mu_{eff,\varepsilon}$ are effective viscosities for the turbulent kinetic energy k, and the turbulent energy dissipation rate ε, respectively. Next, u, v, and w is the x, y, and z-component of the velocity vector field, $\boldsymbol{u}$. S_{Mx}, S_{My}, and S_{Mz} are momentum source terms for all three directions. Other source terms are the energy source term S_{En}, the turbulent kinetic energy source term S_k, and the turbulent energy dissipation rate S_ε.

The code solves equations governing the phenomena at discrete points defined by the mesh, rather than obtaining a solution from continuous equations defined in Table 2.2. Therefore, discretization methods are employed. Outcomes of such methods are algebraic discretized equations involving the unknown values of a transport property variable at points defined by the mesh.

There are various discretization methods available. To name a few, in (Patankar, 1980) is listed the Central difference scheme, the Upwind scheme, and the Power-law scheme. Also, there is the Hybrid scheme of Launder & Spalding (1972) or the QUICK scheme of Leonard (1979). Other higher order schemes can be found listed in e.g., (Moukalled et al., 2016). But as presented in (Turek et al., 2016), employing a first-order scheme, such as the Power-law, is expected to be sufficient in simplified calculations equivalent to the ones in this thesis. Mainly due to a low mesh resolution. Similarly, the k-ε model with standard wall function is implemented based on its computational and programming simplicity.

Based on the previous work done by Turek (2018) following CFD model parameters are used:

- Pressure-based solver, double-precision floating-point format
- The Power-law discretization scheme
- The SIMPLEC algorithm for pressure-velocity coupling (van Doormaal & Raithby, 1984)
- 10^{-3} scaled residual limit for all transport variables
- Conjugate gradient (CG) matrix solver for the pressure correction equation, BiConjugate Gradient Stabilized (BiCGstab) solver for other transport variables

Both matrix solvers come from the Parallel Colt library that is imported into the code. However, these solvers have been modified to return the current solution vector in case of a breakdown (common especially at the beginning of the CFD solution process) or when the inner iteration limit is exceeded. In other words, the solution would not often be obtained at all without this enhancement.

Moreover, it is possible to apply under-relaxation to momentum, energy, and both turbulence equations in the code. In the pressure correction equation, there is no under-relaxation as it is removed in the SIMPLEC variant (van Doormaal & Raithby, 1984). The value of 0.35 was chosen for under-relaxation in momentum equations. The energy equation was not enabled, so there was no under-relaxation applied. In case of turbulence equations, under-relaxation of 0.4 was used. Likewise, there is also a possibility to turn off solving of the energy equation and the turbulence model by setting their respective state variables to the `false` value. Furthermore, it is possible to perform steady-state calculations, as well as transient ones.

After the creation of the mesh, the code proceeds to the loop where the solution of the physical phenomena occurs. First, momentum equations for x, y, and z are solved producing the velocity field. Then the continuity equation in the form of the pressure correction equation is solved. Next, pressure and velocity fields are updated. These steps follow the procedure presented in (Versteeg & Malalasekera, 1995). After obtaining the corrected velocity and pressure fields, equations for remaining scalar variables are built and solved in the following order: energy, turbulent kinetic energy, and turbulent energy dissipation. The programming of discretization of these equations is based on (Moukalled et al., 2016). To better understand the order of processes and the data flow, the block diagram for the simplified CFD code is provided in Figure 2.6. Also, the block diagram of the SIMPLEC algorithm structure is in Figure 2.7.

All velocity, turbulence, and energy fields in all problems are initialized with inlet values, the gauge pressure field is initialized with 0 Pa, i.e., the value of the reference pressure level. This way of initialization was found during the testing of the code to produce the fastest convergence. Tolerances for CG and BiCGstab solvers of linearized equations are set as follows:

- Divergence tolerance 1e16
- Absolute tolerance 1e-16
- Relative tolerance

Momentum	$\min\left\{10^{-3}, \max\left\{\frac{velocity\ residual}{100}, 10^{-10}\right\}\right\}$
Pressure	$\max\left\{\frac{pressure\ residual}{100}, 10^{-10}\right\}$
Energy	$\min\left\{10^{-4}, \max\left\{\frac{energy\ residual}{100}, 10^{-10}\right\}\right\}$
Turbulent kinetic energy	$\min\left\{10^{-3}, \max\left\{\frac{k\ residual}{100}, 10^{-10}\right\}\right\}$
Turbulent dissipation rate	$\min\left\{10^{-3}, \max\left\{\frac{\varepsilon\ residual}{100}, 10^{-10}\right\}\right\}$

Where $velocity\,residual$ are levels of residuals obtained after solving the discretised momentum equation for each direction. Similarly, $pressure$, $energy$, k, and $\varepsilon\ residual$ are levels of residuals obtained after solving the corresponding discretised equation.

After obtaining a converged solution in each timestep, values of residuals and mass flow rates in each individual tube are also written into respective text files.

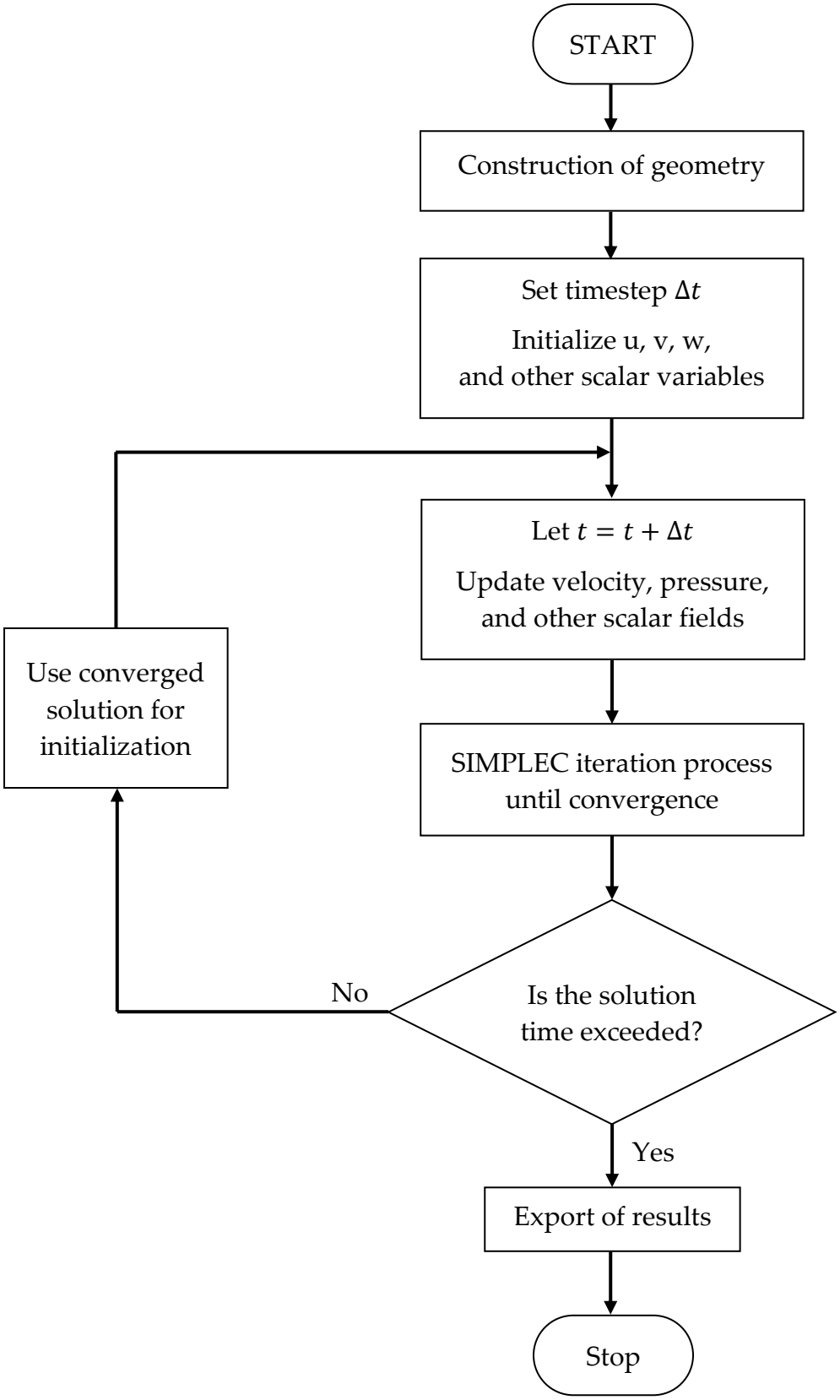

Figure 2.6 Block diagram of the simplified CFD code

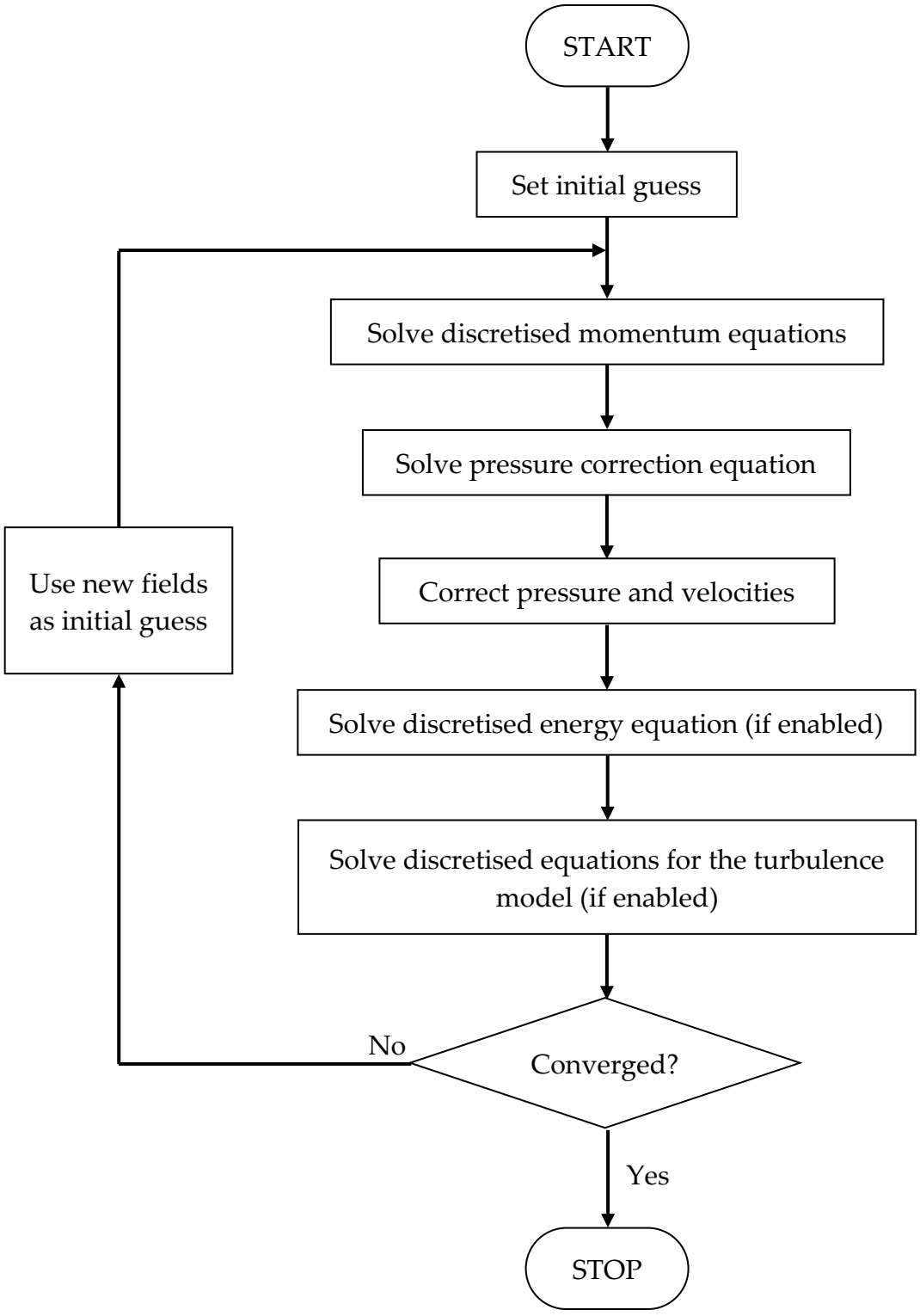

Figure 2.7 Block diagram of the SIMPLEC algorithm

2.3 Physical properties of the fluid

An important part of the fluid flow simulation is the fluid itself. For purposes of this work, water at the temperature of 300 K was used.

The code does not have any built-in material database so the user must provide the data for physical properties of the fluid that is in the interest of flow analysis. Next, the user must define a temperature range for which the data is valid. Based on this approach, it is possible to simulate the flow of any fluid.

Essentially, the code needs lists of values for density and dynamic viscosity valid for a required temperature range. In case of solving the energy equation, two additional lists are needed, one for thermal conductivity and second for specific heat capacity.

The data covering fluid physical properties are interpolated for every mesh cell using the temperature in that cell. Linear interpolation is used to quickly obtain values for physical properties of the fluid in the cell. This procedure happens at the initialization stage, when cells are filled with the information about the fluid, and at the end of each timestep in case of a transient simulation when the information about the fluid in cells is being updated.

Currently, there are lists with values of physical properties for water taken from (ThermExcel, 2014). This set of data offers values for density, dynamic viscosity, and specific heat capacity for water with temperature from 273.15 K up to 373.15 K with 1 K resolution. The data for thermal conductivity was taken from (Perry, 1998).

Data sets of physical properties already implemented in the code are plotted in the following figures, Figure 2.8 to 2.11.

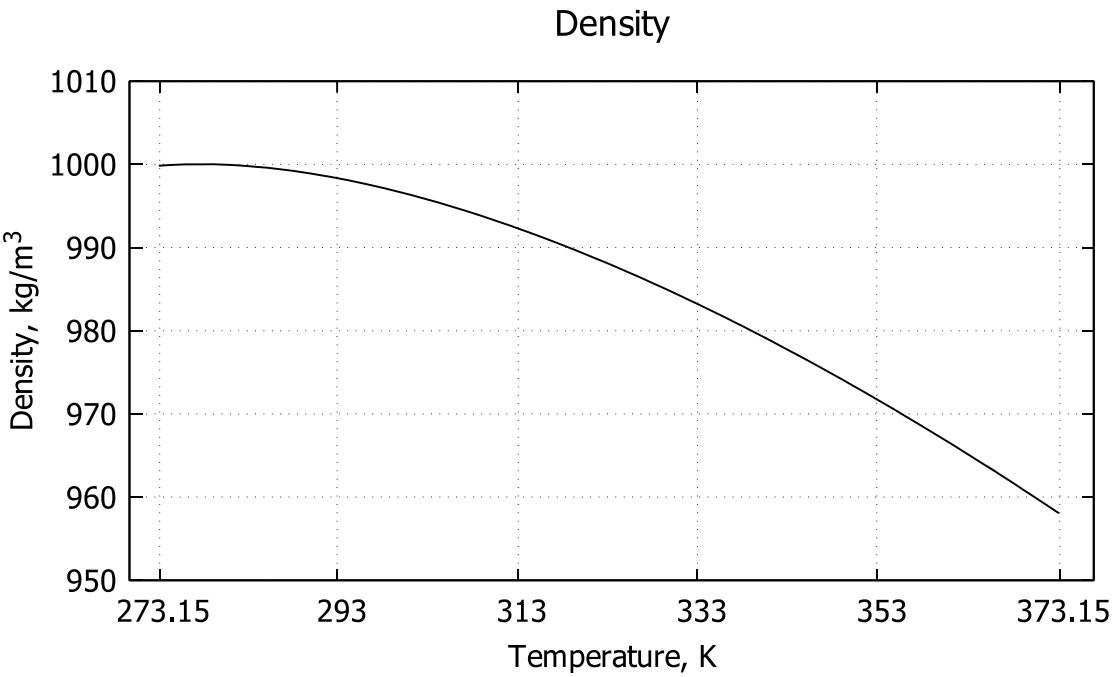

Figure 2.8 Density

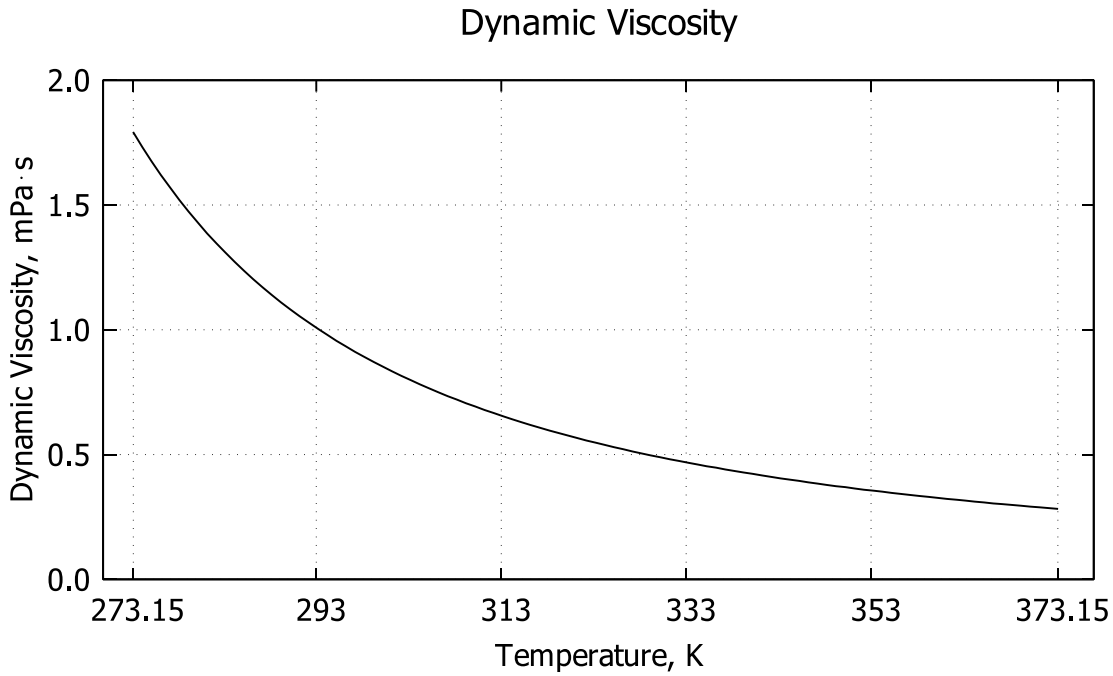

Figure 2.9 Dynamic viscosity

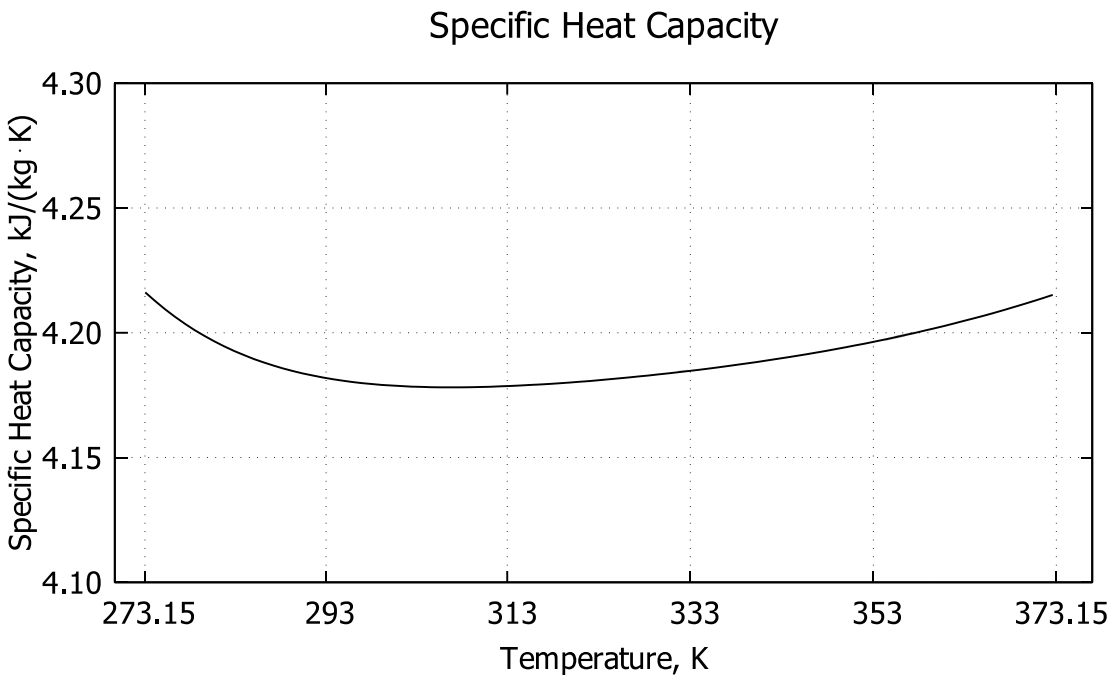

Figure 2.10 Specific heat capacity

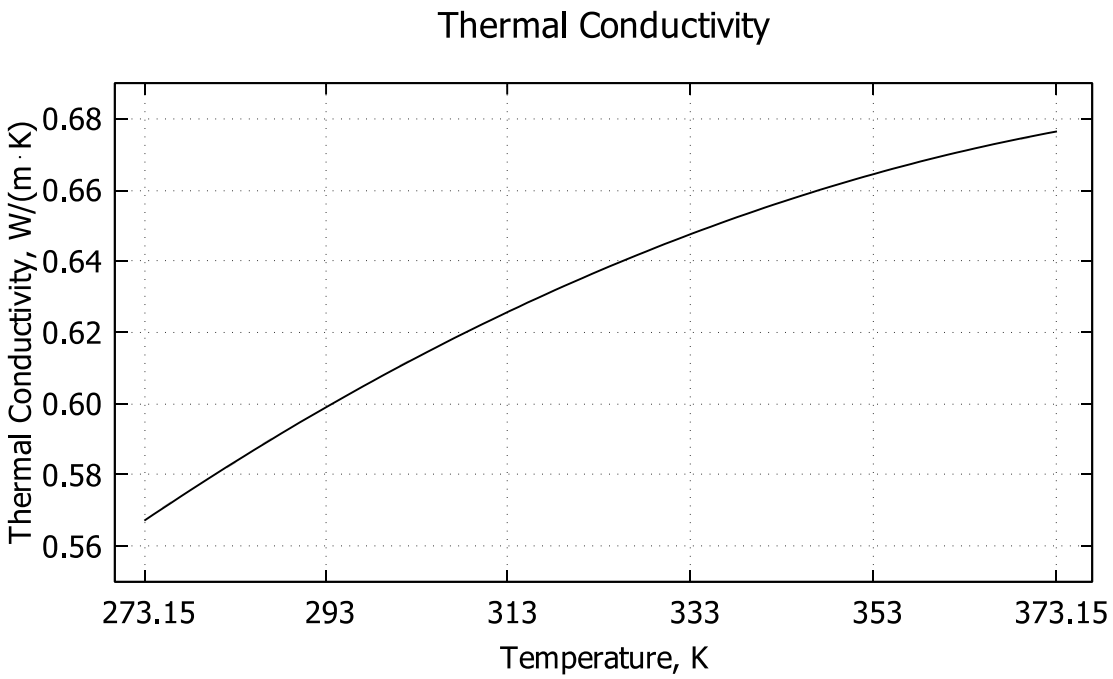

Figure 2.11 Thermal conductivity

3 Model validation

The results that come from the CFD code introduced before will be in this chapter compared with the data from detailed CFD analyses. Simulation results presented in (Fialová, 2017) are used for this purpose. Besides these simulations, there is also one extra case where a different type of tube bundle is used. The point is to prove that the computational part of the code is built to be universal in terms of handling various shapes and configurations of objects, provided that the user is able to define such mesh using programming statements. It is also important to note that both fluid flow simulations of this extra case were carried out by the author of this diploma thesis. Again, the ANSYS Fluent was used for the detailed analysis.

3.1 Tube bundle configurations of interest

Configurations of tube bundles that are in the interest of this diploma thesis, along with a naming scheme, are taken from (Fialová, 2017). That thesis investigates the effect of geometrical parameters such as the tube layout or the overall configuration of a tube bundle on fluid flow distribution.

All configurations are categorized into three main flow arrangements: U, Z, and T (Figure 3.1). The difference between them is in the position of inlets and/or outlets. There are also some variations in dimensions of three key components: distributor, collector, and the tube bundle. This way, it is possible to accommodate a different number of tubes.

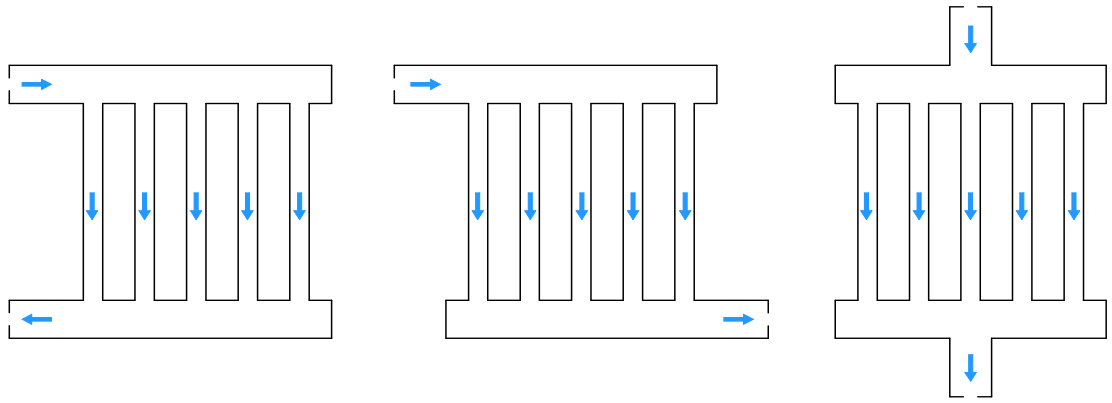

Figure 3.1 Flow arrangements

Generally, the fluid enters the distributor with a defined mass flow rate and flows through tubes to the collector where the separation of the fluid flow occurs. The fluid flow then combines in the collector and leaves the apparatus through the outlet.

Apart from different position of inlets and/or outlets, the tube arrangement is being changed, as well. There are three basic tube arrangements, shown in Figure 3.2:

- 90° inline
- 45° staggered
- 60° staggered

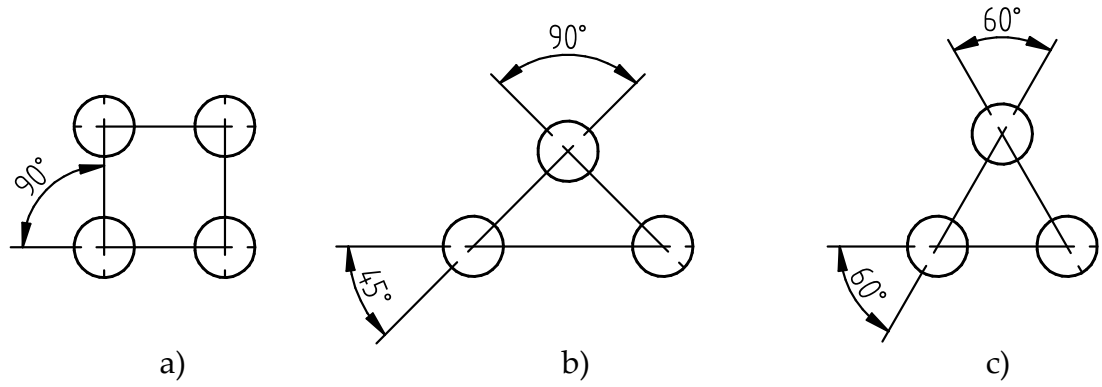

Figure 3.2 Tube arrangements
a) 90° inline, b) 45° staggered, c) 60° staggered

All simulations were performed in a custom CFD code and will be described in the next chapter. The results are later compared with detailed CFD analyses using ANSYS Fluent.

Fialová (2017) also works with 60° staggered configuration in cases ID03 and ID05. However, this tube arrangement is difficult to model with the presented CFD code at this moment. The reason is that the tube spacing in these two cases is too small

eventually producing overlapping mesh cells. Therefore, this type of staggered configuration would need to be composed of cells with skewed faces. That would violate one of the essential simplifications assumed in the code so those cases were ignored. Nevertheless, if the tube spacing was large enough to avoid that problem, 60° staggered tube arrangement could be modelled.

The list of all considered arrangements, together with their characteristic parameters, is in Table 3.1 where the following nomenclature is used:

- W is the width, L is the length, and H is the height of main channels
- L_{in} and L_{out} is the length of inlet and outlet regions, respectively
- R_L and S_L is the number of tube rows and their spacing in the longitudinal direction of channels
- R_W and S_W is the number of tube rows and their spacing in the lateral direction of channels

Tubes in all cases are 2,000 mm long with a diameter of 10 mm. The nomenclature is shown in Figure 3.3 and 3.4.

Table 3.1 Parameters of considered geometries (ID0x)

Case		Stg*	W	L	H	L_{in}	L_{out}	R_L	R_W	S_L	S_W	Number of tubes
ID01	U Z T	90°	20	320	40	60	913.6 913.6 983.5	20	1	15.6	15.6	20
ID02	U Z T	90°	40	320	40	60	913.6 913.6 983.5	20	2	15.6	15.6	40
ID04	U Z T	90°	55	320	55	70	913.6 913.6 1,283.5	20	3	15.6	15.6	60
ID06	U Z T	45°	65	235	70	70	883.9 883.9 1,222.4	10	5	11.03	22.06	50
ID07	U Z	90°	70	320	70	70	913.6 913.6	20	4	15.6	15.6	80

*Staggered tube layout

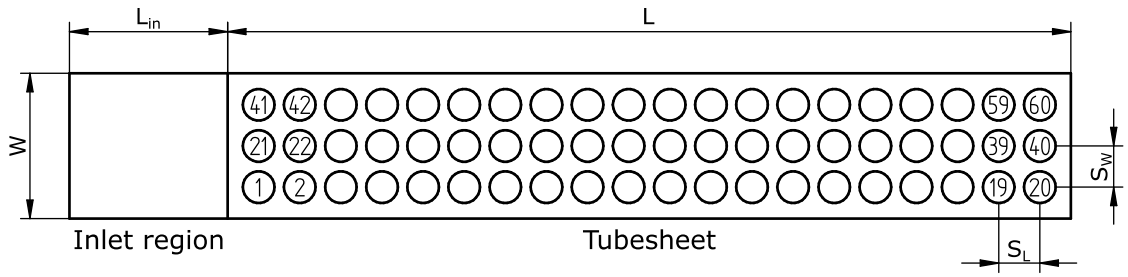

Figure 3.3 Geometric parameters of ID04

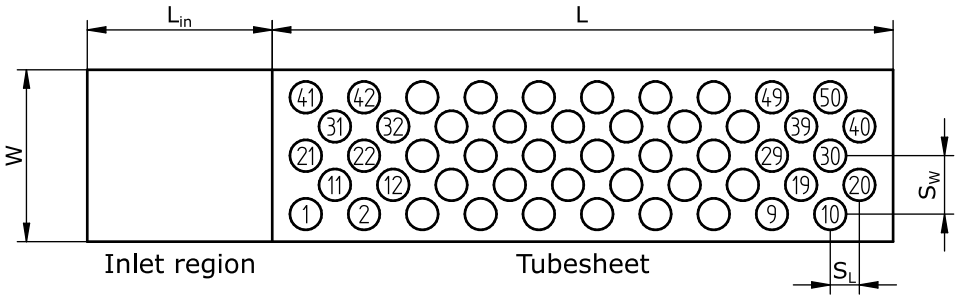

Figure 3.4 Geometric parameters of ID06

Furthermore, to illustrate the meshing capability and the versatility of the computational part of the proposed CFD code, a simulation of the tube side of a shell & tube heat exchanger were performed, as well. The detailed information on dimensions of this case is in Table 3.2 where the nomenclature is used as follows:

- D is the diameter and L is the length of both manifolds, the distributor and the collector
- D_{in} is the diameter that circumscribes the tube bundle on tubesheets
- W_1 and W_2 are dimensions of cross-sections of inlet and outlet regions
- L_{in} and L_{out} is the length of inlet and outlet regions, respectively (Measured from the centre of the tubesheet)
- S_L and S_W are spacings of tubes in two orthogonal directions

Tubes are again 2,000 mm long with a diameter of 10 mm. Since the tubesheet is of a circular shape, the number of tube rows in both orthogonal directions vary from 7 to 21. Sketch where these parameters are shown is in Figure 3.5. Detailed view on the tubesheet is in Figure 3.6.

Table 3.2 Parameters of the S&T geometry

Case	Stg*	D	D_{in}	L	W_1	W_2	L_{in}	L_{out}	S_L	S_W	Number of tubes
S&T	90°	400	360	160	100	100	360	360	15.6	15.6	349

*Staggered tube layout

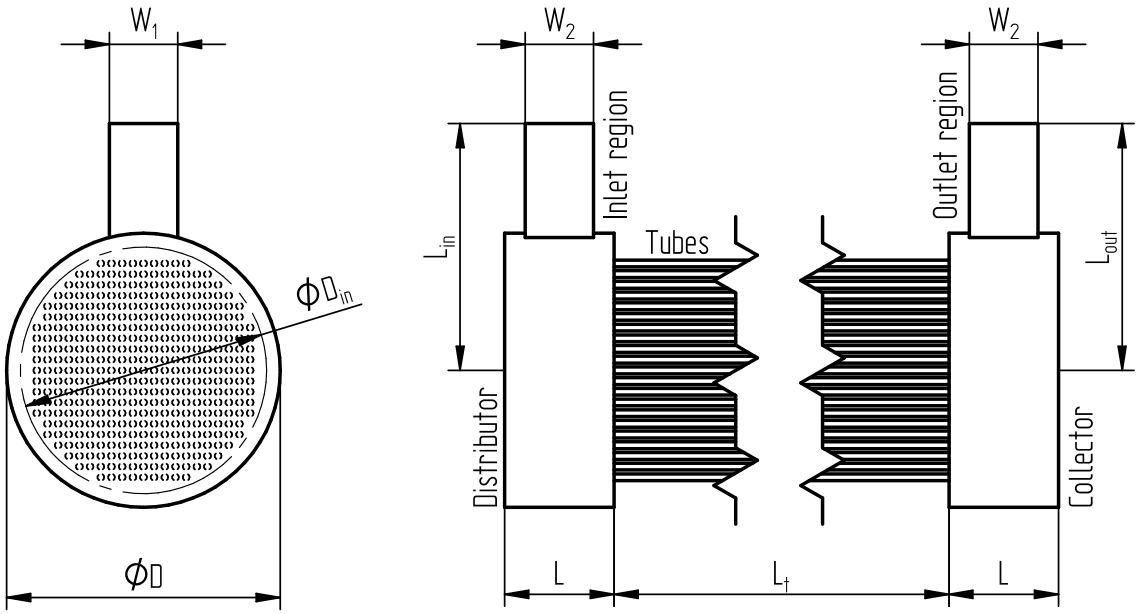

Figure 3.5 Geometric parameters of the S&T case

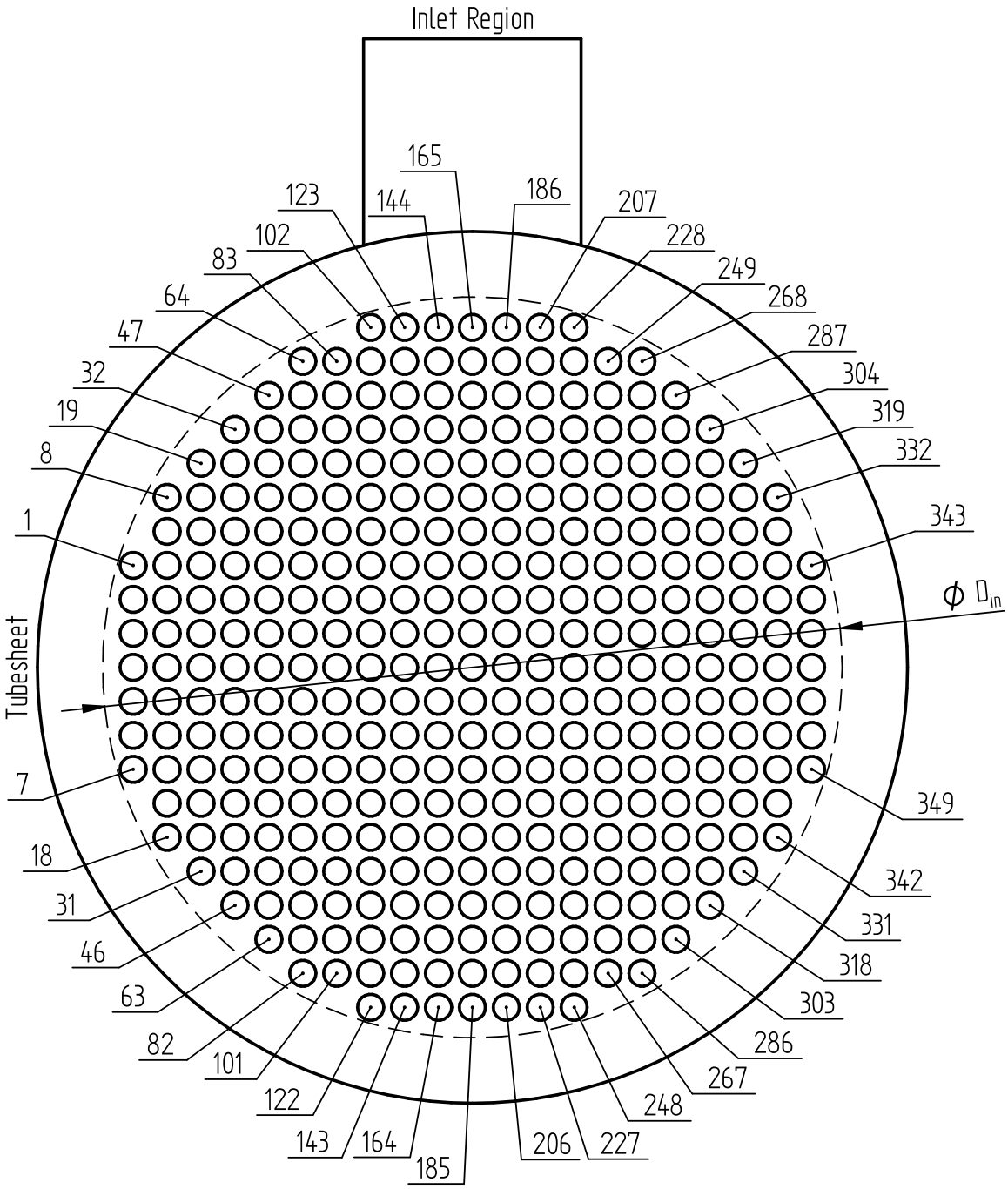

Figure 3.6 Tubesheet of the S&T case

3.2 Description of data from ANSYS Fluent

The data from (Fialová, 2017) were selected due to their quality and quantity. In the presented work, it is greatly described how simulations were prepared and carried out, and how the convergence was evaluated.

If the reader is interested in the detailed setup of those simulations, it is recommended to see the original work. But in a nutshell, following basic setup of the CFD solver in ANSYS Fluent was used in all cases:

- Pressure-based solver, absolute velocity formulation
- Double precision
- SIMPLE pressure-velocity coupling
- Green-Gauss node-based gradient
- k-ε realizable turbulence model, non-equilibrium wall function
- Mass flow inlet, pressure outlet

The calculation procedure was divided into three steps. The goal was to obtain an initialization field for the transient simulation that is to some extent similar to the final solution and will result in the divergence-free transient calculation.

In the beginning, a steady-state simulation with first order spatial discretization was used together with lower values of under-relaxation factors. After obtaining a converged solution of this initial problem, the spatial discretization was switched to second order. Values of under-relaxation factors were increased, as well, and another steady-state simulation was carried out. Next, the converged solution of this second simulation was used as the initial field of subsequent transient simulation. The convergence of solution was decided based on levels of residuals, and stabilization of flow properties in tubes, i.e., mass flow rates, pressure, and velocity.

The detailed CFD simulation of the additional problem, the tube side of a shell & tube heat exchanger, was set up in the same way.

3.3 Description of simulations performed using the simplified CFD code

Before presenting the actual outputs, it might be worth to briefly summarize the setup of simulations performed using the simplified CFD code and how the convergence was evaluated.

The size of timesteps was not set according to the Courant number (ANSYS Inc., 2018), but rather according to the rule of thumb, i.e., each time step should converge within 25 iterations. This rule was preferred due to the mesh being

substantially simplified. The convergence of problems was evaluated based on the levels of residuals and stabilizing mass flow rates in tubes. Flow rates were monitored in the middle of tubes where it was expected that the flow is uniform in its direction and is not disrupted by entering and exiting tubes. A converged solution for a time step was considered when levels of residuals were below 10e-3. While the difference of mass flow rates in tubes between two subsequent time steps below 10e-5 was required for getting a final solution in transient simulations. The pressure drop was obtained as the pressure difference between the inlet and the outlet area.

The computational time was in the range of minutes for the smallest case up to a few hours for the largest case using a typical modern desktop PC with Intel Core i5 8600K processor. Nevertheless, computational times of simulations done in the ANSYS Fluent using a server PC with Intel Xeon E5 2690 were in the order of hours or days.

3.4 Results

Results of simulations performed in the simplified CFD code are listed in Table 3.3 together with their pairs obtained in detailed CFD analyses. In almost all cases, the pressure loss predicted by the code is significantly higher. Similarly, the RSD values of all cases do not seem to follow the pattern found in (Fialová, 2017) at all. That is, the U configuration does not seem to offer the most even fluid distribution, but it is rather the Z or T configuration.

Figures 3.7 to 3.18 presented on the following pages detail the mass flow rates in tubes calculated using both approaches. These figures give a deeper look into the final fluid distribution than just RSD values and, therefore, provide another way of reviewing result. Since there were 14 simulations performed, and the results of similar cases are to a large extent alike, only two of them will be presented.

Table 3.3 Simulation results

		Simplified CFD code		ANSYS Fluent	
Case ID	mass flow, kg/s	RSD, %	Δp, Pa	RSD, %	Δp, Pa
01U	3.20	8.36	53,973.6	6.18	32,420.6
01Z	3.20	2.22	53,117.1	13.78	33,875.1
01T	3.20	1.91	55,059.1	12.31	36,020.0
02U	6.40	8.32	51,337.6	6.11	31,315.7
02Z	6.40	2.23	50,987.1	13.57	32,234.6
02T	6.40	1.91	51,968.7	11.28	35,590.6
04U	19.20	1.81	159,742.6	3.34	112,874.2
04Z	19.20	1.79	155,812.1	8.51	112,969.8
04T	19.20	1.32	167,628.2	6.97	126,703.2
06U	16.00	2.72	97,027.1	1.30	94,752.3
06Z	16.00	0.73	103,841.4	2.73	65,727.4
06T	16.00	0.29	95,805.6	2.76	94,108.4
07U	25.60	5.58	133,140.4	2.61	102,426.6
07Z	25.60	1.48	133,030.1	6.43	103,500.3

3.4.1 Case ID02

The first case to be presented is ID02 in all three flow arrangements, Figure 3.7 to 3.12. There are 20 tubes in 90° staggered position in this geometry.

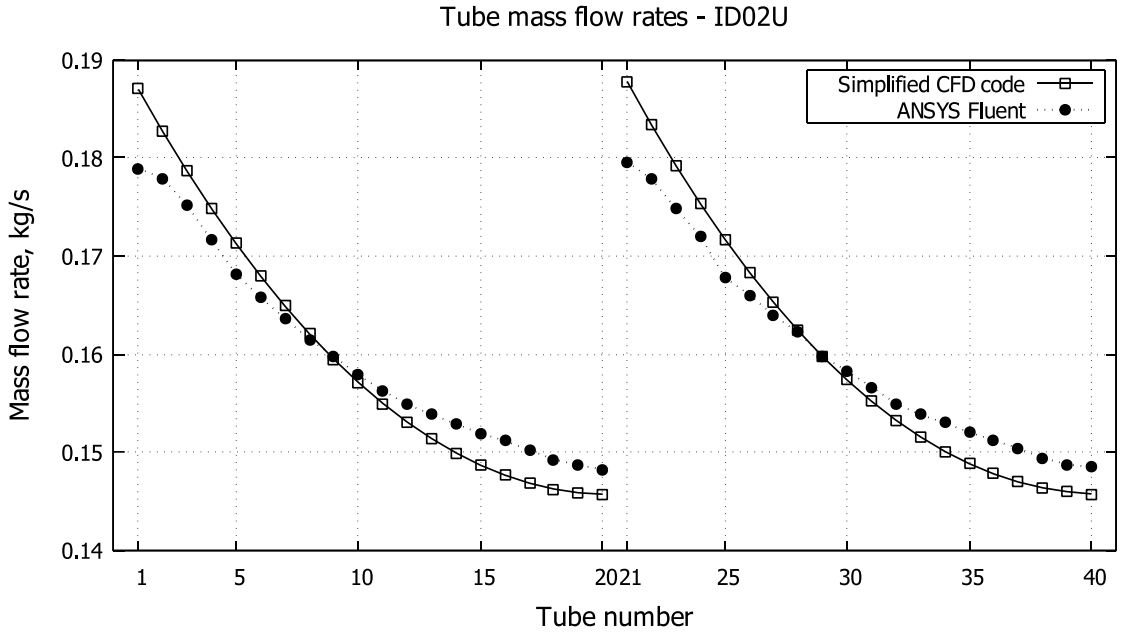

Figure 3.7 Tube mass flow rates – ID02U

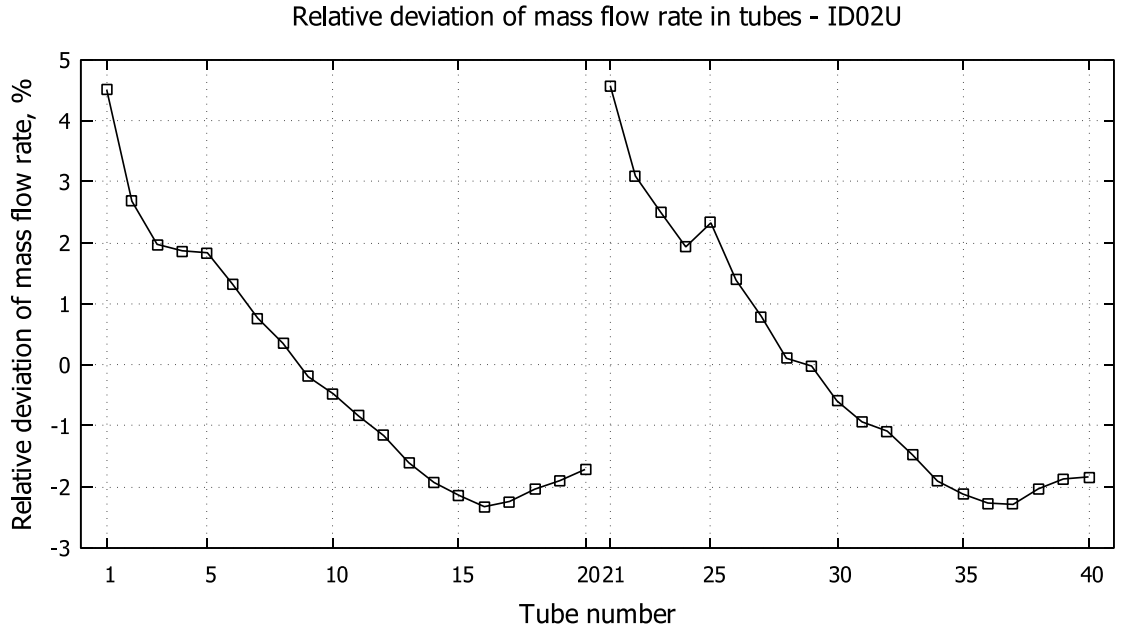

Figure 3.8 Relative error of mass flow rates – ID02U

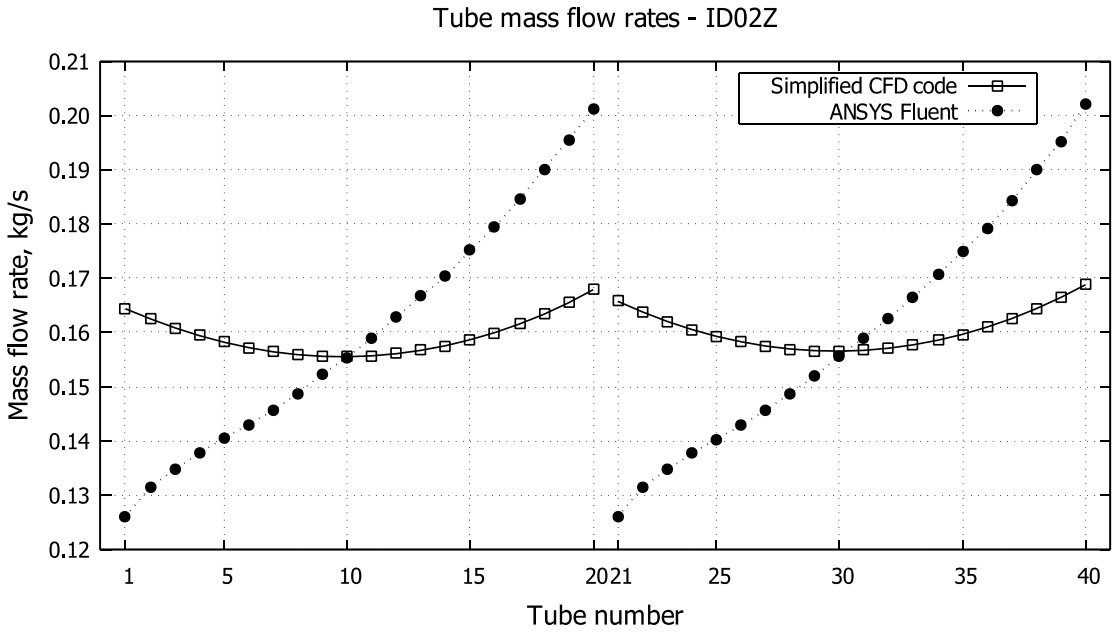

Figure 3.9 Tube mass flow rates – ID02Z

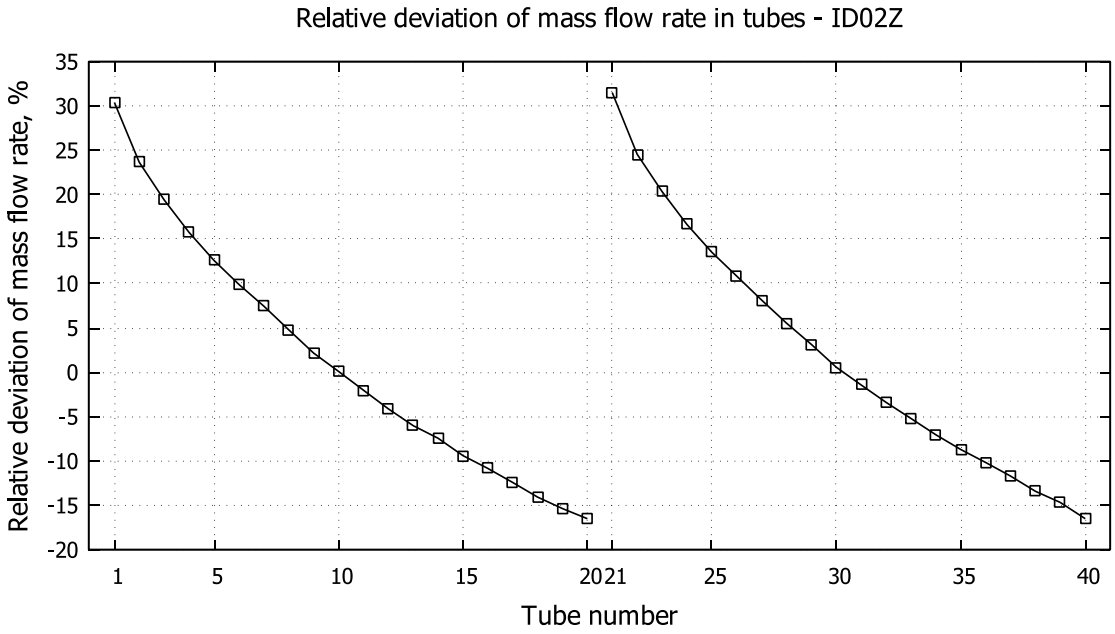

Figure 3.10 Relative error of mass flow rates – ID02Z

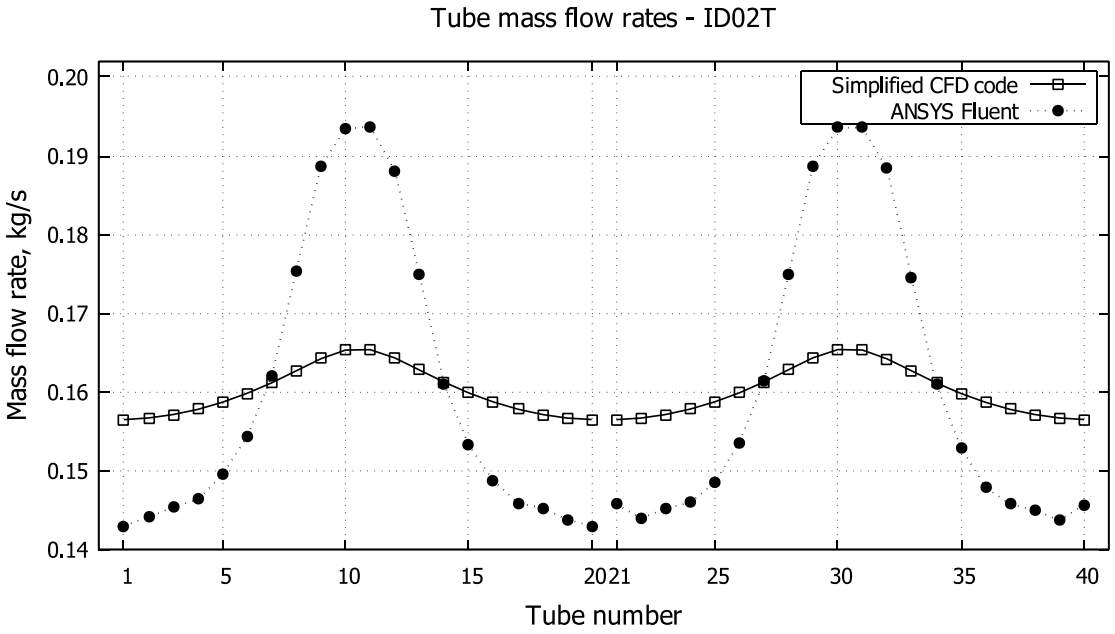

Figure 3.11 Tube mass flow rates – ID02T

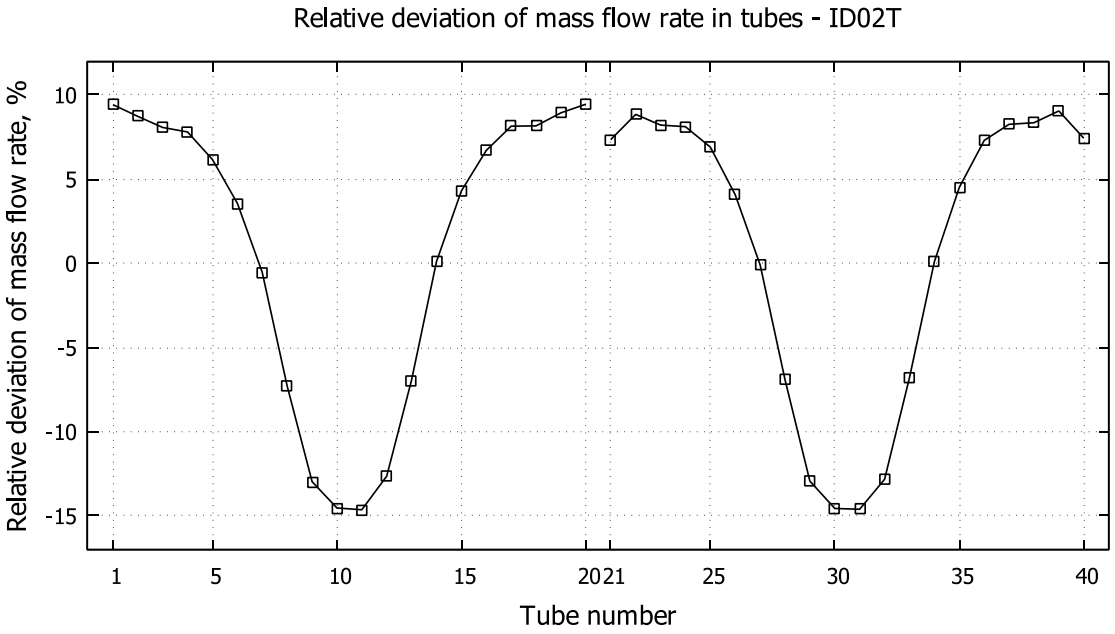

Figure 3.12 Relative error of mass flow rates – ID02T

Mass flow rates in tubes for ID02U configuration (Figure 3.7) follow the results of detailed CFD simulation rather well. This is also proved by a relative deviation of less than 5 %. In fact, it is lower than 3 % for all but three out of 20 tubes. Simulations of Z and T configurations in the simplified CFD code, on the other hand, rendered a lot more even fluid distribution compared to the detailed CFD simulations of these arrangements.

In U and Z flow arrangements, the predicted flow distribution from the simplified CFD code matches the detailed analysis with the best accuracy in the middle of the tube rows. While the difference in the mass flow rate in tubes increases more towards the edges of the main channels. This is the most obvious in the Z configuration where the relative difference in mass flow rates reaches around 30 % (Figure 3.10).

The plot with the mass flow rates in the Z flow arrangement (Figure 3.9) shows a significant difference in the pattern of this flow property. The ANSYS Fluent predicts that mass flow rates will grow for every tube the further from the inlet area it is. The simplified code, on the other hand, predicts a decrease of mass flow rates for the tubes the closer their position is to the middle of the tube bundle.

The T flow arrangement (Figure 3.11), in contrast to that, does follow the trend of flow rates that is provided by the detailed data. Mass flow rates in the middle area of this tube bundle, where inlets and outlets are located, are very similar to those in outer tubes. However, the difference between the highest and the lowest value is around 0.01 kg/s meaning that the fluid flow in the tube bundle is rather uniform.

Since the simplified simulation of the U flow arrangement follows the detailed data quite accurately and results of the other two arrangements are showing a larger deviation, it can be assumed that the flow in the Z and T arrangement experienced higher gradients and flow instabilities that the mesh and/or the turbulence model could not describe sufficiently. As explained before, increasing the mesh size may go against the main goal of this CFD code and significantly increase the computational time. Therefore, using a different turbulent model seems like a better option.

3.4.2 Case ID06

Next case to be presented is ID06, Figure 3.13 to 3.18. This case has 50 tubes in 45° staggered layout. There were all three flow arrangements considered, too.

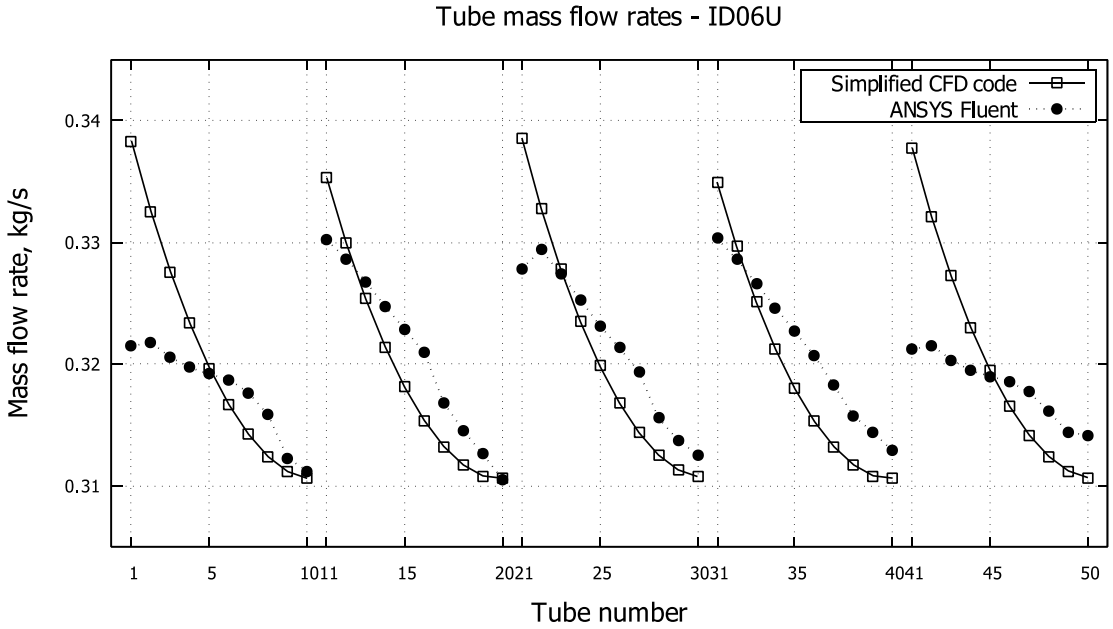

Figure 3.13 Tube mass flow rates – ID06U

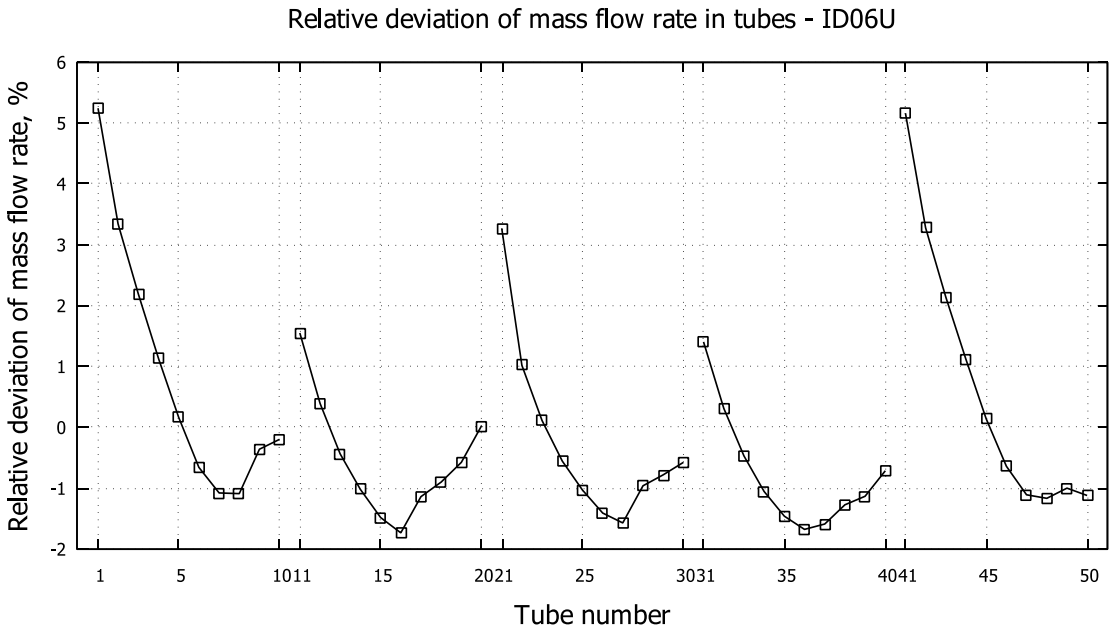

Figure 3.14 Relative error of mass flow rates – ID06U

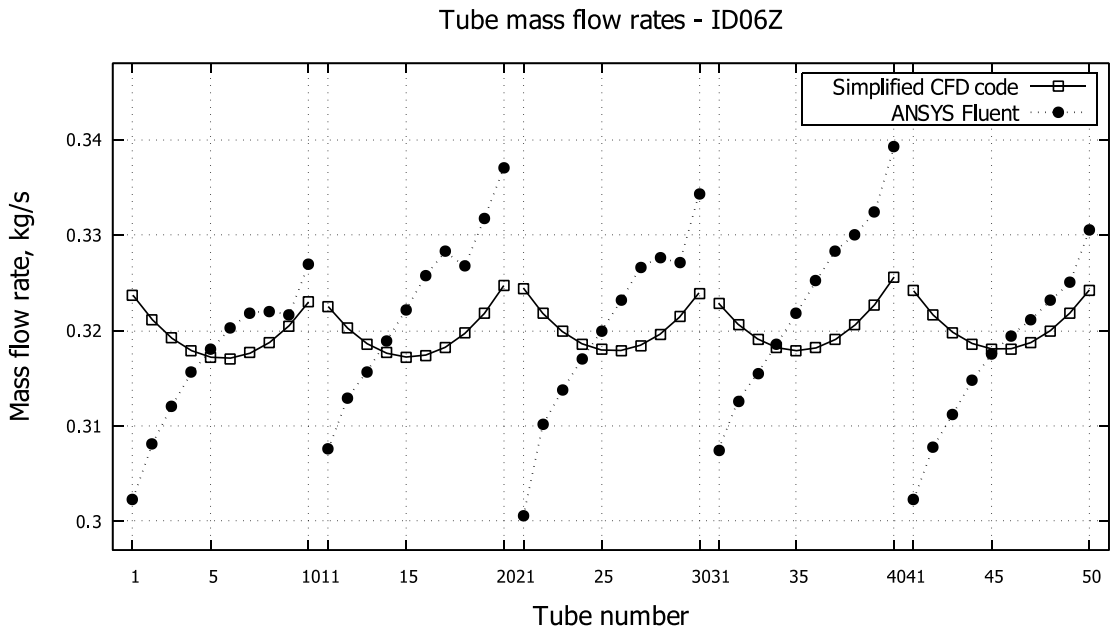

Figure 3.15 Tube mass flow rates – ID06Z

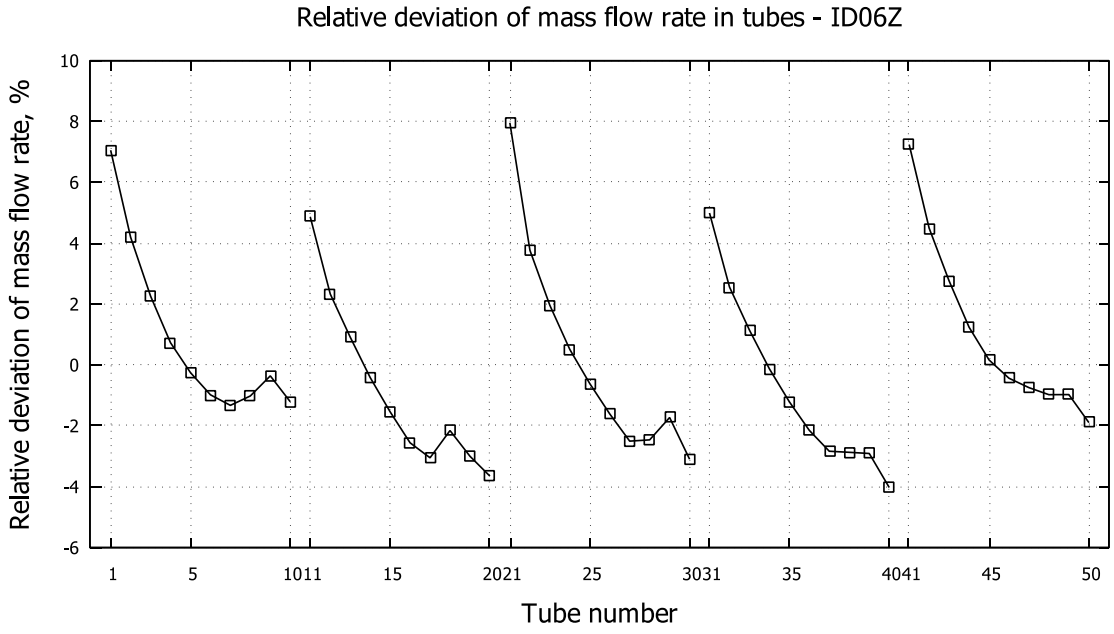

Figure 3.16 Relative error of mass flow rates – ID06Z

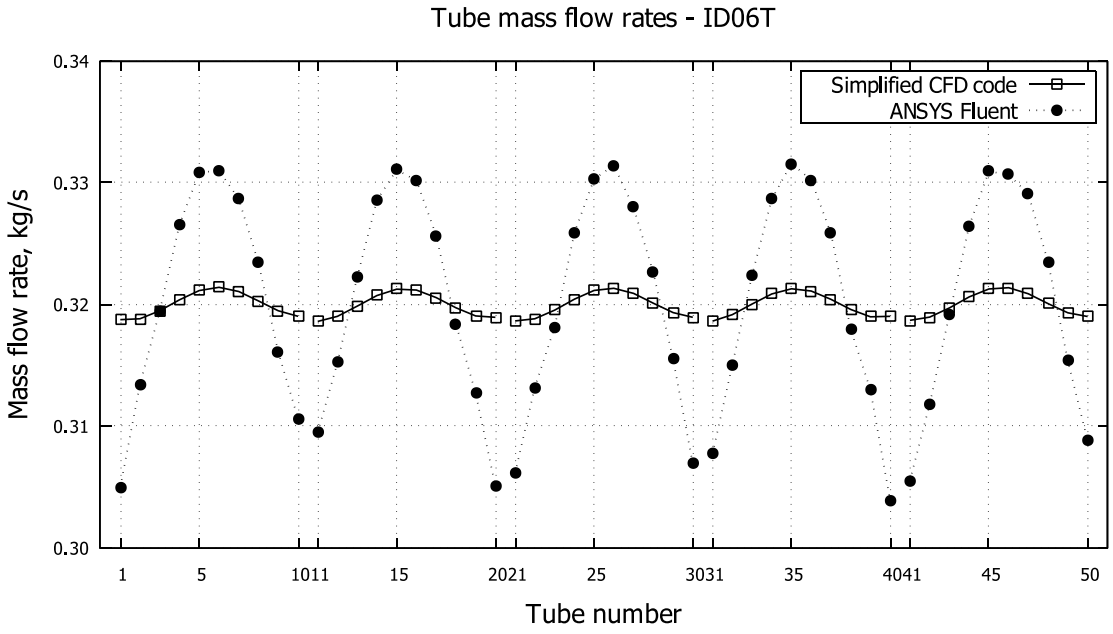

Figure 3.17 Tube mass flow rates – ID06T

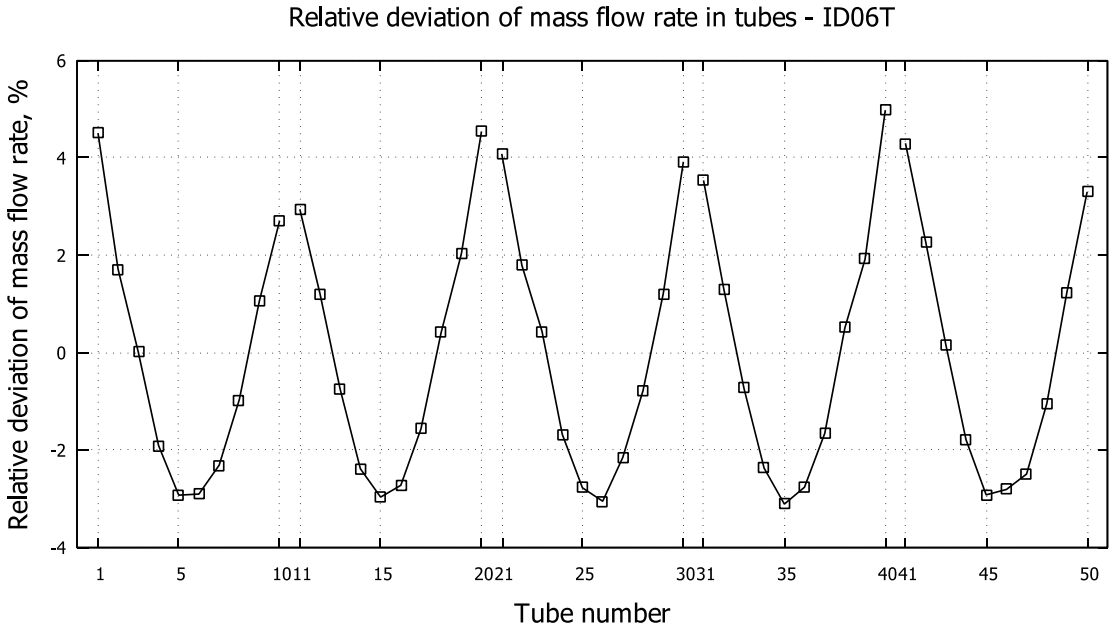

Figure 3.18 Relative error of mass flow rates – ID06T

Again, the U flow arrangement (Figure 3.13) follows outcomes of the detailed CFD analysis rather well. Rendering the relative error to be under 6 % for all tubes (Figure 3.14). As for other arrangements of ID06 case, Z and T, the fluid distribution from the proposed CFD code is again more even than from the detailed CFD simulations.

In the z flow arrangement (Figure 3.15), the mass flow rates once more show a different trend than the detailed data. While the pattern of the mass flow rates in the T flow arrangement (Figure 3.17) is similar to the mass flow rates from the detailed CFD simulation. Albeit, it appears that the variations of this flow property are again somewhat limited in its range.

In all considered cases, the CFD code yielded mass flow rates in tubes in the Z and T arrangement in a very limited range compared to the range of mass flow rates obtained from detailed CFD analyses. Based on this, it can be assumed that the simplified CFD code has its limitations in handling the repeating changes of the tubesheet geometry, i.e., the openings of the tubes, when there are larger pressure gradients present. As explained before at the ID02 case, this limitation probably comes from the way the meshing process is done and/or from the selected turbulence model.

3.4.3 S&T bundle

As mentioned before, the presented CFD code is able to work with various types of geometries, provided it is acceptable to use spatial discretisation using rectangular cuboids. To demonstrate this capability, a tube bundle consisting of 349 tubes with cylindrical distributor and collector was simulated using the simplified CFD code and ANSYS Fluent. This tube bundle represents the tube side of a shell & tube heat exchanger. The key results of this analysis are in Table 3.4, plots of mass flow rates are in Figure 3.20 and 3.21.

The predicted mass flow rates from the simplified CFD analysis follow the same trend as seen before. The CFD code rendered higher pressure loss then the ANSYS Fluent with a relative deviation of 6.7 %. The RSD value being lower in case of the simplified CFD code means that this code, again, predicts more uniform flow distribution. This fact can be also seen on figures 3.20 and 3.21 where values of mass flow rate in tubes are plotted for both simulation tools. The location of the plane from which the data for flow properties are taken is in the middle of tubes. Its orientation within the coordinate system is visualised in Figure 3.19.

Table 3.4 Simulation results, the S&T case

		Simplified CFD code		ANSYS Fluent	
Case	mass flow, kg/s	RSD, %	Δp, Pa	RSD, %	Δp, Pa
S&T	20	2.30	8,969.8	7.34	8,409.0

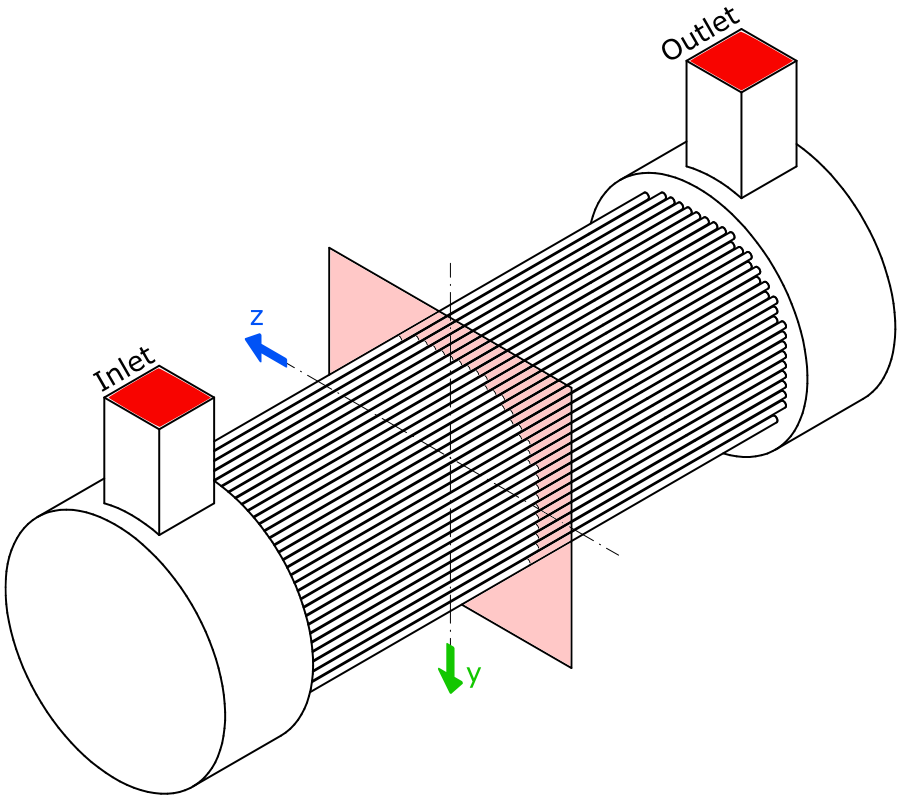

Figure 3.19 Orientation of the interrogation plane within the coordinate system

The simplified CFD code predicts the highest mass flow rates in the area close to the inlet region (Figure 3.20). From there, towards the opposite side of the tubesheet, flow rates are decreasing. However, the difference in flow rates for the tubes that are right by the inlet and the tubes that are the farthest is around 0.002 kg/s.

In contrast, when comparing these two figures, it can be observed that ANSYS Fluent indicates in Figure 3.21 a wider range of mass flow rates. The difference between the lowest and the highest value is around 0.005 kg/s. The distribution of the fluid also follows a completely different pattern. There, mass flow rates of tubes in the middle of the tube bundle are the lowest while they grow rather quickly for tubes away from that part of the tube bundle. This is also where the relative deviation (Figure 3.22) reaches its magnitudes.

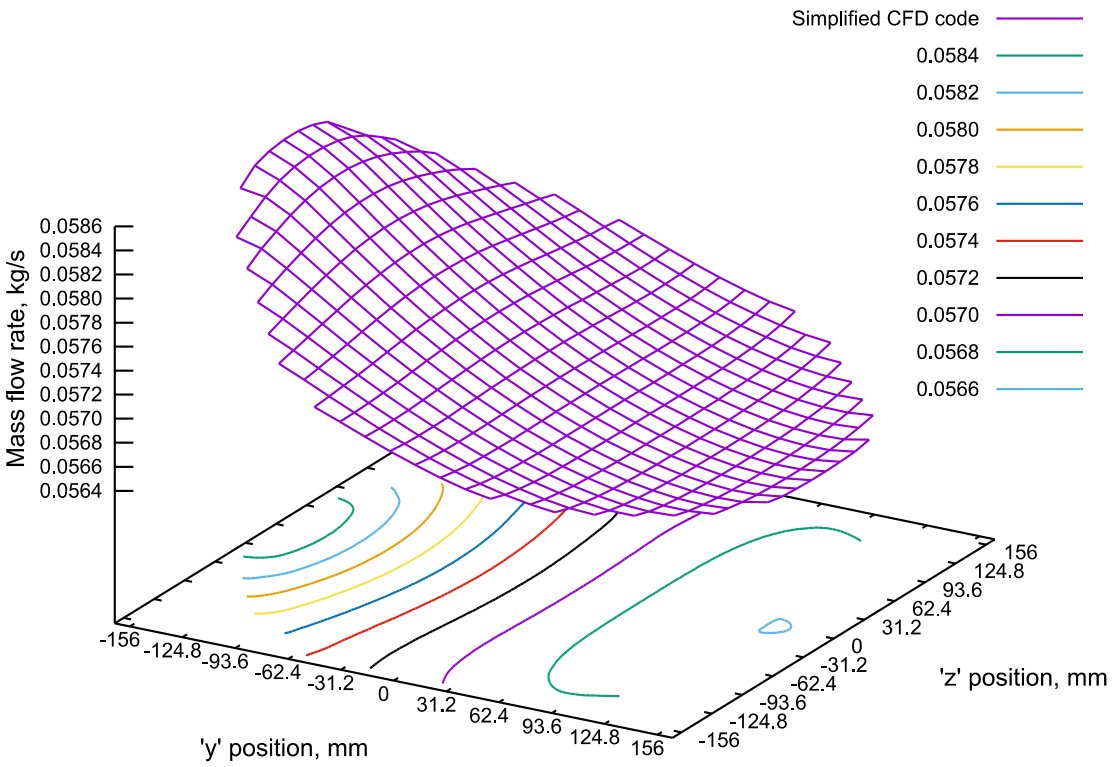

Figure 3.20 Mass flow rates in tubes in the S&T case – simplified CFD code

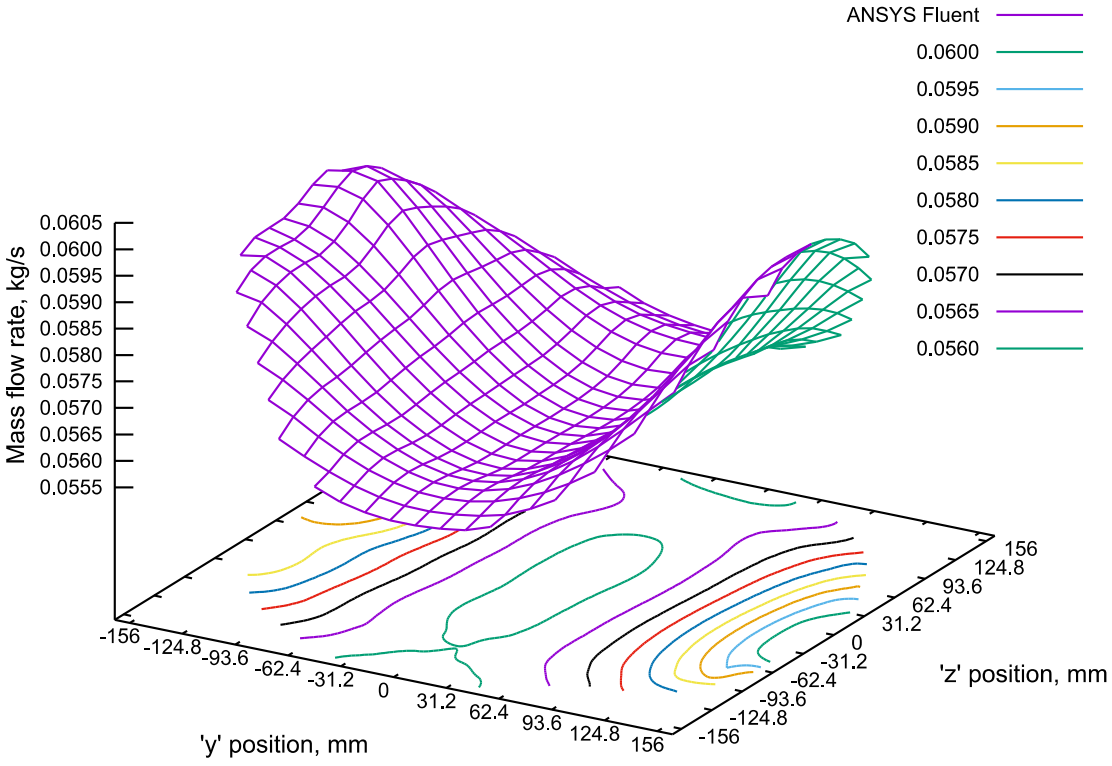

Figure 3.21 Mass flow rates in tubes in the S&T case – ANSYS Fluent

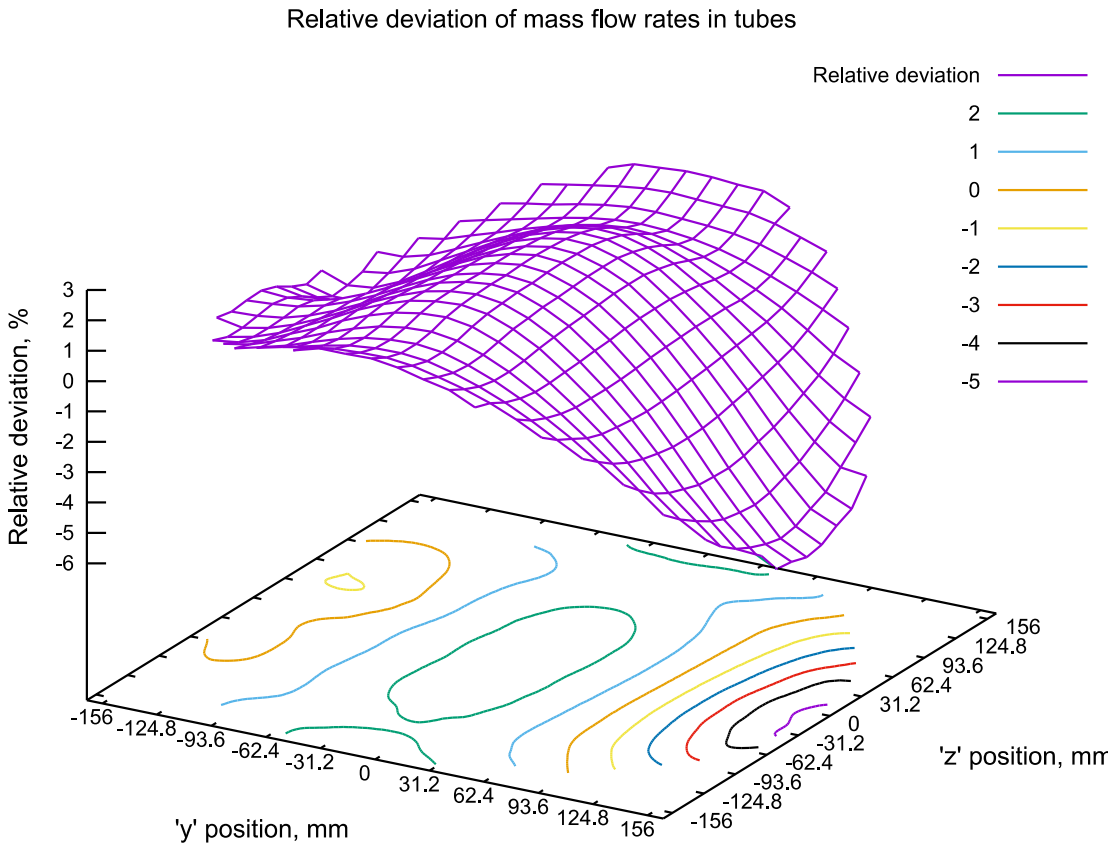

Figure 3.22 Relative error of mass flow rates – the S&T case

3.5 Future work

As presented, the simplified CFD code at this stage is able to carry out flow simulations of a variety of geometries. Even the ones consisting of a larger number of tubes. The extent of employed simplifications gives short computational times compared to the detailed CFD analyses, too. With regard to that, the accuracy of the results may be considered as satisfactory for the initial stage of a design process.

The code in this state seems good enough for the comparison of individual geometries against each other. Once an optimal candidate for the required application is found, the detailed CFD analysis should follow. The improvement of the accuracy could be gained with a different turbulence model and an increase in the mesh size. Reduction of computational times, on the other hand, could be accomplished by optimization of the code itself. Other features that can make the code more usable are a graphical user interface and a material database. Anyhow, implementing most of these modifications will need a solid knowledge of the Java programming language.

4 Data visualisation

Another feature that is built in the simplified CFD code is the export of the data for post-processing. Specifically, there is a method that writes cell values, i.e., x, y, and z position, x, y, and z velocity, pressure, and temperature into the CSV file that can be read in Autodesk 3ds Max (Autodesk Inc., 2019). This software is primarily intended for artists and designers but it has also tools for visualisation of CFD data. Therefore, with the export of data set up in the proper way in the simplified CFD code, it is possible to use it for rendering contours, drawing vectors, or generating pathlines.

Figures 4.1 up to 4.4 on subsequent pages demonstrate this capability using the data from case ID02U. The first figure is the velocity contour, the second contour shows gauge pressure. Figure 4.3 shows pathlines in the distributor, while Figure 4.4 shows pathlines in the collector.

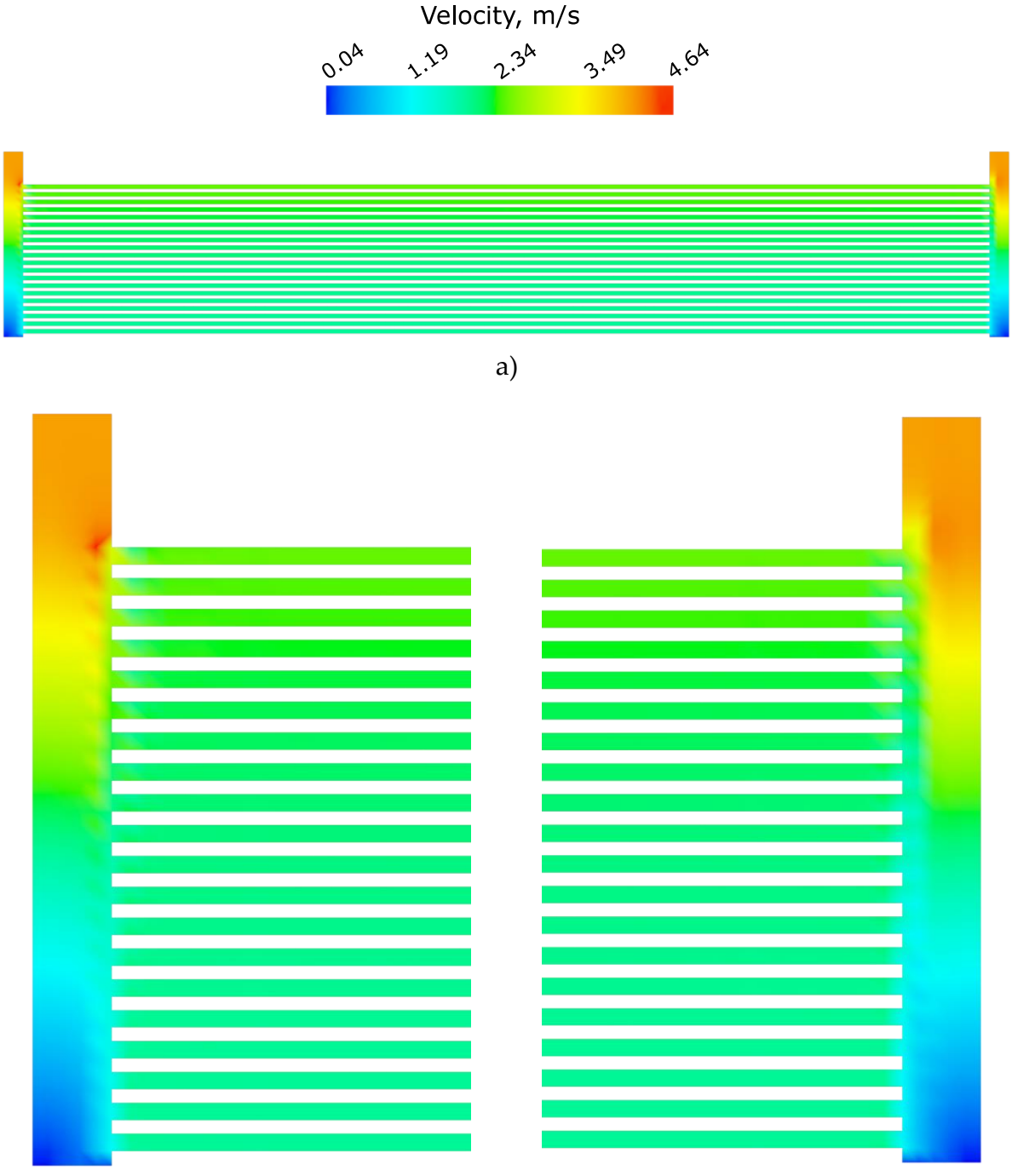

Figure 4.1 Velocity contour plot (ID02U)
a) complete geometry, b) distributor, c) collector

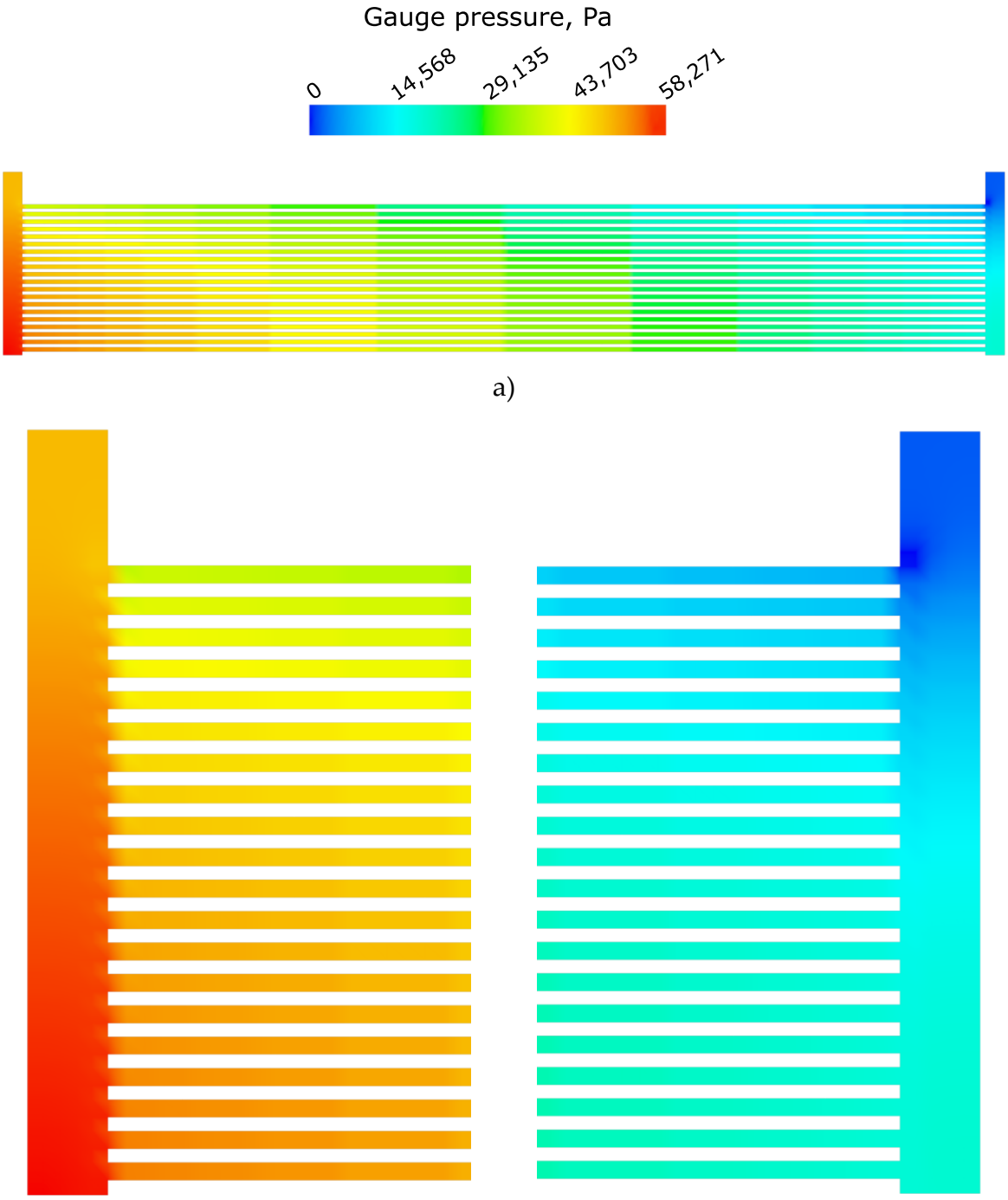

Figure 4.2 Pressure contour plot (ID02U)
a) complete geometry, b) distributor, c) collector

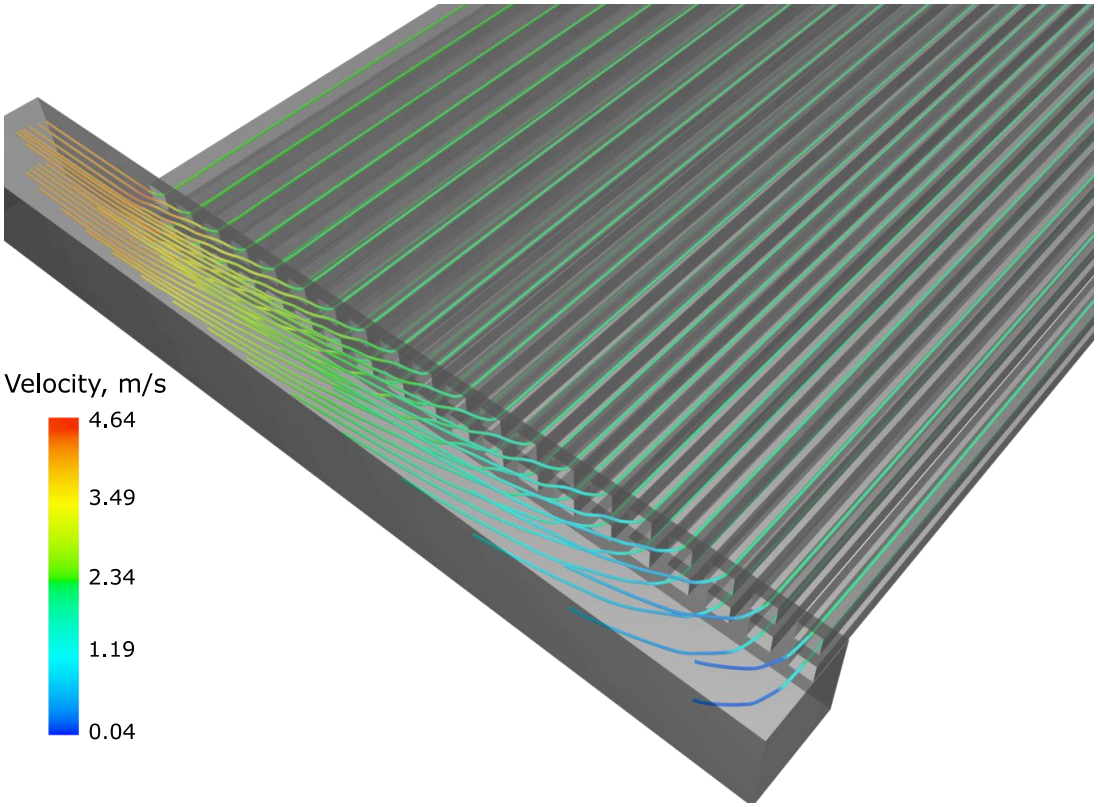

Figure 4.3 Pathlines in the distributor (ID02U)

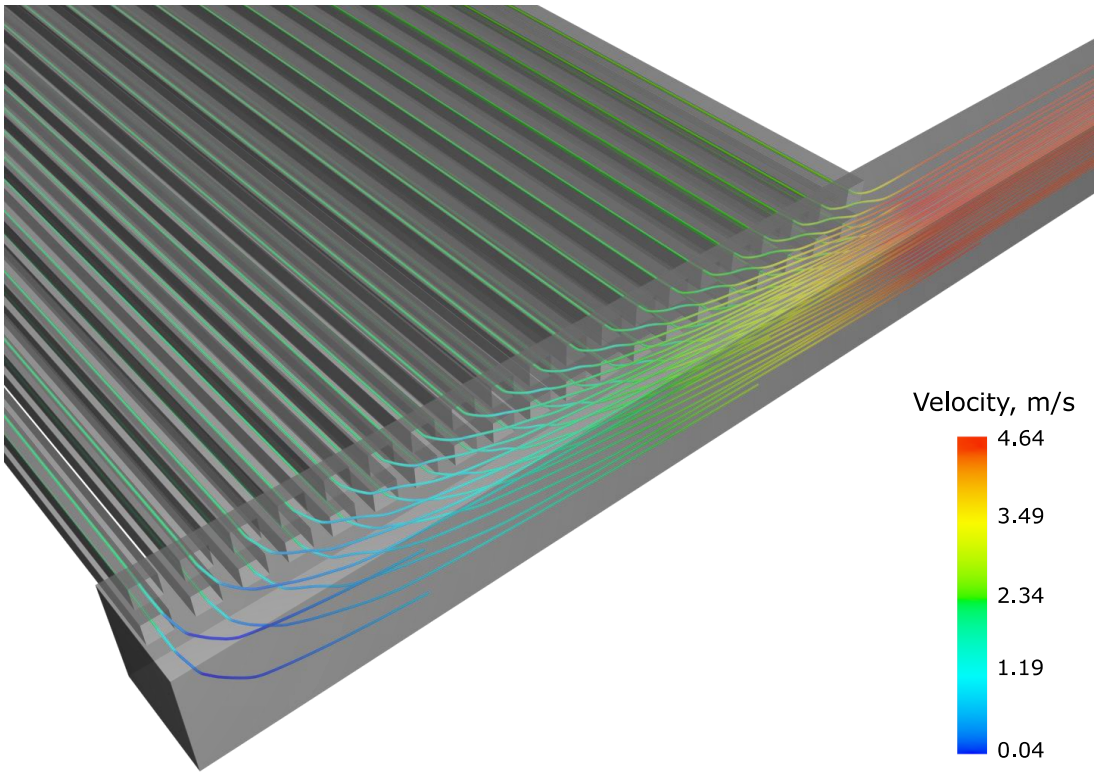

Figure 4.4 Pathlines in the collector (ID02U)

Conclusions

The goal of this thesis was to develop and test a computer code that would evaluate flow distribution in systems with parallel tubes. In the first part, a brief overview of the historical development of methods for flow analysis is presented. Different criteria for evaluation of the flow distribution uniformity are presented there, too.

The following part introduces the custom CFD code. The CFD code was written in the Java programming language and uses third-party low-level libraries for performance reasons. In addition to that, the computational model uses certain simplifications, as well. These features were expected to result in short evaluation times with relatively good data accuracy. This was assessed using data from detailed CFD analyses carried out in a commercial CFD tool. The corresponding data were taken from (Fialová, 2017).

Flow systems with various flow arrangements, tube arrangements, and numbers of tubes were analysed (15 geometries in total). In these cases, water was used as the working fluid. The computational model was solving discretized equations governing mass conservation, momentum conservation, the turbulent kinetic energy, and the turbulent energy dissipation rate in the transient formulation. Solving the energy equation was not enabled despite the fact that it is implemented in the code. In order to monitor the convergence, levels of residuals and mass flow rates in the tubes were tracked.

The predicted mass flow rates in tubes were used to evaluate the flow distribution uniformity. The relative standard deviation from uniform flow (RSD) is the overall criterion. Obtained RSD values together with pressure drops were then compared with the results presented in (Fialová, 2017). It was found that the least uniform flow distribution predicted by the simplified CFD code can be expected with the U flow arrangement. This does not follow the pattern seen in the detailed data where the U flow arrangement gives the best uniformity. Also, the code generally predicts higher pressure drops.

Subsequently, the predicted mass flow rates for individual tubes from both CFD tools were compared and their relative errors were evaluated. It was concluded that the mass flow rates in U flow arrangements follow the detailed data best. Mass flow rates in the Z flow arrangements from the simplified CFD code do not follow the trend given by the detailed CFD analyses. Similarly to the U flow arrangement, the trend for the Z flow arrangements seem to follow the detailed data. However, the predicted mass flow rates are somehow limited in their range, i.e., they are almost the same in all tubes. This and the results for the Z flow arrangements are probably due to the presence of more vigorous changes in the fluid flow that the mesh and the selected turbulence model may not be able to properly describe. For such flow arrangements, it might be worth to try a different turbulence model. However, its implementation would likely be laborious, and it would result in longer computational times, as well. Therefore, the implementation of such models must be carefully considered as they could also introduce significant instabilities due to the nature of the mesh.

In any case, the code provides data in rather short evaluation times, compared to the detailed CFD analyses. What is more, further improvement in this regard is possible with proper code optimization. The code is universal in terms of possible geometries to be analysed but creating of the respective meshes is quite laborious at this stage. Even though the code does not have any built-in features for post-processing, it can generate data for other software tools. Once the meshing process is made more user-friendly and possible issues regarding the turbulence model are fixed, the code might have the potential to be used in its targeted applications.

Bibliography

Acrivos, A., Babcock, B.D. & Pigford, R.L., 1959. Flow distributions in manifolds. *Chemical Engineering Science*, 10(1), pp.112-124. Available at: http://www.sciencedirect.com/science/article/pii/0009250959800300.

Anbumeenakshi, C. & Thansekhar, M.R., 2016. Experimental investigation of header shape and inlet configuration on flow maldistribution in microchannel. *Experimental Thermal and Fluid Science*, 75(2016), pp.156-161. Available at: http://www.sciencedirect.com/science/article/pii/S0894177716300188.

ANSYS Inc., 2018. *ANSYS Fluent User's Guide, Release 19.1*, Canonsburg, PA, USA: ANSYS Inc.

Autodesk Inc., 2019. Autodesk CFD. Available at: https://www.autodesk.com/products/cfd/overview [Accessed May 13, 2019].

Autodesk Inc., 2019. 3DS MAX. Available at: https://www.autodesk.com/products/3ds-max/overview [Accessed May 13, 2019].

Bailey, B.J., 1975. Fluid Flow in Perforated Pipes. *Journal of Mechanical Engineering Science*, 17(6), pp.338-347. Available at: https://doi.org/10.1243/JMES_JOUR_1975_017_048_02.

Bajura, R. & H. Jones Jr, E., 1976. Flow Distribution Manifold. *Journal of Fluids Engineering*, 98(4), pp.654-665.

Bava, F. & Furbo, S., 2016. A numerical model for pressure drop and flow distribution in a solar collector with U-connected absorber pipes. *Solar Energy*, 134(2016), pp.264-272. Available at: http://www.sciencedirect.com/science/article/pii/S0038092X16301074.

Dassault Systèmes, 2019. SOLIDWORKS Flow Simulation. Available at: https://www.solidworks.com/product/solidworks-flow-simulation [Accessed May 13, 2019].

Fialová, D., 2017. *Distribuce toku v zařízeních s hustými svazky trubek*. Master thesis. Brno. Available at: http://hdl.handle.net/11012/66702 [Accessed May 13, 2019].

Gandhi, M.S. et al., 2012. CFD simulation for steam distribution in header and tube assemblies. *Chemical Engineering Research and Design*, 90(4), pp.487-506. Available at: http://www.sciencedirect.com/science/article/pii/S0263876211003005.

Chýlek, M., 2018. *Tok látek v nestandardních procesních a energetických zařízeních*. Master thesis. Brno. Available at: http://hdl.handle.net/11012/81651 [Accessed May 13, 2019].

Jiao, A., Zhang, R. & Jeong, S., 2003. Experimental investigation of header configuration on flow maldistribution in plate-fin heat exchanger. *Applied Thermal Engineering*, 23(10), pp.1235-1246.

Kelvion, 2019. Box cooler. *Marine Cooling System*. Available at: https://www.kelvion.com/products/product/box-cooler/ [Accessed May 13, 2019].

Kelvion, 2019. Searle MV. *Industrial Condenser*. Available at: https://www.kelvion.com/products/product/searle-mv/ [Accessed May 13, 2019].

Kumaran, R.M., Kumaraguruparan, G. & Sornakumar, T., 2013. Experimental and numerical studies of header design and inlet/outlet configurations on flow mal-distribution in parallel micro-channels. *Applied Thermal Engineering*, 58(1-2), pp.205-216.

Labbadlia, O. et al., 2017. Numerical study of the influence of tube arrangement on the flow distribution in the header of shell and tube heat exchangers. *Applied Thermal Engineering*, 126(1), pp.315-321. Available at: http://www.sciencedirect.com/science/article/pii/S1359431117313807.

Lalot, S. et al., 1999. Flow maldistribution in heat exchangers. *Applied Thermal Engineering*, 19(8), pp.847-863. Available at: http://www.sciencedirect.com/science/article/pii/S1359431198000908.

Launder, B. & Spalding, D., 1972. Mathematical Models of Turbulence. *Academic Press*.

Leonard, B.P., 1979. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19(1), pp.59-98. Available at: http://www.sciencedirect.com/science/article/pii/0045782579900343.

Moukalled, F.H., Mangani, L. & Darwish, M., 2016. *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*, Cham: Springer.

NUMECA International, 2019. FINE™/Turbo. Available at: https://www.numeca.com/product/fineturbo [Accessed May 13, 2019].

Oracle, 2019. What is Java?. Available at: https://www.java.com/en/download/whatis_java.jsp [Accessed April 13, 2019].

Patankar, S.V., 1980. *Numerical heat transfer and fluid flow*, Bristol, PA, USA: Taylor & Francis.

Perry, R.H., 1998. *Perry's chemical Engineer's Handbook* 7 ed., New York: McGraw-Hill.

Pigford, R.L., Asharf, M. & Miron, Y.D., 1983. Flow distribution in piping manifolds. *Ind. Eng. Chem. Fundam.*, 22(4), pp.463-471.

Schildt, H., 2014. *Java A Beginner's Guide: Create, Compile, and Run Java Programs Today* 6 ed., USA: McGraw-Hill Education.

Siemens PLM Software, 2019. STAR-CCM+. Available at: https://mdx.plm.automation.siemens.com/star-ccm-plus [Accessed May 13, 2019].

The Apache Software Foundation, 2016. Commons Math: The Apache Commons Mathematics Library. Available at: http://commons.apache.org/proper/commons-math/ [Accessed May 13, 2019].

The OpenFOAM Foundation Ltd, 2019. OpenFOAM. Available at: https://openfoam.org/ [Accessed May 13, 2019].

ThermExcel, 2014. Physical characteristics of water. Available at: https://www.thermexcel.com/english/tables/eau_atm.htm [Accessed May 13, 2019].

Turek, V., 2018. On improving computational efficiency of simplified fluid flow models. *Chemical Engineering Transactions*, 70(2018), pp.1447-1452.

Turek, V. et al., 2015. Efficient 2D Model of Flow Distribution in Dense Tube Bundles. *Chemical Engineering Transactions*, 45(2015), pp.1177-1182. Available at: https://www.cetjournal.it/index.php/cet/article/view/CET1545197.

Turek, V., Fialová, D. & Jegla, Z., 2016. Efficient Flow Modelling in Equipment Containing Porous Elements. *Chemical Engineering Transactions*, 52(2016), pp.487-492. Available at: https://www.cetjournal.it/index.php/cet/article/view/CET1652082.

Turek, V. et al., 2011. Optimum design of fluid distribution systems in heat exchangers. *Asia-Pacific Journal of Chemical Engineering*, 6(5), pp.750-759. Available at: https://doi.org/10.1002/apj.516.

van Doormaal, J.P. & Raithby, G.D., 1984. Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows. *Numerical Heat Transfer Applications*, 7(1984), pp.147-163.

Versteeg, H.K. & Malalasekera, W., 1995. *An introduction to computational fluid dynamics: The finite volume method*, Harlow: Longman.

Wei, M. et al., 2016. Numerical and experimental investigation on the realization of target flow distribution among parallel mini-channels. *Chemical Engineering Research and Design*, 113(2016), pp.74-84. Available at: http://www.sciencedirect.com/science/article/pii/S0263876216301757.

Wendykier, P. & Nagy, J.G., 2010. Parallel Colt: A High-Performance Java Library for Scientific Computing and Image Processing. *ACM Transactions on Mathematical Software*, 37(3), pp.31:1-31:22.

Wen, J. & Li, Y., 2004. Study of flow distribution and its improvement on the header of plate-fin heat exchanger. *Cryogenics*, 44(11), pp.823-831. Available at: http://www.sciencedirect.com/science/article/pii/S0011227504001390.

Yaïci, W., Ghorab, M. & Entchev, E., 2014. 3D CFD analysis of the effect of inlet air flow maldistribution on the fluid flow and heat transfer performances of plate-fin-and-tube laminar heat exchangers. *International Journal of Heat and Mass Transfer*, 74(1), pp.490-500. Available at: http://www.sciencedirect.com/science/article/pii/S0017931014002403.

Zhang, Z. & Li, Y.Z., 2003. CFD simulation on inlet configuration of plate-fin heat exchangers. *Cryogenics*, 43(12), pp.673-678. Available at: http://www.sciencedirect.com/science/article/pii/S0011227503001796.

Nomenclature

a	Edge size of the rectangular tube representation, m
c_P	Specific heat capacity at constant pressure, $\mathrm{J/(kg \cdot K)}$
d	Inner diameter of the tube, m
D	Diameter of a main channel, m
DC	Division Criterion, -
δ	Relative standard deviation (RSD), %
Δp	Pressure loss, Pa
Δt	Size of the timestep, s
DF	Deviation factor, -
D_{in}	Diameter that circumscribes the tube bundle on tubesheets, m
ε	Turbulent energy dissipation rate, $\mathrm{m^2 \cdot s^{-3}}$
E_t	Percentual nonuniformity, %
H	Height of main channel, m
k	Turbulent kinetic energy, $\mathrm{m^2 \cdot s^{-3}}$
λ	Thermal conductivity, $\mathrm{W/(m \cdot K)}$
L	Length of main channel, m
L_{in}	Length of inlet region, m
L_{out}	Length of outlet region, m
$\dot{m}_{avg}$	Average mass flow rate in tubes, kg/s
$\dot{m}_i$	Mass flow rate in the *i*-th tube, kg/s
$\dot{m}_{id}$	Mass flow rate in a tube at ideal distribution of the fluid, kg/s
$\mu_{\mathrm{eff},\varepsilon}$	Effective viscosity for the Turbulent energy dissipation rate, $\mathrm{Pa \cdot s}$
$\mu_{\mathrm{eff},k}$	Effective viscosity for the turbulent kinetic energy, $\mathrm{Pa \cdot s}$
N	Number of tubes, -
p	Pressure field, Pa
ρ	Density, $\mathrm{kg/m^3}$
R_L	Number of tubes in longitudinal direction, -
R_W	Number of tubes in lateral direction, -
S	Velocity criterion of flow nonuniformity, -
S'	Velocity criterion of flow nonuniformity, -
S_{En}	Energy source term in the energy equation, $\mathrm{W/m^3}$
S_ε	Energy dissipation rate source term in the turbulent energy dissipation rate equation, $\mathrm{kg/(m \cdot s^4)}$
S_k	Kinetic source term in the turbulent kinetic energy equation, $\mathrm{kg/(m \cdot s^3)}$
S_L	Spacing of tubes in longitudinal direction, m
S_{Mx}	Momentum source term in x-momentum equation, $\mathrm{kg/(m^2 \cdot s^2)}$
S_{My}	Momentum source term in y-momentum equation, $\mathrm{kg/(m^2 \cdot s^2)}$
S_{Mz}	Momentum source term in z-momentum equation, $\mathrm{kg/(m^2 \cdot s^2)}$

S_W	Spacing of tubes in lateral direction, m
T	Temperature, K
t	Time, s
u	Velocity in x direction, m/s
$\boldsymbol{u}$	Velocity vector, m/s
u_{avg}	Average velocity of fluid in a branch (tube), m/s
u_i	Velocity of fluid in the i-th branch (tube), m/s
v	Velocity in y direction, m/s
$\dot{V}_i'$	Dimensionless flow rate in the i-th tube, -
$\dot{V}_i$	Volume flow rate in the i-th tube, m^3/s
w	Velocity in z direction, m/s
W	Width of main channel, m
W_1	Dimension of cross-section of inlet or outlet region, m
W_2	Dimension of cross-section of inlet or outlet region, m

B	Bottom face of a control volume
BC	Boundary Condition
BiCGstab	BiConjugate Gradient Stabilized matrix solver
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CG	Conjugate Gradient matrix solver
CPU	Central Processing Unit
CSV	Comma Separated Values
DTBFMM	Dense Tube Bundle Flow Modeller Matrix
E	East face of a control volume
ID	Identification number
N	North face of a control volume
PC	Personal Computer
QUICK	Quadratic Upstream Interpolation for Convective Kinetics
RSD	Relative Standard Deviation
S	South face of a control volume
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
SIMPLEC	Semi-Implicit Method for Pressure Linked Equations Consistent
T	Top face of a control volume
W	West face of a control volume