Analysis of a gas flow in structured packed bed columns using computational fluid dynamics

Niina Jäntti

Porvoo, 18.10.2013

Aalto University
School of Engineering
Department of Applied Mechanics

Niina Jäntti

Analysis of a gas flow in structured packed bed columns using computational fluid dynamics

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo October 18, 2013
Thesis supervisor: Professor Timo Siikonen
Thesis advisor: Johanna Vaittinen, D.Sc. (Tech.)

AALTO UNIVERSITY
SCHOOL OF ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Niina Jäntti

Title: Analysis of a gas flow in structured packed bed columns using computational fluid dynamics

Date: 18.10.2013 Language: English Number of pages:9+139

Department: Department of Applied Mechanics

Professorship: Aeronautical Engineering Code: Kul-34

Supervisor: Professor Timo Siikonen

Advisor: Johanna Vaittinen, D.Sc.(Tech.)

In many countries, fossil fuels are a vast energy resource in spite of the growing sector of renewable energy resources. Large amounts of carbon dioxide is produced by the power plants using fossil fuels, like coal, peat and natural gas. The CO_2 emissions induce global warming which leads to the fact that releasing CO_2 into the atmosphere is no longer environmentally sustainable.

A common technology for carbon dioxide recovery is the post-combustion carbon dioxide capture via absorption. The post-combustion carbon-dioxide capture is conducted typically in structured packed bed columns which have large dimensions, since the flue gas streams to the absorber are vast. Controlling the flow is difficult due to the large dimensions and maldistribution is a real problem. Maldistribution of the flue gas and the liquid solvent in the absorber bed decrease mass transfer efficiency in the column.

In this work, flow field and gas phase distribution in a structured packed bed column is studied by means of computational fluid dynamics (CFD). A porous medium model for a structured packed bed is created with aid of experimental results from literature. The porous medium model is validated against pressure loss measurements and maldistribution measurements. According to the comparison between simulations carried out and experimental results from literature, the anisotropic porous model seems to capture the gas phase maldistribution quite well. Finally, the model is applied for an industrial scale column, and it is noticed that it is not guaranteed that a severe maldistribution will smooth out even in a relatively high bed.

Flow in a packed bed is complicated, thus simplifications were made in this study. In order to increase the model reliability, some propositions for improving the model are given in the end of this work.

Keywords: computational fluid dynamics (CFD), packed bed, porous medium, anisotropic porous model, OpenFOAM

AALTO-YLIOPISTO
INSINÖÖRITIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Niina Jäntti		
Työn nimi: Kaasuvirtaus strukturoidussa täytekappalepedissä laskennallisen virtausmekaniikan keinoin		
Päivämäärä: 18.10.2013	Kieli: Englanti	Sivumäärä:9+139
Laitos: Sovelletun mekaniikan laitos		
Professuuri: Lentotekniikka		Koodi: Kul-34
Valvoja: Professori Timo Siikonen		
Ohjaaja: D.Sc. (Tech.) Johanna Vaittinen		

Monissa maissa fossiiliset polttoaineet ovat edelleen merkittävä energianlähde huolimatta kasvavasta uusiutuvan energian sektorista. Fossiilisia polttoaineita, kuten hiiltä, turvetta ja maakaasua, käyttävät voimalaitokset tuottavat suuria määriä hiilidioksidia. Hiilidioksidipäästöt edistävät ilmaston lämpenemistä ja sen vuoksi hiilidioksidin päästäminen ilmakehään ei ole ympäristön kannalta kestävä ratkaisu.

Hiilidioksidin talteenotto savukaasuista absorption avulla on yleinen teknologia hiilidioksidipäästöjen vähentämiseksi. Absorptiokolonnit hiilidioksidin talteenotossa ovat tyypillisesti suuria dimensioiltaan suurien savukaasumäärien vuoksi. Virtauksen hallinta suuressa kolonnissa voi olla vaikeaa. Kaasu- ja nestefaasin epätasainen jakautuminen absorptiokolonnissa on ongelma, sillä se heikentää kolonnin aineensiirtotehokkuutta.

Tässä työssä kaasun jakautumista strukturoidussa täytekappalepedissä tutkitaan laskennallisen virtausmekaniikan keinoin. Strukturoidulle täytekappalepedille muodostettiin malli perustuen virtaukseen huokoisen aineen läpi. Malli muodostettiin ja validoitiin kirjallisuudesta löytyneiden kokeellisten tulosten avulla. Simulointi- ja kokeellisten tulosten vertailun perusteella voidaan sanoa, että mallin avulla kaasufaasin jakautuminen voidaan ennustaa. Anisotrooppisen huokoisen aineen mallia täytekappalepedille sovellettiin teollisuusmittakaavaa vastaavaan kolonniin, ja huomattiin, etteivät erot kaasun nopeusjakaumassa välttämättä tasaannu korkeassakaan strukturoidussa täytekappalepedissä.

Virtaus täytekappalepedissä on monimutkainen ja työn lopussa on ehdotettu aiheita, joita tutkimalla mallin luotettavuutta voitaisiin parantaa.

Avainsanat: laskennallinen virtausmekaniikka (CFD), täytekappalepeti, huokoinen aine, anisotrooppinen huokoisen aineen malli, OpenFOAM

Preface

This thesis was conducted as a part of the CLEEN Carbon Capture and Storage Program and was funded by Tekes, the Finnish Funding Agency for Technology and Innovation. The thesis was written at Neste Jacobs Oy in the Competence Center Technology and Process between April 2013 and October 2013. It has been a fascinating and interesting opportunity to be a part of this project.

I would like to thank my advisors Johanna Vaittinen and Tuomo Keskitalo for their guidance, encouragement and interest in this work. I would also like to thank Jussi Laitio, Kari Keskinen and Timo Nyman for guidance in this work and all of their feedback. I would like to acknowledge Professor Timo Siikonen from Department of Applied Mechanics, Fluid Dynamics, for being my supervisor.

I would also like to thank my friends and family for all the support during my studies and this work. Especially, I would like to thank my fiancé Tuomas, for all the love and support.

Kilpilahti, 18.10.2013

Niina Jäntti

Contents

Symbols and abbreviations

Symbols

a	specific area	$\mathrm{m}^2/\mathrm{m}^3$
a_N	matrix coefficient corresponding to the neighbour N	
a_P	matrix coefficient corresponding to the point P	
A	area	m^2
A_p	surface area of the packing	m^2
c	speed of sound	m/s
C_s	gas capacity factor	$\mathrm{Pa}^{0.5}$
C_V	maldistribution factor	
d_h	hydraulic diameter	m
d_{ij}	viscous term coefficient in Darcy-Forchheimer Eqn.	$1/\mathrm{m}^2$
D	diameter	m
f_{ij}	inertial term coefficient in Darcy-Forchheimer Eqn.	1/m
e	internal energy	$\mathrm{m}^2/\mathrm{s}^2$
$e1, e2, e3$	local coordinates	
F	mass flux through a face	
F_S	vapour (gas) capacity factor	$\mathrm{Pa}^{0.5}$
$\mathbf{g}$	body force	m/s^2
h_L	liquid holdup	
h	enthalpy	$\mathrm{m}^2/\mathrm{s}^2$
H	height	m
k	thermal conductivity	W/mK
k	turbulence kinetic energy	$\mathrm{m}^2/\mathrm{s}^2$
$\mathbf{K}$	permeability tensor	
$\dot{m}$	mass flux	kg/s
Ma	Mach number	
M_f	maldistribution factor	
$\mathbf{n}$	normal vector	
N	number of measuring points	
r	radius	m
R	radius	m
R	individual gas constant	J/kgK
Re	Reynolds number	
p	pressure	Pa
s_0	liquid film thickness	m
$\mathbf{S}_f$	face area vector	
S	strain rate tensor	
S_i	source term	
t	time	s
T	temperature	K
u_V	superficial gas or vapour velocity	m/s
u_{V0}	relative gas or vapour velocity	m/s
u_L	liquid velocity	m/s
u_{L0}	relative liquid velocity	m/s

u^+	dimensionless velocity	
$\mathbf{U}$	velocity vector	m/s
V_l	liquid volume	m^3
V_c	column volume	m^3
V_p	packing volume	m^3
$\mathbf{x}$	position vector	
$\mathbf{Q}$	source term	
x, y, z	global coordinates	
y^+	dimensionless distance from the wall	
α	inclination angle	
α	diffusion factor	
α_p	pressure under-relaxation factor	
α_u	velocity under-relaxation factor	
ϵ	void fraction	$\mathrm{m}^3/\mathrm{m}^3$
ϵ^n	convergence error	
λ	eigenvalue	
κ	von Kármán constant	
ν	kinematic viscosity	m^2/s
ν_T	turbulent kinematic viscosity	m^2/s
μ	dynamic viscosity	kg/ms
ω	turbulence kinetic energy specific dissipation	1/s
ϕ	general quantity	
ϕ	porosity	
ψ	eigenvector	
ρ	density	kg/m^3
τ	shear stress tensor	$\mathrm{kg}/\mathrm{m}^2\mathrm{s}^2$

Abbreviations

CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
HETP	Height equivalent to theoretical plate
NTP	Number of theoretical plates
LES	Large Eddy Simulation
RANS	Reynolds Averaged Navier Stokes
RHS	Right-hand-side
SST	Shear stress transport

1 Introduction

In many countries, fossil fuels are a vast energy resource in spite of the growing sector of renewable energy resources. Large amounts of carbon dioxide is produced by the power plants using fossil fuels, like coal, peat and natural gas. A typical volume fraction of carbon dioxide in flue gases from power plants is $4 - 14$ %. The CO_2 emissions induce global warming which leads to the fact that releasing CO_2 into the atmosphere is no longer environmentally sustainable.

One solution to decrease the carbon dioxide emissions is the recovery of carbon dioxide from the flue gases. Several technologies for recovering exists, but post-combustion carbon dioxide capture via absorption can be regarded as one of the most important technologies. The post-combustion carbon-dioxide capture is conducted typically in an absorption column which has large dimensions, since the flue gas streams to the absorber are vast. Additionally, the solvents are usually fed to the absorption column from the top of the column and flue gases from the bottom, which may create a very complex flow field into the column. The controlling of the flow is difficult due to the large dimensions and the maldistribution is a real problem. The maldistribution of the flue gas and the liquid solvent in the absorber bed decreases mass transfer efficiency in the column.

In this work, a flow field and a gas phase distribution in a structured packed bed column is studied by means of computational fluid dynamics (CFD). Structured packed beds are commonly used for example in absorber columns, like post-combustion CO_2 absorber columns.

The research is part of the Carbon Capture And Storage Program (CCSP) of CLEEN Ltd. and Neste Oil Oyj/Neste Jacobs Oy is one of its shareholders. The research within CCSP concentrates on carbon neutral energy production and this study especially on the gas purification with CO_2-amine absorbers.

Currently, CLEEN Ltd. manages six on-going research programs; Distributed Energy Systems (DESY), Efficient Energy Use (EFEU), Carbon Capture and Storage Program (CCSP), Future Combustion Engine Power Plants (FCEP), Measurement, Monitoring and Environmental Efficiency Assessment (MMEA) and Smart Grids and Energy Markets (SGEM). The present study is made under Carbon Capture and Storage Program (CCSP) whose research agenda deals with carbon neutral energy production. [1]

Previously, a lot of CFD simulations for absorber and other columns have been modelled at Neste Jacobs Oy to study for example flow distribution in a column. In those simulations, the packed beds in columns are taken into account as an isotropic porous medium, which cannot capture the maldistribution of the phases in the bed. The flow in the packed bed itself has not been an object of interest, but for example the flow field below the bed. The packed bed modelled with the isotropic porous media has been modelled only to provide the appropriate pressure loss across the bed. With the model presented in this study the effect of the gas maldistribution in the bed can also be taken into account.

In this study, the goal is to analyse the gas phase distribution in the column with means of computational fluid dynamics. It is assumed, that gas maldistribution is an indicative of the liquid maldistribution, too. Packing in the absorption column is considered to be Sulzer's Mellapak 250Y structured packing, which is widely used in absorption columns. This kind of corrugated structured packings have the preferential flow direction in the direction of the channels. This leads to good radial spreading, but can also cause maldistribution of the phases, which lowers the column efficiency. Therefore, it is essential to take the preferential flow directions into account when modelling structured packing.

The calculations are done with open source CFD software package OpenFOAM. The applicability of models describing flow in porous medium for this problem is studied. Calculations for validating the model are made for a laboratory scale column and the results are compared to experimental results from literature. Eventually the calculations with only gas phase are made in an industrial scale column.

2 Carbon Capture and Storage (CCS)

Carbon Capture and Storage (CCS) refers to the capture of CO_2 emissions from the usage of fossil fuels like oil, coal, peat and natural gas. In spite of the increasing amount of renewable energy production, fossil fuels remain a vast energy resource for many countries. The CO_2 emissions induce global warming which leads to the fact that releasing CO_2 into the atmosphere is no longer environmentally sustainable. Therefore, the abatement of greenhouse gases from the combustion and processing of fossil fuels and reduction of the local pollution is an important technological and political issue today. [3]

Carbon capture and storage is applied to fossil fuelled power plants, in industrial processes and in the fuel production and transformation sectors. The CCS chain includes capturing CO_2, transporting and finally storing CO_2 securely in subterranean geological structures [3].

There are three main capturing technologies for CCS: oxyfuel combustion (denitrogenation), pre-combustion capture and post-combustion capture. In oxyfuel combustion the fuel is combusted in oxygen instead of air. This leads to higher concentration of CO_2 in the flue gas and thus the removal of CO_2 is more efficient. This method requires an additional air separation unit in the plant in order to obtain pure oxygen for the process.

In CO_2 pre-combustion capture the fuel is gasified, when carbon monoxide (CO) and hydrogen (H_2) is produced. Gasification is conducted in high temperatures and pressure. The resulting CO and H_2 react with steam and they are shifted into CO_2 and H_2. The CO_2 is removed from the exhaust stream and the H_2 can be used as a fuel for a gas turbine.

In the post-combustion capture the CO_2 is absorbed from the flue gas into the solvent in absorption column. Solvents applied can be for example aqueous amines or chilled ammonia. After absorption, the CO_2-rich solvent is led to a stripper column, where the CO_2 is separated from the solvent by heating. The separated CO_2 is compressed for transport and storage. The regenerated solvent is fed again to the absorption column. In this work, a post-combustion technology is studied.

The best CCS technology for the power plant depends on the type power plant and its fuel characteristics. The post-combustion capture based on chemical absorption is the preferred technology for current coal- and gas fired power plants. For coal fired integrated gasification combined cycle plants the pre-combustion capture with physical absorption would be the best technology. CCS requires energy and therefore it adds the costs and reduces energy efficiency. The expected extra fuel consumption caused by carbon dioxide recovery is $10 - 40$ % and the increment in the costs of the generation of electricity would be $20 - 90$ % [5]. The highest target capture rates are frequently $85 - 90$ % [3]. By improving the CCS technology designs CO_2 capturing can be made more effective and profitable.

3 Structured packing columns

3.1 Column structure

Packed bed columns are typically used in chemical industry for separation processes, e.g. absorption, distillation or stripping. In a packed bed column the most essential element is the packed bed itself, but the column consists of several other elements, too. The column geometry is normally cylindrical or rectangular. In addition to the packed bed, inside the column there are support structures for the packing installation, gas and liquid distributors and liquid collectors. There can be multiple separate packed beds in one column. Fig. 1 shows the structure of a typical packed bed column.

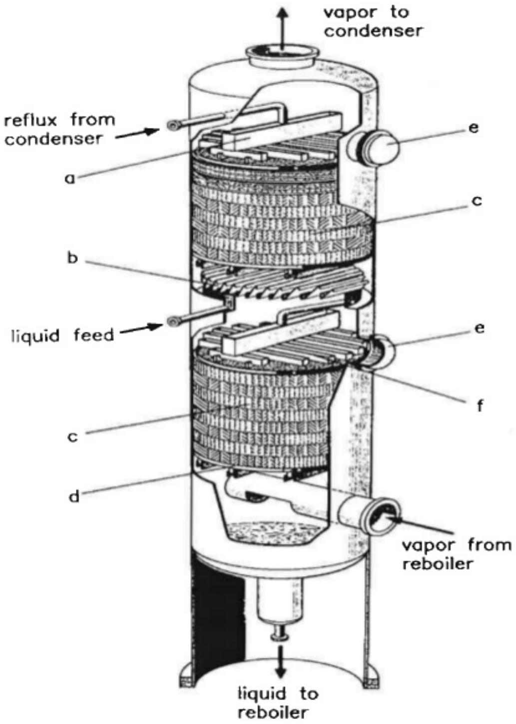


Figure 1: *A packed bed column and its internal structures: (a) liquid distributor, (b) liquid collector, (c) packed bed, (d) support plate, (e) maintenance access, (f) liquid redistributor [7].*

3.1.1 Packings

There are a variety of packings available for separation process applications. The most general packings can be divided into two groups: random and structured packings [12]. Random packings are pieces of packing of a specific geometrical shape like presented in Fig. 2. Random packings are dumped into the column so that they form a packed bed of a random order of packing pieces.

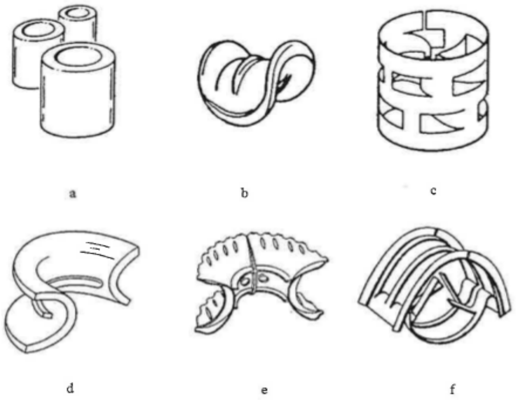


Figure 2: *Random packings: (a) Raschig-rings, (b) ceramic Berl-saddle, (c) metallic Pall-ring, (d) ceramic Intalox saddle, (e) plastic Intalox saddle, (f) IMTP. [6].*

Structured packings are elements of corrugated sheets which form channels as can be seen in Fig. 3. The packing elements are piled into a column above each other. Usually the packing elements are rotated 90° with respect to the layer below in order to obtain varying directions for the channels in the bed.

The structured packing has normally lower pressure drop per mass transfer efficiency in comparison to random packing but random packings are significantly cheaper. For atmospheric processes random packings are competitive but in higher pressure the structured packing is better because of the better efficiency [12].

One example of structured packing is the trademark Mellapak manufactured by Sulzer Chemtech. Mellapak type structured packing is widely used in petrochemical, oil and gas industries and exhaust air cleaning. Mellapak 250Y structured packing shown in Fig. 3 is selected for this work because of its usage for example in CO_2 absorber columns. The number 250 refers to the specific area of the packing (Chapt. 3.3.1) and letter Y to the inclination angle of the channels.

The channels in structured packings are inclined to a certain angle which is denoted by X and Y. An X-type channel is inclined at 60 degrees to the horizontal line and Y-type at 45 degrees correspondingly. The greater the inclination angle, the lower the pressure drop. Flow path is in case of 60 degrees inclination shorter than in 45 degrees inclination. Therefore efficiency in case of greater inclination angle is lower compared to less inclined packing [13, p.13]. The separation efficiency of Mellapak 250Y is presented in Fig. 4.

Sulzer has a variety of Mellapak-type structured packings for different applications, which differ from each other mainly by specific area of the packing. There are packings available manufactured of various materials, which can be chosen according to the application. [17]

Figure 3: *Mellapak 250Y structured packing from Sulzer [17].*

Table 1: Mellapak 250Y properties.

Specific area of the packing	a	250 m^2/m^3
Void fraction	ϵ	0.96
Channel inclination	α	45°

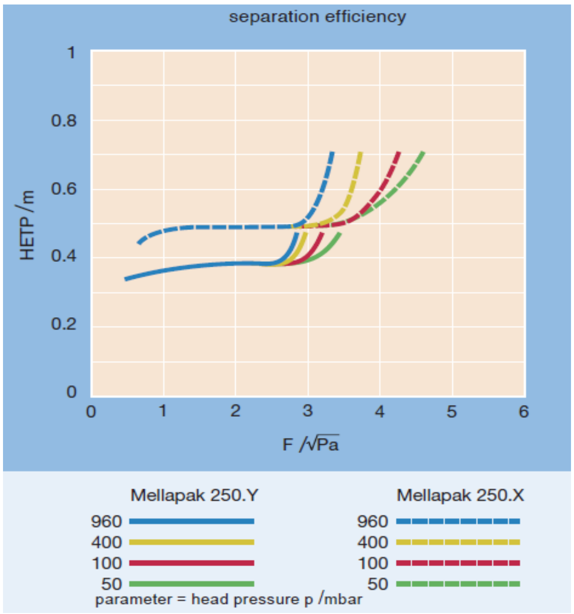


Figure 4: *Separation efficiency of the Mellapak 250Y and 250X [17]. Explanations for F and HETP are in Chapt. 3.3.*

3.2 Operation

The columns are usually operated co-current or counter-current. In both cases the liquid distributor is at the top of the column. In a counter-current operating column, which is mainly applied in columns for separation processes, the gas inlet is at the bottom of the column. In co-current operating column the gas inlet is at the top of the column.

For example, in the case of CO_2-amine absorber, flue gas containing CO_2 and lean amine solvent flow through the packed bed. It is a counter-current flow system as the flue gas is fed from the bottom and amine solvent from the top of the column. The purpose of the packing is to provide the column with a maximal contact area between the phases, as the liquid flows as a film along the packing surface and gas flows in the middle of the channels. The CO_2 is absorbed to the solvent and the rich solvent flows down to the bottom of the column and is collected and routed to the stripper. In the stripper, CO_2 is removed from the rich solvent and collected. The lean solvent is led again to the top of the absorber column. The purified gas from the absorber can be vented to the atmosphere from the top of the column. A schematic presentation of the CO_2 absorber-stripper system is shown in Fig. 5, where the absorber is referred as scrubber.

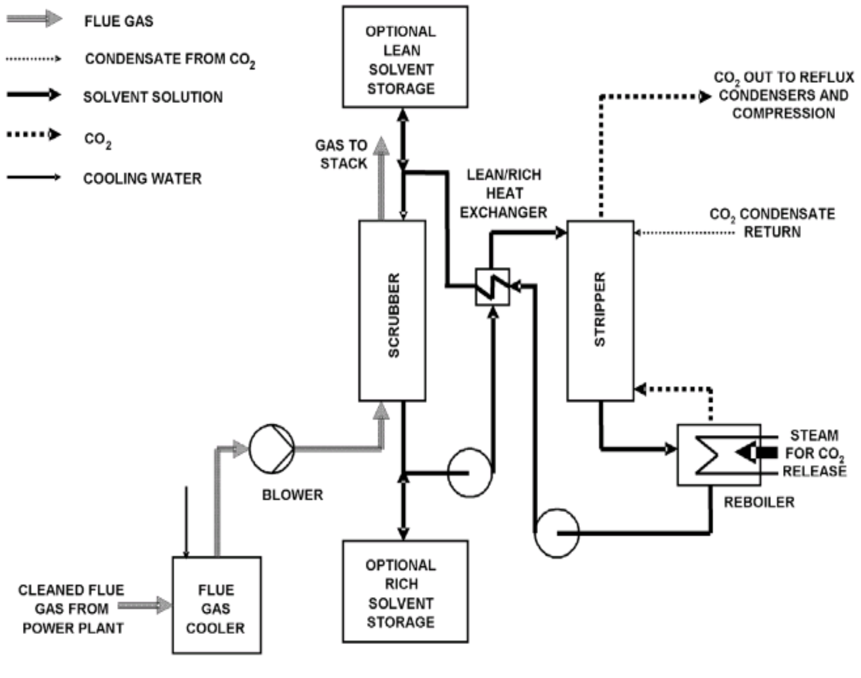


Figure 5: *A schematic figure of a CO_2 absorber-stripper system [8].*

3.3 Hydrodynamics of the packed bed columns

3.3.1 Key characteristics

In this chapter some of the essential quantities that are characteristic for packed bed columns and their hydrodynamics are presented. The flow system in packed bed columns can be either co-current or counter-current, as explained in Chapt. 3.2. In the ideal case the liquid flows as a film of an uniform thickness s_0 along the packing surface and gas flows in the middle of the channels.

Specific area a is the ratio of the surface area of the packing to the volume of the bed

$$a = \frac{A_p}{V_p} \tag{1}$$

where A_p is the total area of the surface presented by the bed of the packing and V_p is the packing volume [15]. The void fraction of the packing ϵ can be calculated from

$$\epsilon = \frac{V_c - V_p}{V_c} \tag{2}$$

where V_c is the column volume. According to Ref. [12], hydraulic diameter for the packing can be defined in a following way with the void fraction of the packing ϵ and the specific area a:

$$d_h = 4\frac{\epsilon}{a} \tag{3}$$

Hydrodynamics of different packings used in different systems are compared usually with gas capacity factor C_s instead of the gas velocity u_V

$$C_s = u_V \sqrt{\frac{\rho_V}{\rho_L - \rho_V}} \tag{4}$$

At low pressures ρ_V is low compared to ρ_L. Values of $\sqrt{\rho_L}$ are very similar in case of practically all liquids. Therefore, Eqn. (4) can be simplified to the following expression, which is also called gas capacity factor.

$$F_S = u_V \sqrt{\rho_V} \tag{5}$$

A pressure drop is often used as a parameter for a comparison of different packings, because the pressure drop causes usually most of the operating costs for an absorption process for example. This is because the energy costs of the blower needed to cover the pressure drop. The pressure drop is usually presented as a function of the gas capacity factor, as

$$\frac{\Delta p}{H} = f(F_S) \tag{6}$$

where Δp is the pressure drop, H is the packed bed height and F_S is the gas capacity factor defined by Eqn. (5) [12]. When a packed bed column is designed, a maximum pressure loss is defined. From the pressure loss the minimum diameter with a chosen packing can be estimated by using pressure drop correlations. There are several pressure drop correlations available, but often the restrictions of the correlations are unknown which makes the application of them difficult. Further information on pressure drop correlations is found for example in Ref. [6].

3.3.2 Liquid holdup

In case of a counter-current operated column, liquid flows in a film of a thickness s_0 at velocity u_L counter-current to the gas flow with velocity u_G. The thickness of the liquid film can be approximated with liquid holdup and specific area of the packing

$$s_0 = \frac{h_L}{a} \tag{7}$$

where a is the complete wetted area of packed surfaces. In the case that surfaces are not completely wetted a is the hydraulic area a_h [15]. The liquid holdup h_L is the volume of the liquid V_l in the column divided by the whole packing volume V_p

$$h_L = \frac{V_l}{V_p} \tag{8}$$

The liquid holdup consists of static and dynamic liquid holdups. Static holdup $h_{L,s}$ means the liquid staying in the packing because of the capillary forces when irrigation is stopped. Dynamic holdup $h_{L,d}$ stays in the packing because of the resistance forces. The dynamic holdup increases as the liquid superficial velocity u_{L0} or packing surface area increases and it decreases with increasing liquid density [13].

Fig. 6 represents the liquid holdup h_L as a function of gas load u_V. It can be seen that liquid holdup remains constant with an increasing gas load until a *loading* point $u_{V,S}$, when the liquid starts to increase with gas velocity. After that the liquid holdup increases until the maximum value, which is the *flooding* point $u_{V,Fl}$, is reached. *Flooding* means that liquid fills the whole column at some cross-section or fills the flow channels in a part of the column, which makes the operation of the column difficult. The gas velocity at loading point $u_{V,S}$ is approximately 70 % of the gas velocity at the flooding point $u_{V,Fl}$ [15]. For example in Fig. 6 the loading point is 65 % of the gas velocity at the flooding point.

If a column with a two-phase counter-current flow system is operated at a loading range between the loading and flooding points, the downward stream of liquid is no longer independent of the gas load because it is held up by the shear forces in the gas stream. According to experiments below the loading point $u_{V,S}$, the gas flow does not have a significant effect on liquid holdup [15].

The correlations for obtaining the liquid holdup for arbitrary operation conditions can be divided into two groups: correlations below and above the loading point. Liquid holdup correlations are applied to estimate the residence time for mass transfer efficiency calculations or liquid holdup effect on the pressure loss and flooding [6]. Further information on the liquid holdup correlations can be found in Ref. [6].

The liquid phase influences the hydrodynamics of the gas phase in two ways. First, an increasing liquid holdup increases the gas velocity according to the following expression

$$u_V = \frac{u_{V0}}{1 - \epsilon - h_L} \tag{9}$$

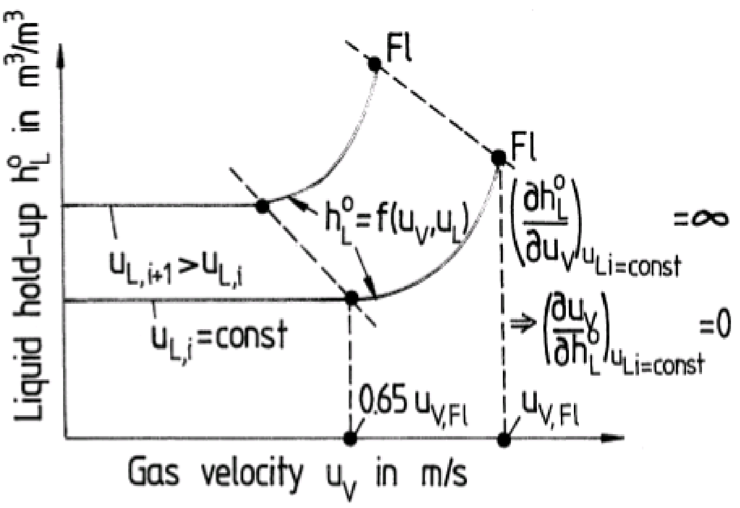


Figure 6: *Liquid holdup as a function of gas velocity u_V [14].*

Second, the pressure drop in a dry packing depends on the relative gas velocity u_{V0} but in an irrigated packing it depends on the sum of the gas velocity and the liquid velocity at the interface of the gas phase $u_{V0}+u_{L0}$. Because the gas velocity is usually many times greater than the liquid velocity, the second effect can be neglegted. [12]

3.4 Mass transfer

The number of theoretical stages or plates N is one parameter used to determine the effectiveness of the columns. The theoretical plate is the apparatus volume where the concentration of the component at the outlet is equal to the equilibrium concentration at the inlet i.e. a height after which the phases are in vapour-liquid thermodynamic equilibrium. The height of the packing can be calculated by the equation

$$H = NTP \cdot HETP \tag{10}$$

where NTP is the number of theoretical plates determined for example by the McCabe-Thiele diagram and $HETP$ is the height equivalent to theoretical plate, which is determined for different packings [12]. $HETP$ can be estimated with correlations and some correlations are presented in Ref. [6]. In Fig. 7 an example of a McCabe-Thiele diagram is presented and a method for obtaining the number of theoretical stages is explained in Ref. [15, p.17-19].

In Fig. 8 $HETP$ is presented as a function of the vapour rate C_s. The region from point (B) to (C) presents the constant separation efficiency. As C_s increases from point (C), the vapour flow starts to interact with the liquid phase. This interaction improves the packing mass transfer efficiency because of the interaction of phases. The efficiency increases until the point (E), where $HETP$ has its minimum. After point (E) the additional axial mixing of the liquid phase because of the increasing vapour flow cause the reduction of the efficiency. It can be seen that column can

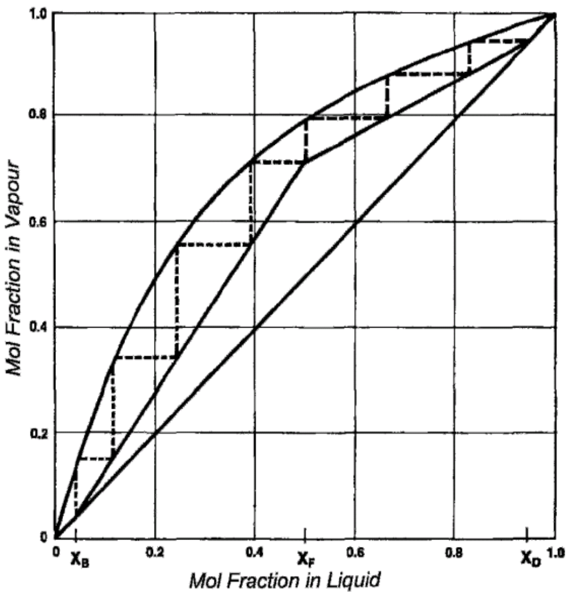


Figure 7: *McCabe-Thiele diagram [12].*

operate up to the flooding point (F) but the maximal efficiency is reached in point (G).

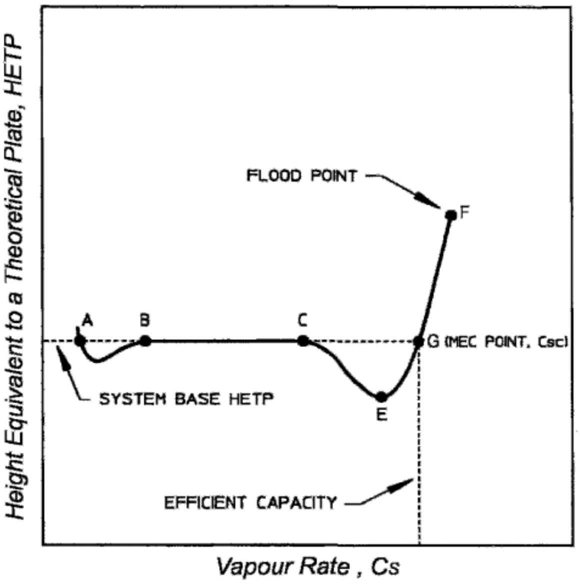


Figure 8: *Height equivalent to theoretical plate as a function of vapour rate [12].*

The intensity of the mass transfer processes G in Eqn. (11) can be measured with the amount of substance transferred in a unit of packing volume per unit of time

$$G = K_G a_e V_p \Delta_G = K_L a_e V_p \Delta_L \tag{11}$$

where K_G and K_L are overall mass transfer coefficients for gas- and liquid side controlled mass transfer, a_e effective packing surface area, V_p packing volume, Δ_G and Δ_L the driving forces of the mass transfer process calculated by the concentration of the gas/liquid. To increase the G-value for a volume V_p, the values of K_G (K_L) and Δ_G (Δ_L) must be increased. The flow rates for a specific apparatus are fixed preliminary so the driving forces cannot be used for intensification of a given process. Increasing the gas and the liquid velocities improves the partial mass transfer coefficients, which in turn increases the overall mass transfer coefficient and a_e but it also increases the pressure drop, which has great effect on the operating costs. A velocity increase is also limited by the loading and flooding points. The only possibility left for intensification is a more effective packing.

To have good mass transfer characteristics the packings in the packed bed column should allow operation at high gas and liquid superficial velocities at small hydraulic diameter of the packing. The packing should break the liquid film frequently and coalescence and break drops and jets in the free volume of the packing. As stated earlier, the energy necessary for the blower to cover the pressure drop is usually the main part of the energy operating costs for absorption processes. Therefore, the pressure drop should be as low as possible. Packing material must also endure in a corrosive environment and process temperatures.[12]

3.5 Distribution of the phases in the packing

Most of the theoretical models for calculating packed beds are valid for the case where the distribution of phases over the cross-section of the column is uniform. However, in real columns the phases are unevenly distributed. The maldistribution of the phases has an influence on the effectiveness of the column.

The uniform distribution of two phases in a column means normally uniform distribution of the average superficial velocities of the phases. The spatial distribution of the phases is calculated for specific local packing cross-sections which corresponds the dimensions of the channels formed in the packing. The maldistribution in the packed bed can be divided in two types. The first maldistribution type, small-scale maldistribution, is the real maldistribution of the phases over specific cross-sections. The first type maldistribution depends on the geometry of the packing, but not on the column design. The second maldistribution type, large scale maldistribution, depends on the design of the column and is caused by the initial maldistribution of the phases over the cross-section of the packing and the effects of the column wall [12].

3.5.1 Liquid maldistribution

Liquid maldistribution in packed beds occurs because of the channeling of the liquid flow. Also walls induce maldistribution in liquid flow, since the liquid tends to flow along the wall. The walls of the column have an effect on mass transfer even with a large diameter column: the liquid to gas ratio is much greater under wall effect than in the main packing. A wall effect can be eliminated by installing a horizontal wiper band into the column, which leads the liquid flowing down the wall back into the packing. Liquid maldistribution reduces the column efficiency, since the phases do not interact effectively. Therefore, in industry the liquid is redistributed at specific length intervals along the packed column.

The maldistribution of liquids in packed bed columns is investigated by many researchers. For example, the results of Stikkelman [9] are presented in Fig. 9, where the distribution of the liquid phase under Mellapak 250Y in two different cases is shown. In Fig. 9, the upper part shows the situation under the loading point and the lower part over the loading point. It can be seen that above the loading point the liquid maldistribution increases strongly.

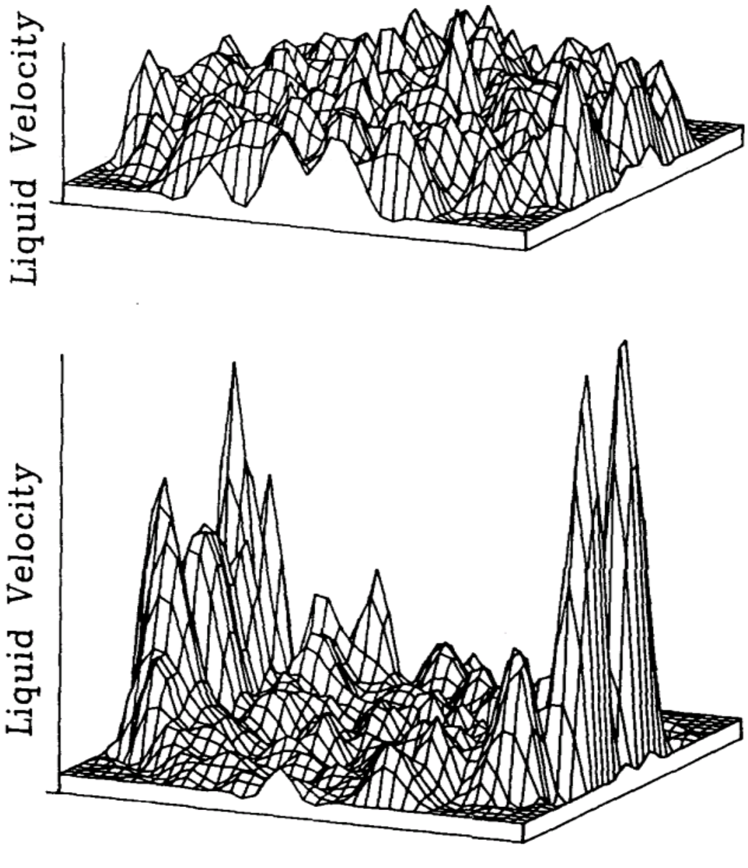


Figure 9: *The liquid velocity profile leaving the bottom of the Mellapak 250Y packing (4 elements of packing) at a gas velocity of* 2 m/s, $M_f = 0.26$ *(upper part of the figure) and of* 3.2 m/s, $M_f = 1.74$ *in the lower part of the figure. [9].*

The effect of the liquid distribution on the column efficiency is studied for example in Ref. [11]. The results indicate that maldistribution of the liquid phase causes reduction in efficiency for both random and structured packings. For example, maldistribution induced by a chordal blockage caused approximately 30 % reduction in efficiency. Quality of the initial liquid distribution is noticed to be critical for column efficiency and therefore liquid distributor designs have been improved over the years.

In this work, the liquid phase is not modelled because the multiphase fluid dynamics calculations are computationally too expensive for a limited schedule and also because lack of the experimental results for validation. However, studying the gas phase maldistribution by CFD provides input data for other modelling tools, such as Aspen Plus [10], which in turn can be used to predict the influence of distorted gas distribution on the final efficiency of CO_2 capture.

3.5.2 Gas maldistribution

In laboratory- and pilot-scale packed columns the gas distribution can be regarded as uniform. However, in industrial scale when columns are of large diameter more than 3 m [24], maldistribution of the gas phase may occur, particularly if the gas is led into the column in radial direction [15]. Laboratory scale columns are usually less than 0.5 m in diameter [24] and pilot scale columns less than 3 m. The difference in liquid and gas flows is that the flow of the liquid phase is due to gravity and gas phase due to pressure drop. Gas maldistribution can be investigated by using statistic methods, theoretical models and experiments.

Maldistribution factor M_f for the gas flow characterizes the divergence between the real flow and the ideal bulk flow [12]

$$M_f = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{u_i - \bar{u}}{\bar{u}}\right)^2} \tag{12}$$

where N is number of the measuring points in the cross-section, u_i is gas flow at point i, $\bar{u}$ is the mean value of the velocity in a given cross-section.

Another way to characterize maldistribution is a coefficient of variation of the velocity measurements which is a measure for the magnitude of the maldistribution [27]

$$C_V = \left[\frac{1}{A_t}\sum_{i=1}^{N} A_i \left(\frac{u_i - \bar{u}}{\bar{u}}\right)^2\right]^{0.5} \tag{13}$$

where A_t is the cross-sectional area, A_i area of a cell, u_i the local velocity, N the total number of cells and overall mean velocity is obtained from

$$\bar{u} = \frac{1}{A_t}\sum_{i+1}^{N} A_i u_i \tag{14}$$

If C_V approaches zero, the flow is uniform bulk flow.

The quantity penetration depth PD refers to the height after which the value of the M_f remains constant in the packing. According to penetration depth it can be determined at which packing height the influence of the second type of maldistribution for the gas phase can be neglected and the gas maldistribution in the packing is only of the first type [12]. According to [18] the value of M_f is after penetration depth $M_f = 0.014 - 0.018$. For example the experimentally obtained value of the penetration depth for Mellapak 250Y is $PD = (0.3 - 0.4)$ m when the gas distribution device is a ring form inlet and $PD = (0.6 - 0.8)$ m when usual pipe is used for gas distribution [19].

The experimental results on the gas flow maldistribution are differing in the conclusion whether the maldistribution of the gas phase affects on the efficiency and operation of the column or not. The scale of the column may be one thing which affects on the difference in the results. In the small scale the effect of the maldistribution may not be so obvious as in the larger scale columns.

The effect of the maldistribution of gas phase to the efficiency of the column was studied by Cai et al. [21] and according to them the vapour maldistribution in a column of 1.22 m diameter does not have a significant effect on packing efficiency.

Olujić, Haaring and van Baak [20] made experiments in a column of 1.4 m inner diameter concerning the relation between a severe initial maldistribution in gas flow and its effect on the structured packed bed hydrodynamics. The packing used was Montz-pak B1-250. The initial maldistribution to the gas flow was obtained with a chordal plate blockage and central blockage, which simulate the real obstacles and initial distributions in the column which may occur in the column. The geometries of the blocking plates are seen in Figs. 55 and 57. According to the results the gas maldistribution generated by the chordal blockage penetrates deep into the structured packed bed, especially in the case where the orientation of the flow channels is parallel to the side of the chordal blockage. The maldistribution in a gas flow causes a premature loading of the bed. In case of premature loading there is a larger pressure drop at design load than normally. According to the results the gas maldistribution affects the liquid phase in the lowest part of the bed by forcing the liquid phase to get maldistributed too, which could reduce the mass transfer efficiency in the column. Though the orientation of the flow channels of the packing perpendicular to the side of the chordal blockage in the first layer of the bed seem to smooth out the initial gas maldistribution faster, since in that case the gas penetrates within one layer into the blocked area of the bed. This also considered the reason why in Ref. [21] the chordal blockage did not cause reduction in efficiency of the column. [20] According to [27] the maldistribution of the gas phase leads also to the increasing of the pressure drop across the bed as is shown in Fig. 10, where the most severe maldistribution is induced by chordal blockage at the inlet.

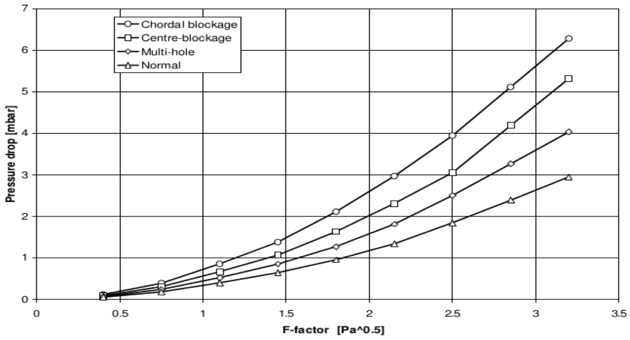


Figure 10: *Pressure as a function of the gas capacity factor F in case of different blockages at the inlet. [27].*

4 Computational fluid dynamics

4.1 Fundamental equations

Mathematically the fluid flow can be presented with the following equations which are in their conservation form in Eqns. (15)-(17). They can be derived from simple physical principles.

Continuity equation Eqn. (15) states that mass is conserved, i.e. mass that flows in, also flows out. Continuity equation includes the time derivative and convection term, but no source terms due to its nature.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{U} = \mathbf{0} \tag{15}$$

Momentum equation arises from the Newtons second law i.e. $\mathbf{F} = m\mathbf{a}$. Momentum equation is called after its inventors Navier-Stokes equations, but commonly in the literature the name refers to the whole equation group Eqns. (15) - (17). Momentum equation includes a time derivative, convection term, pressure gradient, diffusion term and body force term, which can be for example gravity.

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \rho \mathbf{U}\mathbf{U} + \nabla p = \nabla \cdot \tau_{ij} + \rho \mathbf{g} \tag{16}$$

Energy equation Eqn. (17) comes from the first law of thermodynamics which states that energy can change from one form to another but not vanish. Energy equation contains the time derivative, combined convection term and pressure gradient, heat conduction term and friction term.

$$\frac{\partial \rho (e + |\mathbf{U}|^2/2)}{\partial t} + \nabla \cdot \rho \mathbf{U}(e + |\mathbf{U}|^2/2 + p/\rho) = \nabla \cdot k \nabla T + \nabla \cdot (\mathbf{U} \cdot \tau_{ij}) \tag{17}$$

where the internal energy and pressure divided by density can be expressed as fluid enthalpy h

$$h = e + \frac{p}{\rho} \tag{18}$$

The right-hand side of the Eqns. (15) - (17) describes the effect of the fluid viscosity and if it is removed, the equation group is called Euler equations, which describes an inviscid flow. The complete derivation of Eqns. (15)-(17) is presented for example in Ref. [31].

For a newtonian fluid, for which linear relationship between shear stress and velocity gradient satisfies, the viscous stresses are

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \lambda (\nabla \cdot \mathbf{U}) \tag{19}$$

where δ_{ij} is Kronecker delta

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{20}$$

In index form Eqns. (15) - (17) can presented in the following way

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_k)}{\partial x_k} = 0 \tag{21}$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)\right] + \rho g_i \tag{22}$$

$$\frac{\partial[\rho(e + u_k u_k/2)]}{\partial t} + \frac{\partial}{\partial x_j}[\rho u_j(e + u_k u_k/2 + p/\rho)] = \frac{\partial^2 kT}{\partial x_j \partial x_j} + \frac{\partial u_i \tau_{ij}}{\partial x_j} \tag{23}$$

where τ_{ij} is defined in Eqn. (19). Index i refers to the direction of the equation and indices j and k are dummy indices that vanish according to the Einstein's summation convention. In case of an incompressible flow the energy equation is not used unless temperature is solved. The compressibility of the flow depends on the Mach number (Eqn. (24)).

$$Ma = \frac{u}{c} \tag{24}$$

where c is the speed of sound in corresponding fluid and u is the velocity of the fluid. When the following condition holds the flow is considered incompressible:

$$Ma < 0.1 - 0.2 \tag{25}$$

If the flow is compressible the connection between the pressure, the density and the temperature is determined through an equation of state, for example the ideal gas law

$$p = \rho \frac{R}{M} T \tag{26}$$

where R is the universal gas constant and M the molar mass. If temperature differences are large enough to have a considerable effect on the viscosity of the fluid temperature, the influence can be approximated for example using Sutherland's law [31].

4.2 Numerical solution

4.2.1 Discretization

The mathematical model describing the fluid flow must be discretised in order to obtain algebraic equations for variables in a discrete space and time domain. For example in a finite volume method the discrete computational domain is called a mesh. The approximation of the fluid flow equations depends on the discretization method of the computational domain.

The time domain is discretized in case of unsteady flows. Unsteady flows are parabolic of their nature, since time goes only forward and there cannot be backward effects. Therefore, the time integration methods are initial value problems and a solution is obtained by advancing step-by-step in time domain.

The main space discretization approaches for the governing equations are finit-difference, finite-volume and finite-element methods. The finite-difference method is based on the approximative equations with the help of definition of the derivatives. Taylor series expansion or polynomial fitting is applied in order to approximate the derivatives. The finite-difference method can be applied to any grid type, but is normally used in case of a structured grid.

In finite-element methods the computational domain is discretized into finite elements. The equations are multiplied with a weight function before integration over the domain. The solution is approximated by a shape function in each element in a way that the solution is continuous across element boundaries. The idea is to minimize the approximation error when fitting the shape functions to the governing equations. [32]

In the finite-volume method the computational domain is discretized into control volumes called cells, which have the computational node in the centre of the control volume. The governing equations in their conservation form are integrated over the volume and integrals are presented as surface integrals with the aid of Gauss divergence theorem

$$\int_V (\nabla \cdot \mathbf{F}) dV = \int_S (\mathbf{F} \cdot \mathbf{n}) dS \tag{27}$$

As an example, the finite volume method is applied to a general conservation equation corresponding to the conservation equations governing the fluid flow. Conservation equation for the general scalar quantity ϕ is represented in the following way

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot \rho \mathbf{U} \phi = \nabla \cdot \alpha \nabla \phi + q_\phi \tag{28}$$

and the integrated form of Eqn. (28) is

$$\int_V \left(\frac{\partial \rho \phi}{\partial t} + \nabla \cdot \rho \mathbf{U} \phi \right) dV = \int_V \left(\nabla \cdot \alpha \nabla \phi + q_\phi \right) dV \tag{29}$$

Applying the Gauss theorem (27) the following form is obtained

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \int_S \rho \mathbf{n} \cdot \mathbf{U} \phi dS = \int_S \alpha \nabla \phi \cdot \mathbf{n} dS + \int_V q_\phi dV \tag{30}$$

The concept of flux $\mathbf{f}$ is introduced and Eqn. (30) can be expressed in the following form

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \int_S \mathbf{f} \cdot \mathbf{n} dS = \int_V q_\phi \tag{31}$$

where the flux $\mathbf{f}$ consists of convective and diffusive fluxes. The values for the fluxes $\mathbf{f}$ at the cell faces can be reconstructed from the values of the cell centres, because the values are always recorded at the cell centres in the co-located grid approach which is applied for example in OpenFOAM. The numerical scheme for a discretization depends on the mesh. The discretized form of Eqn. (31) is

$$V_i \frac{\rho_i \phi_i}{\Delta t} + \sum_f \mathbf{f}_f \cdot \mathbf{n}_f S_f = q_{\phi i} V_i \tag{32}$$

where the normal vector $\mathbf{n}$ defines the direction outwards from the computational cell. The flux $\mathbf{f}_f$ is defined in the following way

$$\mathbf{f}_f = \rho_f u_f \phi_f - \alpha_f (\nabla\phi)_f \tag{33}$$

In order to obtain an implicit equation group the fluxes are linearised with respect to the variable being solved

$$\mathbf{f}_f^{n+1} = \mathbf{f}_f^n + \sum_i \frac{\partial \mathbf{f}_f}{\partial \phi_i} \phi_\mathbf{i} \tag{34}$$

where ϕ_i is the variable being solved.

In the finite-volume method, the quantities are specified at the cell centres, but fluxes are calculated at cell faces. Therefore, the cell-centre values must be interpolated with an appropriate method to the cell face. There are various methods for the interpolation. The mathematical interpolation method should be chosen according to the physical nature of the term in governing equations to assure proper behaviour of the modelled flow. For example, the diffusion term should be discretized with the central difference and the convection term with some upwind biased method. Interpolation is a part of the discretization. Next the three interpolation methods used in this study are presented.

The upwind interpolation is the simplest interpolation scheme for the convection term. The upwind interpolation scheme interpolates the upwind cell node value to the cell face under consideration. Upwind method is robust but produces much numerical diffusion, since it is first-order accurate in space. Therefore, it should not be used for actual calculations, but is often a good method to conduct initializing calculations because of its robustness:

$$\phi_f = \begin{cases} \phi_P, & \text{if } f > 0 \\ \phi_N, & \text{if } f < 0 \end{cases} \tag{35}$$

The linear interpolation method can be applied to any term in the governing discretized equations. In the linear interpolation method a linear interpolation between cell centres is applied to obtain the value at the cell face in the following way [33]

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N \tag{36}$$

$$(\nabla\phi)_f = f_x (\nabla\phi)_P + (1 - f_x)(\nabla\phi)_N \tag{37}$$

where f_x is defined in the following way, which can also be noticed in Fig. 11

$$f_x = \frac{|\mathbf{x}_f - \mathbf{x}_N|}{|\mathbf{d}|} \tag{38}$$

where indices P and N refer to the adjacent cells according to Fig. 11. The linear interpolation method is also called central differencing, which is the corresponding method in finite difference discretization method. The linear interpolation is second-order accurate in space, which is noticed by applying Taylor series expansion of ϕ_P about the point P [49].

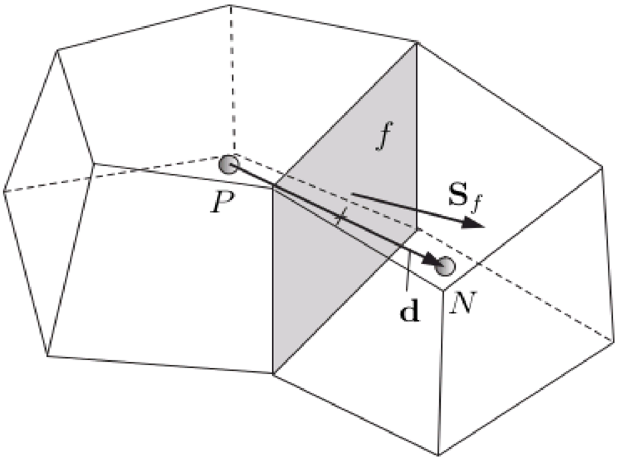


Figure 11: *Parameters in the finite-volume discretisation of OpenFOAM [50].*

The upwind linear interpolation is a second-order accurate upwind biased method. Upwind linear interpolation in a structured mesh is written in the following way.

$$\phi^{-} = \frac{3\phi_i^n - 4\phi_{i-1}^n + \phi_{i-2}^n}{2\Delta x}, \text{ if } f > 0 \tag{39}$$

$$\phi^{+} = \frac{-\phi_{i+2}^n + 4\phi_{i+1}^n - 3\phi_i^n}{2\Delta x}, \text{ if } f < 0 \tag{40}$$

Some terms, like the diffusion term in Eqn. (33), contain a gradient on the face. In case of an orthogonal grid the gradient is calculated as the difference between adjacent cell centre values divided by the distance between the cell centres. In case of a non-orthogonal grid the gradients are calculated in each cell centre and interpolated onto the face under consideration. This gradient is applied to correct the normal gradient calculated according to the method used with orthogonal grid [50, p. 34].

When solving a numerical problem, the solution may have too large values locally due to the interpolation of the values on the cell face. Therefore, in practice flux limiters are usually applied in numerical methods.

4.2.2 Solving linear equations

The discretization and linearisation of the governing equations of fluid flow Eqns. (15)-(17) leads to a linear equation group. In case of fluid dynamics, the coefficient matrices are usually sparse, and therefore the direct methods for solving the system, like Gauss elimination or LU-decomposition, are not computationally efficient. Iterative methods are commonly applied in fluid dynamics because they are usually more efficient for sparse matrices. A general linear equation system is written in the following way

$$A\phi = \mathbf{Q} \tag{41}$$

where A is the coefficient matrix of the linearized equation system, ϕ is the unknown quantity and $\mathbf{Q}$ is the source term. In equation system Eqn. (41) ϕ satisfies the equation system. In case of iterative methods the approximate solution ϕ^n does not satisfy the equations after n iterations, thus the equation system is expressed by the following way

$$A\phi^n = \mathbf{Q} - \rho^n \tag{42}$$

where the superscript n indicates the iteration rounds and ρ^n is the non-zero residual i.e. difference to the correct solution. The iteration procedure drives the residual towards zero.

There are several iterative methods appropriate for fluid dynamics calculations. For example TDMA method is very effective for a set of tridiagonal equations. However, tridiagonal equation matrices appear only in one-dimensional cases. Iterative methods for three dimensional cases are for example Jacobi method, successive over-relaxation method (SOR), incomplete LU decomposition i.e. Stone's method, alternating direction implicit (ADI) method and conjugate gradient methods. As an example, one kind of SOR method, Gauss-Seidel method, is presented in Eqn. (43)

$$\phi_P^{n+1} = \frac{\mathbf{Q}_P - \sum\limits_{nb} A_{nb}\phi_{nb}^n}{A_P} \tag{43}$$

where A_P refers to the diagonal coefficients of the matrix, and A_{nb} to the neighbour cells. In the Gauss-Seidel method, the values of ϕ_{nb}^n are updated immediately after solving. The Jacobi iteration method is similar to method Eqn. (43), but the values of ϕ_{nb}^n are updated after solving all the neighbour cell values in one iteration round [32].

To solve the equation system Eqn. (41) with a multigrid method, a small number of iterations is conducted with an iterative method such as Gauss-Seidel. This routine is called smoothing, while for example Gauss-Seidel method produces a smooth error: the errors corresponding high frequencies are reduced, but the errors corresponding low frequencies remain after few iteration rounds. The current solution can be mapped onto the coarser grid, where the errors corresponding low frequency reduce, while on the coarser grid they correspond high frequency. This procedure can be performed for example in a V-cycle, i.e. first going from finer to coarser grids and then back from coarse to fine grid. Transferring a discrete function from the coarse grid to the fine grid is called *prolongation* and transferring from fine grid to coarse grid is called *restriction.* [34]

4.3 Turbulence

In turbulent fluid flow the flow quantities oscillate around their local mean value. A turbulent flow contains eddies, which in their smallest length scale dissipate into heat energy. The energy spectrum describes the turbulent energy distribution from the largest to smallest eddies. Due to its chaotic nature turbulence enhances mixing, diffusion and dissipation. For example in case of duct flows, turbulence increases the friction coefficient and pressure losses. Whether the fluid flow is turbulent or not is determined with the help of Reynolds number

$$Re = \frac{ud}{\nu} \tag{44}$$

where u is an average velocity, d is a characterizing length, for example diameter for ducts and chord length for airfoils and ν is the kinematic viscosity. At small Reynolds numbers the fluid flow is laminar and at high Reynolds numbers turbulent. Transition from laminar to turbulent flow occurs between these two regions but cannot usually be accurately predicted. The Reynolds number where transition occurs is called the critical Reynolds number Re_{cr}. According to reference [30, p. 279], the critical Reynolds number for internal flows, like flows in ducts, is approximately $Re_{cr} \approx 2100 - 4000$. For an external flow around an immersed body the corresponding critical Reynolds number is $Re_{cr} \approx 2 \times 10^5 - 3 \times 10^6$ [30, p. 340].

4.3.1 Turbulence modelling

As discussed in Chapt. 4.3.1, turbulence has often a significant effect on the fluid flow which needs to be predicted for example for equipment design purposes. Turbulence can be studied experimentally, but often experiments are difficult and time consuming. Turbulence can be modelled with different methods, such as correlations, Reynolds averaged Navier-Stokes (RANS), large-eddy simulation (LES) or direct numerical simulation (DNS).

When considering turbulent flow, the flow quantities can be divided into two components, a mean value and time-dependent fluctuations around it.

$$\rho u_i(t) = \rho \bar{u}_i + \rho u_i'(t) \tag{45}$$

In case of RANS-based turbulence modeling Eqn. (45) is substituted into the equations governing the fluid flow (15), (16) [28]. The equations obtained after substitution are time-averaged and the following equations are obtained

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho}\bar{u}_i}{\partial x_i} = 0 \tag{46}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \bar{\rho}\bar{u}_i\bar{u}_j}{\partial x_j} + \frac{\partial \bar{p}}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} - \frac{2}{3}\delta_{ij}\frac{\partial \bar{u}_k}{\partial x_k}\right)\right] + \frac{\partial}{\partial x_j}(-\overline{\rho u_i' u_j'}) \tag{47}$$

Eqns. (46) and (47) are called Reynolds averaged Navier-Stokes equations (RANS). Even if the mean value of the fluctuations is $\overline{u_i'(t)} = 0$, time-averaging leads to an additional term because of the non-linearity of the convection term. Thus, the equations obtained Eqns. (46) and (47) differ from the original Eqns. (15), (16) by six terms, $(-\overline{\rho u_i' u_j'})$, which are called the Reynolds stresses. These terms are unknown and need to be modelled.

RANS-based turbulence models can be divided into three groups: linear eddy viscosity models, non-linear eddy viscosity models and Reynolds stress models. Eddy viscosity models assume that turbulence is isotropic. In eddy viscosity approach the

Boussinesq hypothesis is employed to calculate the Reynolds stress and close the equations. Boussinesq hypothesis is written

$$-\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\left(\rho k + \mu_t\frac{\partial\bar{u}_k}{\partial x_k}\right)\delta_{ij} \tag{48}$$

It is noticed that according to the Boussinesq hypothesis turbulence is modelled as a diffusion like term which corresponds to the fact that turbulence is a diffusive phenomenon. In Eqn. (48) two unknowns arise, k and ν_t. Quantity k is the turbulence kinetic energy

$$k = \frac{1}{2}\overline{u_i'^2} \tag{49}$$

i.e. a sum of the normal stresses. Eddy viscosity or turbulent kinematic viscosity ν_t is a diffusion coefficient in a case of a turbulent flow and can be modelled in different ways depending on the exact turbulence model. An example is the SST k-ω turbulence model which is described in Chapt. 4.3.2.

In a Reynolds stress model (RSM) the Reynolds stresses from equation (47) are directly solved either by an algebraic stress model or by a differential equation for the rate of change of stress. With the RSM it is possible to solve problems with anisotropic turbulence, but it is also computationally more intensive since there is one equation for each of the six Reynolds stresses.

The turbulent flow contains eddies with a wide range of length and time scales. Large eddies in turbulent flow contain more energy than smaller ones and are often comparable in size to the characteristic length of the mean flow. In the smallest scale of eddies the turbulence kinetic energy dissipates to heat. In large-eddy simulation (LES) the large-scale motions are solved directly in a time-dependent simulation, where spatially filtered Navier-Stokes equations are applied instead of time-averaged Navier-Stokes. With a LES model the fluid flow is always calculated in a transient way. The not resolved small-scale motions are modelled with subgrid models, like Smagorinsky subgrid-scale model [35]. LES models are usually used only in research purposes because they are computationally expensive. Nowadays the LES modelling has though become more common tool in some applications, like in HVAC simulations.

In direct numerical simulation the Navier-Stokes equations are solved accurately without modelling turbulence with any approximative model, i.e. the only approximations are the numerical discretizations. The flow field is thus always calculated in a transient way. DNS is the most exact approach to model turbulence, but also the most time-consuming. DNS is mostly too expensive to be used for engineering purposes. DNS is, however, a good approach to study phenomena behind turbulence. [32]

4.3.2 The SST k-ω turbulence model

The SST (Shear Stress Transport) k-ω turbulence model is a linear two-equation eddy viscosity model where the k-ω turbulence model of Wilcox is merged with high Reynolds number k-ϵ model which is transformed into k-ω formulation. The k-ω

model is applied in the viscous linear sublayer and also in the logarithmic region of the boundary layer. The k-ϵ model is applied in free flow. In the SST k-ω turbulence model two supplementary equations in addition to the governing equations of the fluid flow are solved, namely equations for kinetic energy of turbulence k and for its specific dissipation ω. The equation for turbulence kinetic energy k can be derived from Navier-Stokes equations [32]

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho\bar{u}_j k)}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\mu\frac{\partial k}{\partial x_j}\right)-\frac{\partial}{\partial x_j}\left(\frac{\rho}{2}\overline{u_j'u_i'u_i'}+\overline{p'u_j'}\right)-\mu\overline{\frac{\partial u_i'\partial u_i'}{\partial x_k\partial x_k}}-\overline{\rho u_i'u_j'}\frac{\partial\bar{u}_i}{\partial x_j} \tag{50}$$

A left-hand-side of Eqn. (50) is the total derivative of k and the first term on the right-hand-side (RHS) is the molecular diffusion of k. The left-hand-side and the first term on the right-hand-side need no modelling [32]. The second term on the RHS represents turbulent diffusion of kinetic energy and is modelled in the following way [32]

$$\left(\frac{\rho}{2}\overline{u_j'u_i'u_i'}+\overline{p'u_j'}\right)=\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_j} \tag{51}$$

The third term is a turbulent kinetic energy sink term i.e. turbulent kinetic energy dissipation [32]

$$\mu\overline{\frac{\partial u_i'\partial u_i'}{\partial x_k\partial x_k}}=\beta^*\rho k\omega \tag{52}$$

The fourth term is the rate of production of turbulent kinetic energy P_k. By substituting the Boussinesq hypothesis (48) for the Reynolds stresses and the definition of the strain rate tensor S_{ij} the production term can be written in the following way [32]

$$P_k=-\overline{\rho u_i'u_j'}\frac{\partial\bar{u}_i}{\partial x_j}\approx\mu_t\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right)\frac{\partial\bar{u}_j}{\partial x_j}=2\mu_t S_{ij}^2 \tag{53}$$

The production term is limited in the following way [37]

$$P_{lim}=\min[P_k,c_1\beta^*\rho k\omega] \tag{54}$$

It can be noticed that the production term is limited to the maximum value of c_1 times the dissipation. The limiting is done to avoid too large turbulence values for example in stagnation regions [37]. The equation for k can be thus written

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho\bar{u}_j k)}{\partial x_j}=P_k-\beta^*\rho k\omega\frac{\partial}{\partial x_j}\left[(\mu+\alpha_k\mu_t)\frac{\partial k}{\partial x_j}\right] \tag{55}$$

The equation for ω is also derived from Navier-Stokes equations but is written directly in a modelled form [37, 38].

$$\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho\bar{u}_j\omega)}{\partial x_j}=\gamma\rho S_{ij}^2-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\alpha_\omega\mu_t)\frac{\partial\omega}{\partial x_j}\right]+2(1-F_1)\rho\alpha_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j} \tag{56}$$

where the blending function F_1 is defined in the following way

$$F_1 = \tanh\left[\min\left[\min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right), \frac{4\rho\alpha_{\omega 2}k}{CD_{k\omega}y^2}\right], 10\right]^4\right] \tag{57}$$

where

$$CD_{k\omega} = \max\left(2\rho\alpha_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-10}\right) \tag{58}$$

and y is the distance to the nearest wall. The blending function F_1 is zero away from the surface and thus the standard $k-\epsilon$ model is applied. Correspondingly F_1 goes to value 1 in the boundary layer and there $k-\omega$ model is applied.

Turbulent kinematic viscosity (eddy viscosity) is

$$\mu_t = \frac{a_1\rho k}{\max(a_1\omega, F_2 S)} \tag{59}$$

where F_2 is a second blending function

$$F_2 = \tanh\left[\left[\max\left(\frac{2\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right)\right]^2\right] \tag{60}$$

and S is the invariant measure of the strain rate

$$S = \sqrt{2}|\frac{1}{2}((\nabla\mathbf{U}) + (\nabla\mathbf{U})^T)| \tag{61}$$

The values of the coefficients γ, α_k, α_ω and β are filtered between the corresponding model coefficients using the blending function F_1 Eqn. (57)

$$\phi = F_1\phi_1 + (1 - F_1)\phi_2 \tag{62}$$

where $\phi = \{\gamma, \alpha_k, \alpha_\omega, \beta\}$. The model coefficients are presented in Tab. 2

Table 2: *Model coefficients for SST $k-\omega$ turbulence model.*

γ_1	α_{k1}	$\alpha_{\omega 1}$	β_1	γ_2	α_{k2}	$\alpha_{\omega 2}$	β_2	β^*	$c1$
0.5532	0.85034	0.5	0.075	0.4403	1.0	0.85616	0.0828	0.09	10.0

4.3.3 Wall functions

In the logarithmic part of the boundary layer, for approximately the range of the dimensionless wall distance $35 < y^+ < 350$ [31, Chapt. 6-4.1], the velocity profile is estimated with a so-called log law

$$u^+ = \frac{1}{\kappa}\ln(y^+) + B \tag{63}$$

where $\kappa \approx 0.41$ is a von Kármán constant and $B = 5.1$ is another constant. Close to the wall, in the linear viscous sublayer of the boundary layer, the velocity profile is linear

$$u^+ = y^+ \tag{64}$$

A boundary layer can be modelled precisely in which case the grid must be fine enough to capture the boundary layer form. Usually the adequate condition is that $y^+ \approx 1$ for the cells at the wall. If a wall function is applied to model the boundary layer, the first cell should be located in the logarithmic region.

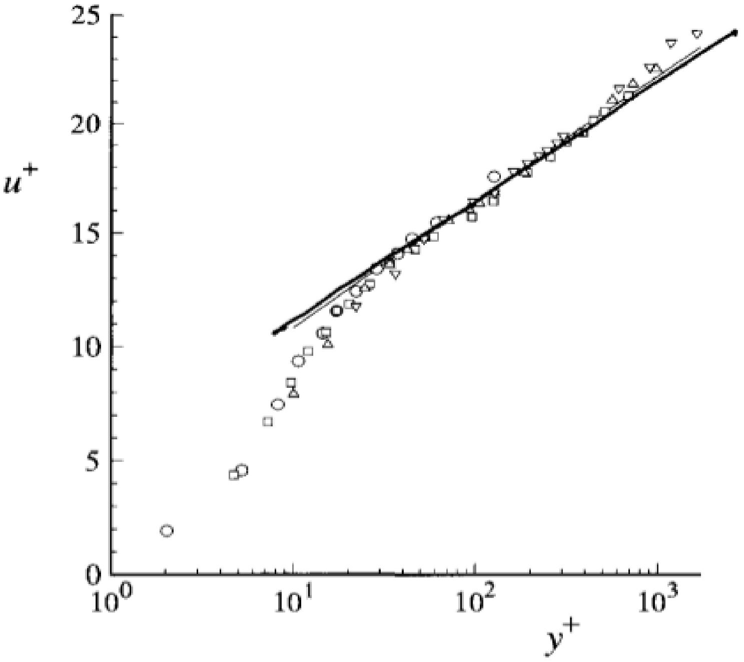


Figure 12: *Dimensionless velocity u^+ as a function of the dimensionless distance from the wall y^+ in boundary layer [36].*

The boundary layer affects the flow through the prediction of friction. If the boundary layer on the wall is not modelled well, the friction and pressure losses caused by it will also be falsely predicted.

4.4 Flow in a porous medium

Packed beds are often modelled as porous regions in CFD models. A porous medium is characterized as a solid medium containing pores i.e. voids, where some of the pores are interconnected to paths so that fluid can flow across the porous medium. The solid phase is called solid matrix. Pores i.e. void areas are filled by some fluid, gas or liquid.

The quantities porosity ϕ and permeability K are used to describe the porous medium. Porosity is defined as

$$\phi = \frac{V_{void}}{V_{void} + V_{solid}} \tag{65}$$

where V_{void} is the volume of the voids and V_{solid} is the volume of the solid phase. The intrinsic permeability tensor $\mathbf{K}$ describes the ability of the fluid to flow through the porous medium and depends on the porous medium. If the porous medium is isotropic, $\mathbf{K}$ is the same in different directions and in a case of anisotropic porous medium it is different in different directions. A hydraulic conductivity tensor $\mathbf{K_f}$ is defined from the intrinsic permeability tensor $\mathbf{K}$

$$\mathbf{K_f} = \mathbf{K}\frac{\rho g}{\mu} \tag{66}$$

where ρ is the density, μ the viscosity and g the gravitational acceleration. For the isotropic porous medium, the intrinsic permeability can be considered as a scalar K. Permeability can be defined for example from experiments. [41]

The flow in the porous medium is divided to four regimes which are Pre-Darcy, Darcy, Forchheimer and turbulent flow regimes. In the Darcy regime viscous forces dominate and the pressure gradient is directly proportional to the flow rate. In the Forchheimer regime inertial forces influence the flow and cause nonlinear relationship between pressure gradient and flow rate. According to [40, p.12], the transition from Dracy flow to Darcy-Forchheimer flow occurs approximately when Re_K is of order 10^2, where the subscript K refers to the length scale. In Fig. 13 A and B are first and second Ergun constants and subscripts F and T refer to Forchheimer and turbulent flow respectively. [39]

The considered scale in investigation of porous medium affects the method of solving flow fields in porous medium. If the porous medium is observed in the small-scale, or pore-scale, only few pores filled by fluid are considered. Conventional fluid dynamics approach can be thus applied in pore-scale. In the large-scale the porous medium contains a large number of pores and the flow field is rather complex and difficult to solve with the conventional fluid dynamics approach. Therefore, in large-scale investigations volume averaging approach i.e continuum model is applied. Continuum model for porous medium is presented below. [40]

Mass balance for porous medium is derived from the conventional fluid dynamics Eqn. (15)

$$\phi\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\phi\mathbf{U}) = 0 \tag{67}$$

According to Ref. [41] the experimentally obtained Darcy and Forchheimer laws are used to describe the macro-scale momentum balance in porous medium. Darcy law describes slow or creeping flows where viscous effects dominate. Forchheimer law describes high velocity flows where the inertial forces start to affect the flow.

According to the Henry Darcy's experiments in 1856 proportionality between the flow rate and the pressure loss in a uniform porous medium with a steady uniform flow is obtained. In an isotropic porous medium the Darcy's equation is

$$\nabla p = -\frac{\mu}{K}\mathbf{U} \tag{68}$$

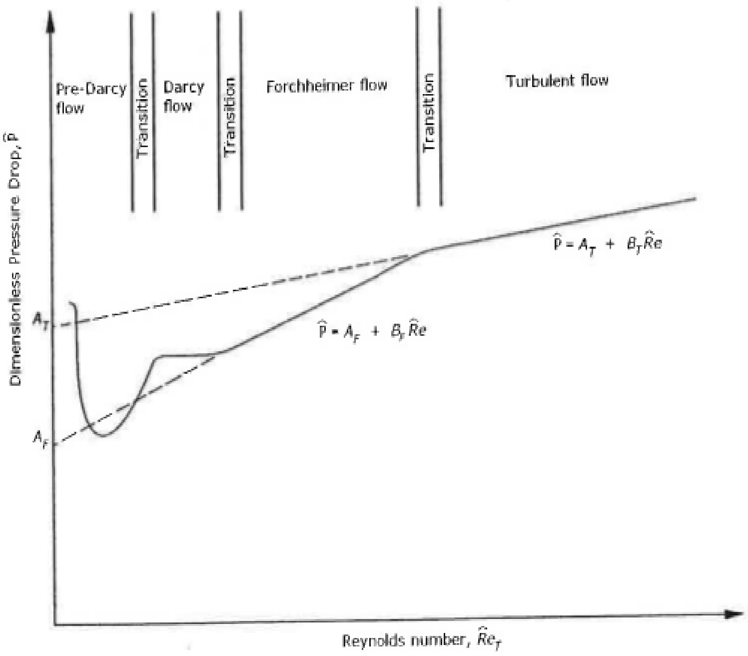


Figure 13: *Porous flow regimes [39].*

where K is intrinsic permeability. Equation (68) is valid only for slow velocities in a porous medium ($Re \leq 1$), since inertial and time dependent effects are neglected in a derivation.

Forchheimer studied the flow in porous medium in high velocities and noticed that the inertial effects start to dominate in the flow as the velocity increases. Thus a term representing the kinetic energy was added to Eqn. (68) and the following equation called Forchheimer equation was obtained

$$\nabla p = \frac{\mu}{K}\mathbf{U} - F\rho|\mathbf{U}|\mathbf{U} \tag{69}$$

where F is the Forchheimer coefficient. In practice, F is usually obtained from the best fit to the experimental data.

4.4.1 Porous medium in OpenFOAM

The porous medium in OpenFOAM is taken into account with the source term in momentum equations (16). The source term is Eqn. (69), but in notations used in OpenFOAM the source term is written in the following way

$$S_i = -\left(\mu d_i + \frac{1}{2}\rho|u_i f_i|\right) u_i \tag{70}$$

where the first term is viscous loss term and the second the inertial loss term.

In OpenFOAM, the local coordinates can be defined for the porous medium. The local coordinates are noted $(e1, e2, e3)$. In OpenFOAM, the local coordinates $(e1, e2, e3)$ and the coefficients of equation (70) d and f are defined in the `constant/porousZones` file in the following way

```
...
    porosity1
    {
        coordinateSystem
        {
            e1  (0.70710678 0.70710678 0);
            e2  (-0.70710678 0.70710678 0);
        }

        Darcy
        {
            d   d [0 -2 0 0 0 0 0] (0 0 0);
            f   f [0 -1 0 0 0 0 0] (-1000 3.5 -1000);
        }
     }
...
```

Coordinate $e3$ is defined by the right hand rule from directions $(e1, e2)$. The values of coefficients d and f are defined in the directions of $(e1, e2, e3)$. Thus the implementation of the Forchheimer equation in OpenFOAM enables the definition of an anisotropic porous media. In the example file, the angles of the coordinates $e1$ and $e2$ are 45° and $-45°$ with respect to global x-coordinate, respectively. The viscous term is neglected by setting the coefficient $d = 0$ in all directions. In a high velocity region inertial term dominates, thus f defines the resistance on the flow induced by the porous medium. The coefficient in direction $e2$ is $f_{e2} = 3.5$ and in directions $e1$ and $e3$ it is set $f_{e1} = f_{e3} = -1000$ which means that the coefficients in those directions are 1000 times the coefficient that has a positive value i.e. f_{e2}. Thus this file defines an anisotropic porous media, which has a lower resistance in direction $e2$ (45° to the x-axis) and practically not permeable walls in the other directions. In this work, this is applied to define a structured packing, which has a specific corrugation angle and permeability.

4.4.2 Turbulence modelling in porous medium

Flow in the porous medium is never purely laminar, since there are always some vortices. The eddy size in porous a medium is limited by the pore size and this distinguishes the turbulence in the porous medium from the turbulence in flow without solid obstruction.

Turbulence in the porous medium is a difficult issue, since at a small permeability and a low velocity the flow is often laminar, but in a highly permeable medium and with a high velocity the flow can be turbulent [45]. Difficulties in turbulence modelling in porous media involve taking into account the solid matrix and the interaction of the fluid flow with the solid phase [44].

The most suitable approach to model turbulence in the porous medium is debated within the research community. Turbulence modelling in the porous medium has thus become an independent research topic. The most common approach to model turbulence in the porous medium is to modify the $k-\epsilon$ model [43]. Turbulence models for the porous medium are studied in Refs. [42] -[45]. In Ref. [46] alternative methods for turbulence modelling in porous medium are discussed.

In the article of Antohe et al. [44] the derivation of the macroscopic turbulence $k-\epsilon$ model for incompressible flow in the porous medium is presented. The derivation of the turbulence model is made by time-averaging the general equations with the time acceleration term, convective inertia term, pressure gradient and Darcy, Forchheimer and Brinkman terms. The purpose of this endeavour is to provide a correct turbulence $k{-}\epsilon$ model formulation which is consistent with general equations, a mathematical model for simulating macroscopic turbulence in complex geometries and the means to verify the ability of a macroscopic general model to represent the microscopic level turbulence. [44]

In OpenFOAM the modelling of the porous media is done with the Darcy - Forchheimer equation. The porous media is taken into account as an additional resistance to flow and there is no real solid obstacle needed to model a solid matrix of the porous medium. Thus there is no solid medium that could have an effect on turbulence either. The modelling of the porous medium by an additional resistance is often reasonable for the application, since then the geometry of a complex porous structure need not to be modelled and a lot of time is spared. In a viewpoint of turbulence modelling, this approximation may lead to difficulties, since the effect of the real solid structure is not taken into account. The assumption is reasonable if the permeability of the medium is quite large and the geometric scale of the medium does not interact with the scale of the turbulent eddies.

5 OpenFOAM

OpenFOAM is an open source CFD software package developed by OpenCFD Ltd at ESI Group and it is distributed by OpenFOAM Foundation. It is written in object oriented programming language C++. OpenFOAM is primarily designed to solve problems in continuum mechanics [47].

In this work, OpenFOAM is used for modelling the gas phase flow field in structured packed bed column. Gas is assumed incompressible. In this chapter, the incompressible solver with porous media of OpenFOAM is explained. The porous media model is in Chapt. 7 used to simulate the packed bed.

5.1 Solver for incompressible flow in porous medium

In Chapt. 7 the computations concerning a gas flow in a structured packed bed are conducted with OpenFOAM. In the computations the density is constant. The solver `porousSimpleFoam` in OpenFOAM 2.1.1 is a steady-state solver for incompressible, turbulent flow with implicit or explicit porosity treatment. Both options are presented in this chapter, but only the implicit porosity treatment is applied in this work because it is more robust as compared to the explicit porosity treatment.

The solver `porousSimpleFoam` is basically a solver applying the well-known SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm [32], where additionally also the porous media is taken into account. The `porousSimpleFoam` solver is defined in `porousSimpleFoam.C` file in the OpenFOAM 2.1.1 code, which is available after installing OpenFOAM [48]. The momentum equation is defined in the file `/solvers/incompressible/ simpleFoam/porousSimpleFoam/UEqn.C` in the following way

```
...
    tmp<fvVectorMatrix> UEqn
    (
        fvm::div(phi, U)
      + turbulence->divDevReff(U)
      ==
        sources(U)
    );
...
```

Mathematically, the momentum equation in the code can be written in the following way.

$$\nabla \cdot (\mathbf{UU}) - 2\nabla \cdot (\nu_{eff} S) = \mathbf{Q} \tag{71}$$

where $S = \frac{1}{2}((\nabla \mathbf{U}) + (\nabla \mathbf{U})^T)$ is the strain rate tensor. On the left hand side the first term is the convection term and the second term is the diffusion term which contains the effect of molecular viscosity ν and turbulent viscosity ν_T, which arises from the Boussinesq approximation to model the Reynolds stress term. On the right hand side of the momentum equation Eqn. (71) the sources are defined as $\mathbf{Q}$. The sources consist of body forces, for example gravity. The equation can also be written

$$\nabla \cdot (\mathbf{U}\mathbf{U}) - \nabla \cdot \nu_{eff}(\nabla \mathbf{U})_f - \nabla \cdot (\nu_{eff}((\nabla \mathbf{U})^T - \frac{1}{3}(\mathrm{tr}(\nabla \mathbf{U})^T)I) = \mathbf{Q} \tag{72}$$

The quantity ν_{eff} is the effective kinematic viscosity

$$\nu_{eff}(x_i, t) = \nu + \nu_T(x_i, t) \tag{73}$$

In the code the momentum equation Eqn. 72 is actually in its discretized form

$$\sum_f \mathbf{S}_f \cdot \mathbf{U}_f \mathbf{U}_f - \sum_f \nu_{eff} \mathbf{S}_f (\nabla \mathbf{U})_f - \sum_f \nu_{eff} \mathbf{S}_f ((\nabla \mathbf{U})^T - \frac{1}{3}\mathrm{tr}(\nabla \mathbf{U})^T I)_f = \mathbf{Q} \tag{74}$$

where the calculation methods for the gradients are defined in file `fvSchemes`. The gradient at the face in the second term in Eqn. (74) is calculated either from the both cell centre values next to face (uncorrected method) or additionally a correction based on gradients calculated in cell centres around the face is applied (corrected method). in the latter case the velocity gradient is calculated according to Eqn. (75). The gradient of the velocity in the third term in Eqn. (74) is also discretized according to Eqn. (75)

$$\nabla \mathbf{U} = \frac{1}{V_p} \sum_f \mathbf{S}_f \mathbf{U}_f \tag{75}$$

The discrete momentum equation (74) is linearised as explained in Chapt. 4.2.1 and can be written

$$a_P \mathbf{U}_P + \sum_n a_n \mathbf{U}_n = \mathbf{RHS} \tag{76}$$

where index P refers to the computational cell for which the equation is written and index n refers to the neighbouring cells. The coefficients a_P and a_n are elements of the coefficient matrix of the linearised equation group for momentum equation. It should be noted that the discretized momentum equation Eqn. (74) is divided by the cell volume V_P to obtain the form in Eqn. (76). The term $\mathbf{RHS}$ (right hand side) comes from the linearisation of Eqn. (74), i.e. it refers to the residual part or source part of the linearised equation. At this point the values of a_P and $\mathbf{RHS}$ depend on the porosity treatment. The differencies are explained in Chapts. 5.1.1 and 5.1.2. By splitting the $\mathbf{RHS}$ into velocity- and pressure dependent parts

$$\mathbf{RHS} = \mathbf{rhs}(\mathbf{U}) - \nabla p \tag{77}$$

and by adopting the notation used in OpenFOAM

$$\mathbf{H}(\mathbf{U}) = -\sum_n a_n \mathbf{U}_n + \mathbf{rhs}(\mathbf{U}) \tag{78}$$

Eqn. (76) can be written

$$a_P \mathbf{U}_P = \mathbf{H}(\mathbf{U}) - \nabla p \tag{79}$$

An explicit expression for a velocity $\mathbf{U}^*$ is written from Eqn. (79) with the known values from the previous time step and velocity is solved according to Eqn. (80).

$$\mathbf{U}^* = \frac{\mathbf{H}(\mathbf{U}, \phi) - \nabla p^n}{a_P} \tag{80}$$

This is how the momentum predictor is implemented in the implicit porosity treatment.

In the explicit porosity treatment, the momentum is predicted by solving the implicit equation group defined in Eqn. (76) and a new velocity $\mathbf{U}^*$ is obtained. To increase the diagonal dominance of the coefficient matrix, a relaxation factor $0 < \alpha_u < 1$ is applied to Eqn. (76) and thus Eqn. (81) is formed

$$\frac{a_P}{\alpha_u}\mathbf{U}_P^i + \sum_n a_n \mathbf{U}_n^i = \mathbf{RHS} + \frac{(1-\alpha_u)}{\alpha_u} a_P \mathbf{U}_P^{i-1} \tag{81}$$

In OpenFOAM, additionally a form of Eqn. (80) without pressure gradient

$$\mathbf{U}^{**} = \frac{\mathbf{H}(\mathbf{U}^*)}{a_P} \tag{82}$$

and a volume flux based on it

$$\phi^* = \left(\frac{\mathbf{H}(\mathbf{U}^*, \phi^n)}{a_P}\right)_f \cdot S_f \tag{83}$$

are written for a later use. These new quantities are written using the newly obtained velocity $\mathbf{U}^*$. If the velocity $\mathbf{U}^*$ is the final velocity, the continuity equation must be satisfied , i.e.

$$\nabla \cdot \mathbf{U} = \nabla \cdot \left(\frac{\mathbf{H}(\mathbf{U}^*, \phi^n)}{a_P}\right) - \nabla \cdot \left(\frac{\nabla p^n}{a_P}\right) = 0 \tag{84}$$

which leads to the pressure equation

$$\frac{1}{a_P}\nabla^2 p^{*,n+1} = \nabla \cdot \left(\frac{\mathbf{H}(\mathbf{U}^*, \phi^*)}{a_P}\right) \tag{85}$$

from which the pressure $p^{*,n+1}$ is solved. The information of the calculated pressure is added to the flux ϕ^*

$$\phi^{*,n+1} = \phi^* - (\nabla p)_f \cdot S_f \tag{86}$$

After under-relaxation of the pressure with a relaxation factor $0 < \alpha_p < 1$ the new value for the pressure is obtained

$$p^{n+1} = p^n + \alpha_p(p^{*,n+1} - p^n) \tag{87}$$

A final value for the velocity $\mathbf{U}$ is obtained in the following way

$$\mathbf{U} = \mathbf{U}^{**} - \frac{\nabla p^{n+1}}{a_P} = \frac{\mathbf{H}(\mathbf{U}^*, \phi)}{a_P} - \frac{\nabla p^{n+1}}{a_P} \tag{88}$$

The difference between implicit and explicit porosity treatment is in the way how the supplementary resistance due to porous media is taken into account in the momentum equation. In the implicit porosity treatment the coefficients from Darcy-Forchheimer equation are added to the diagonal terms of the coefficient matrix A of the momentum equation. In the explicit porosity treatment the coefficients from the Darcy-Forchheimer equation are added to the right-hand-side of the momentum equation. In the explicit porosity treatment the implicit momentum equation is solved before the pressure equation. In the implicit porosity treatment the momentum equation is not solved for in its implicit form.

The exact description of how the porosity influences the terms a_P and $\mathbf{RHS}$ in Eqn. (76) is given in the following subsections Chapts. 5.1.2 and 5.1.1.

5.1.1 Explicit porosity treatment

The porosity influence is implemented in function `addResistance` in `porousZonesTemplates.C`. In case of the explicit porosity treatment it reads:

```
...
template<class RhoFieldType>
void Foam::porousZone::addViscousInertialResistance
(
    scalarField& Udiag,
    vectorField& Usource,
    const scalarField& V,
    const RhoFieldType& rho,
    const scalarField& mu,
    const vectorField& U
) const
{
    const tensor& D = D_.value();
    const tensor& F = F_.value();

    forAll(cellZoneIds_, zoneI)
    {
        const labelList& cells = mesh_.cellZones()
        [cellZoneIds_[zoneI]];

        forAll(cells, i)
        {
            const tensor dragCoeff = mu[cells[i]]*D
                + (rho[cells[i]]*mag(U[cells[i]]))*F;

            const scalar isoDragCoeff = tr(dragCoeff);

            Udiag[cells[i]] += V[cells[i]]*isoDragCoeff;
            Usource[cells[i]] -=
```

```
                V[cells[i]]*((dragCoeff - I*isoDragCoeff) &
                U[cells[i]]);
        }
    }
}
...
```

The `dragCoeff` in the code refers to the drag coefficient according to the Darcy-Forchheimer equation

$$C_{Dij} = \mu D_{ij} + \rho |u_{kk}| F_{ij} = \mu d_{ij} + \frac{1}{2}\rho |u_{kk}| f_{ij} \tag{89}$$

The `isoDragCoeff` is the sum of the diagonal terms (trace) of the drag coefficient matrix, Eqn. (89), and is written according to Eqn. (90).

$$\mathrm{tr}(C_{Dij}) = C_{Dkk} = \mu d_{kk} + \frac{1}{2}\rho |u_{kk}| f_{kk} \tag{90}$$

The resistance is added on the right side of the Eqn. (76) by multiplying the drag coefficient by the cell volume, so in the explicit porosity treatment, the term **RHS** contains the following expression $C_{Dij}V_P \cdot U_i$. On the left-hand-side the isotropic part of the drag coefficient is added into the diagonal term of the coefficient matrix. Accordingly the same isotropic part must be added into the right-hand-side of the equation and multiplied by the corresponding velocity. Thus the diagonal term becomes

$$a_P = a_P + V_P C_{kk} \tag{91}$$

and the right-hand-side becomes

$$RHS_i = RHS_i + V_P C_{kk} U_i - C_{Dij} V_P \cdot U_i \tag{92}$$

5.1.2 Implicit porosity treatment

In case of the implicit porosity treatment function `addResistance` in `porousZonesTemplates.C`, in which the porosity influence is implemented, reads:

```
...
template<class RhoFieldType>
void Foam::porousZone::addViscousInertialResistance
(
    tensorField& AU,
    const RhoFieldType& rho,
    const scalarField& mu,
    const vectorField& U
) const
{
    const tensor& D = D_.value();
    const tensor& F = F_.value();
```

```
    forAll(cellZoneIds_, zoneI)
    {
        const labelList& cells = mesh_.cellZones()
        [cellZoneIds_[zoneI]];

        forAll(cells, i)
        {
            AU[cells[i]] += mu[cells[i]]*D +
            (rho[cells[i]]*mag(U[cells[i]]))*F;
        }
    }
}
...
```

The resistance due to the porosity is added into to the diagonal terms $a_{P,i}$ in the coefficient matrix of the momentum equation. Thus the diagonal term becomes

$$a_{P,i} = a_{P,i} + \mu d_{ij} + \frac{1}{2}\rho|u_{kk}|f_{ij} \tag{93}$$

where d_{ij} and f_{ij} are the coefficients in Darcy-Forchheimer equation.

6 Previous studies

Structured packings are popular in various heat and mass transfer applications. Many researches have tried to model structured packing behaviour by CFD. The study of Mahr and Mewes [22] and the study of Raynal and Royon-Lebeaud [23] deal with the two-phase flow in structured packed beds. Also some experiments made on structured packed bed columns are presented in this chapter. From the study of Owens [24] experimental data on the pressure drop across the packed bed was obtained. The study of Olujić provided data concerning the maldistribution of a gas flow in structured packed beds.

6.1 Study of Mahr & Mewes 2007

In the study of Mahr & Mewes (2007) [22] the three-dimensional macroscopic gas-liquid (water and air) two-phase flow field is modelled in an entire column in order to model large scale maldistribution of the phases by applying elementary cell model. The elementary cell model was developed by Arbogast et al. 1990 and adapted by Mewes et al. 1999. The counter-current operating conditions are taken into account and both steady-state and dynamic cases are calculated below the loading point. The entire domain is divided into elementary cells and one elementary cell is a representative volume comprising several pores. In corrugated structured packing the elementary cell is the smallest structure recurring in all three dimensions. Each elementary cell consists of two sheets of corrugated packing material due to the periodicity. Conservation equations for mass and momentum are formed in the scale of one elementary cell and the entire media. Therefore the exact pore structure does not need to be modelled. The modelled packing is Sulzer Mellapak 250Y.

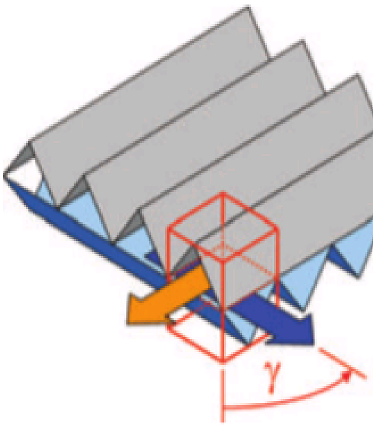


Figure 14: *Schematic drawing of two sheets of packing. The boundaries of the elementary cell are presented as an outline. The preferential flow direction and angle to the vertical column axis are indicated.*

Liquid phase is mainly governed by gravity and the resulting forces between packing and liquid. The physically homogeneous liquid phase is modelled as two liquid phases, each of them for one preferential flow direction. Conservation equations are formulated for each liquid phase and laminar film flow assumption is made. Gas phase is considered to be driven mainly by a pressure gradient. Mass transfer in between gas and liquid phase is neglected, but mass transfer between the two liquid phases is taken into account through a volume flux.

In order to measure a directional pressure drop, the Mellapak 250.Y PP structured packing was cut in various angles and measurements were conducted in a wind tunnel of rectangular cross section. Pressure drop measurements are done in different directions for the gas phase in order to form the anisotropic gas flow resistance tensor to take into account the anisotropic structure of the packing.

In the final model, there are three sets of conservation equations for each phase, one for each liquid phase and one for gas phase. The model was implemented in the CFX 10.0 code. Three continuous Eulerian phases were applied to model gas phase and two liquid phases.

Numerical results are tested against X-ray radiographic measurements on a quasi two-dimensional segment of structured packing with a liquid flow. The spreading pattern obtained with CFD modelling agrees well with the experimental results. The results show that this model can well predict dynamic flow fields and maldistribution of the phases which is seen in the Figure 15. [22]

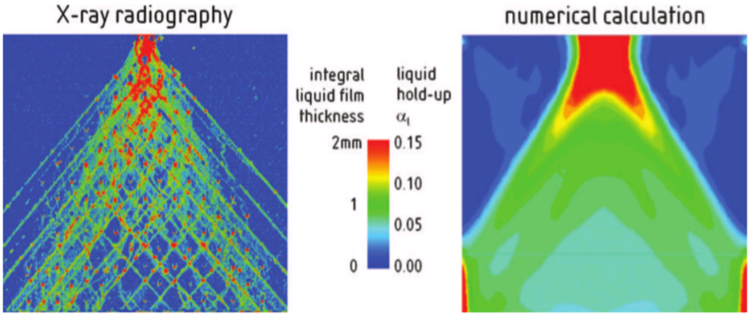


Figure 15: *Left: X-ray radiographic film thickness measurement in structured packing. Right: Calculated liquid hold-up for a two-dimensional numerical grid.*

6.2 Study of Raynal and Royon-Lebeaud 2007

Raynal and Royon-Lebeaud [23] modelled two-phase flow within structured packing using multi-scale approach. The studied geometry was also Sulzer Mellapak 250.Y structured packing and the simulations were performed with Fluent 6.2 solver.

First, in a small-scale, two-dimensional two-phase calculations include liquid-wall and liquid-gas interaction are predicted by using VOF (volume of fluid) method. The calculations are made in the scale of corrugation so that wall texture and its influence on the liquid flow are taken into account. Liquid holdup, liquid velocity at gas-liquid interface and a thickness of the liquid film are calculated.

Second, the previous results from small-scale are used in three-dimensional meso-scale gas flow calculations. The meso-scale geometry in calculations corresponds to a periodic element representative of the real packing geometry, which is the volume between the opposite smooth metal sheets. The relationship between a pressure drop and a gas superficial velocity can be calculated. The presence of liquid is taken into account indirectly in the following way: The superficial gas velocity is converted into interstitial gas velocity by taking the liquid hold-up into account and the liquid velocity at gas-liquid interface is considered as a moving wall boundary condition for the gas flow.

The meso-scale results are used in three-dimensional calculations at the large scale corresponding to the entire column. In the simulations at the column scale, the packed bed is modelled as anisotropic porous media where pressure drop coefficients are obtained from meso-scale calculations. The two pressure loss coefficients are the same $K_z = K_{(x\ or\ y)}$ and the coefficient in the third direction $K_{(y\ or\ x)}$ is considered to be infinite to model the influence of the perpendicular plates. In practice it is 1000 times larger than the two others.

Results are compared against experimental data and partly against theory. According to comparison this method gives satisfactory results below loading point and for low and intermediate liquid flows. This is the condition for CO_2 absorbers, so this method could be applied in calculations of structured packing columns loaded by gas-liquid flow.

6.3 Study of Owens

Owens [24] made experiments with Mellapak 250Y structured packing (packing with specific area of 250 m^2/m^3). The experiments measured the pressure loss of a gas phase flow against gravity through Mellapak 250Y. In experiments, the system was brought to steady flow and pressure losses were then measured.

The column in experiments made by Owens was assembled from a cylindrical aluminium base and two sections of 146.05 mm inner diameter polycarbonate tube (see Fig. 16). The tube has flanges at each end to connect the tubes and a base to each other. The base had a 33.4 mm threaded process connection drilled perpendicular to its axis. The base consists of a conical throat drilled along the central axis of the base and the connection intersects with the throat. The throat diameter at an intersection of the throat and the process connection is 25.4 mm and at the top of the throat its diameter is 76.2 mm. This kind of channel inlet was made to distribute the fluid uniform into the column.

The lower tube section was 764 mm tall and the upper 966 mm tall, that is 1730 mm in total. The lower tube section acted only as an inlet, such that the flow profile developed before entering to the packing. In the upper tube section the pressure was measured at three elevations of 127 mm, 699.3 mm, and 829.5 mm above the flanged tube connection (Fig. 16). At each elevation, four ports were spaced evenly around the column perimeter in a plane perpendicular to the column axis. A picture of the column, packed with two whole-elements of Mellapak N250Y is shown at right in Fig. 16. All three pressure measurement elevations are clearly visible in this figure.

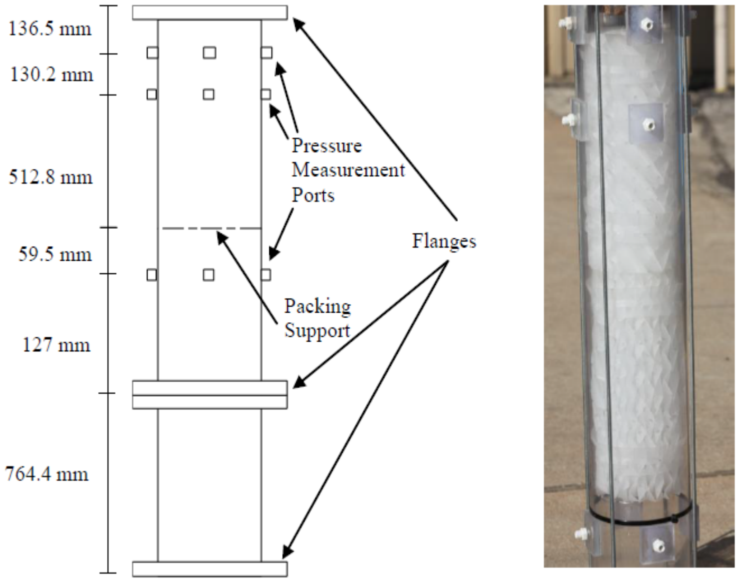


Figure 16: *Laboratory column with two whole elements of Mellapak 250Y.*

The packing support is placed 186.5 mm above the flanged tube connection in the upper tube section. The packed bed is 610 mm long in total. The grid support in the nitrogen experiments was a 1.59 mm wire, which should not cause considerable disturbances into the flow field in the bottom part of the packing. Owens made also experiments with water, but the packing support in water experiments was an aluminium grate.

Gas phase experiments were conducted with a single pass flow of building supplied nitrogen. Before entering into the column, the flow was distributed for two mass flow meters and after them fed to the base of the column. Nitrogen flowed through the column and was then vented to the atmosphere. The nitrogen temperature was monitored and found not to vary outside the range of 22.3 – 23.9 °C. The nitrogen flow was monitored and brought to desired value. The system was stable at least three minutes before data was recorded. One recording lasted for three minutes. Pressure drops were measured in at nitrogen flows of $F_S = (0.610 - 3.36)\ \mathrm{Pa}^{1/2}$.

Lab-scale columns are typically less than 500 mm in diameter. Packing designed for these columns typically have a gap of $6-8$ mm between the edge of the packing and the column wall. In industrial columns which can exceed 3000 mm in diameter with a typical wall gap of less than 12 mm, so the annular-to-packed area ratio is much smaller in industrial columns compared to laboratory scale columns. Therefore the scaling up is difficult with laboratory scale models.

The dissertation of Owens [24] included reasonable well documented experimental data, which was considered useful in this thesis work. CFD calculations made in this work are, therefore, compared to the experiments to validate the model in laboratory scale. The validated model is used for further calculations in larger, industrial scale columns.

6.4 Study of Olujic 2003

Olujic et al. [27] made large-scale experiments conserning of the relation between quality of the initial gas distribution and hydraulics of a structured packed bed. Also smoothing of the initial gas maldistribution during the bed height was studied, which is interesting in the viewpoint of this thesis. The initial maldistribution was obtained with the aid of a plate at the packed bed inlet. In the study of Olujic, four different cases were studied: uniform distribution, initial maldistribution with chordal blockage, with central blockage and with multi-hole inlet. The chordal blockage and central blockage cases are studied also in this thesis in Chapt. 7.4.

The experiments were made in a column which diameter was $d = 1.4$ m with air-water system under atmospheric pressure and ambient temperature. The air was blown into the column with a powerful blower ($7-8$ m^3/s). The packing used in the column was Montz-pak B1-250. The packing is made of unperforated metal sheet with corrugation angle of 45°. Packing element height is 0.2 m and maximal five layers of packing is installed, thus total bed heigt is 1 m. Each packing layer is rotated 90° compared to the previous layer. The experiments are carried out in at atmospheric pressure (10^5 Pa) and ambient temperature (20 °C).

The air velocity in each cross-section under dry conditions was measured a Pitot-tube in 2450 points per cross-section. Measured local velocities are presented in a 2D plot and also maldistribution factors are calculated for different layers of packing.

7 CFD analysis

The aim of this research is to validate a practical CFD model for engineering purposes for a structured packed bed according to the experimental results from literature and then apply the model to the large-scale CO_2 absorption column to investigate a gas maldistribution. In this thesis it is hypothesised, that the gas maldistribution is indicative of the liquid maldistribution, too. Therefore, the analysis of the gas phase distribution might also provide an indication of the liquid maldistribution.

In this work, the structured packed bed is modelled as an anisotropic porous media. One alternative could be to model the exact geometry of the packing, but this option is not feasible because of the large size of the column and the small length-scales of the packing channels. The approach to study only the gas flow is justified with the fact that the multiphase CFD calculations are computationally expensive and also more complicated and unstable. Large-scale two-phase CFD simulations would require more time than available in the current work. The approach to study packed bed columns in large scale only with gas flow was also applied for example in the study of Raynal presented in Chapt. 6.2 from Ref. [23].

The model parameters for anisotropic porous media are adjusted against experimental results of Owens [24] in Chapt. 7.3. Olujic [27] made experiments on the gas maldistribution and these results are also used for validation of the model in Chapt. 7.4.

A fluid dynamics solver `porousSimpleFoam` of OpenFOAM is used to conduct the calculations. The solver is described in detail in Chapt. 5.1. The fluid flow is assumed incompressible, since the pressure changes are not large and the flow velocities are low compared to the speed of sound. Also the temperature is assumed to be constant in both validation cases described in Chapts. 7.3 and 7.4.

The figures in this chapter are made with a self-made plotting tool written in Python programming language. A plotting library *matplotlib* and *NumPy* extension for Python are applied in coding. The two or three dimensional figures are made with a ParaView open-source post-processing tool [51]. When using OpenFOAM, ParaView can be launched from the case home directory with a script `paraFoam` using the reader module supplied with OpenFOAM.

7.1 Computers

The calculations presented in Chapt. 7.3 were conducted using OpenFOAM on a computer containing four quad-core AMD Opteron 8378 processors of 2 GHz with 32 GB RAM and 1 TB of HDD space.

The calculations presented in Chapts. 7.4, 7.5 and 7.6 were conducted on two identical computers each containing four 12-core AMD Opteron 6172 processors of 2.3 GHz with 128 GB RAM and 12 TB of HDD space. These two computers are connected with the Infiniband (40 Gbps).

7.2 Schemes and solvers

In OpenFOAM, the numerical schemes for the terms of the governing equations, such as derivatives, are defined in a file called `fvSchemes`. In this work, the numerical schemes applied are principally similar for all calculated cases. For all discretizations the Gauss theorem Eqn. (27) is applied. Numerical schemes for different terms are shown in Tab. 3 and more detailed explanations of the schemes are found in Chapt. 4.2.1.

Table 3: Numerical schemes

Gradient schemes		
$\nabla \mathbf{U}$	`linear`	Linear interpolation (central differencing)
∇p		
∇k		
$\nabla \omega$		
Divergence schemes		
$\nabla \cdot (k\mathbf{U})$	`linear`	Linear interpolation (central differencing)
$\nabla \cdot (\omega\mathbf{U})$		
$\nabla \cdot (\nu_{eff} \text{dev}((\nabla \mathbf{U})^T))$		
$\nabla \cdot (\mathbf{UU})$	`linearUpwindV grad(U)`	Linear upwind differencing
Laplacian schemes		
$\nabla \cdot (\gamma \nabla \phi)$	linear corrected	Linear interpolation for γ, non-orthogonality correction for surface normal gradients
Interpolation schmemes		
$\mathbf{U}$	`linear`	Linear interpolation (central differencing)
∇p		

In Tab. 3, the Laplacian term is presented with a generalised quantities γ and ϕ. The equations for momentum, pressure, k and ω include Laplacian terms. The expression `corrected` means non-orthogonality correction for surface normal gradients i.e. that surface normal gradient is calculated using values and gradients from adjacent cell centres.

The upwind interpolation can be used in initializing the calculation after which the more accurate method like linear interpolations is chosen. The upwind method is noted in OpenFOAM simply with `upwind`. The file `fvSchemes` is found in Appendix A.

The equation solvers, tolerances and algorithms are defined in a file `fvSolution`. In all calculations pressure is solved with the generalised geometric algebraic multigrid method (`GAMG`) explained in Chapt. 4.2.2. As a smoother, Gauss-Seidel method is applied. In a case of a turbulent flow the solver option `smoothSolver` and Gauss-Seidel as the smoother is applied in OpenFOAM for solving k and ω. The file `fvSolution` is found in Appendix B. The boundary conditions in OpenFOAM are defined in `0`-directory, which contains files for each quantity that is solved for.

The flow near the walls is modelled with wall functions, which are defined for each turbulence quantity in corresponding files in OpenFOAM `/case/time/` directory.

For turbulence kinetic energy k the `kqRWallFunction` and for turbulence kinetic energy specific dissipation ω the `omegaWallFunction` is applied.

7.3 Adjusting the porous model for Mellapak 250Y: Case Owens

Packed beds are often modelled as an isotropic porous media in CFD. However, in the packing material selected for this work, Mellapak 250Y, the fluid flows via channels and it is suspected, that the isotropic porous model cannot properly predict flow field in Mellapak 250Y. It is important to use CFD to study gas maldistribution both outside and inside the packed bed. An improved porous model set-up is therefore developed in the current work.

The purpose of simulating the column in the study of Owens presented in Chapt. 6.3 is to define values for coefficients for the porous media model in equation (70) against the experimental pressure loss data by Owens. The packed bed in the column of Owens consisted of structured packing Mellapak 250Y and it was modelled as an anisotropic porous media, see Fig 21.

7.3.1 Grid generation

The geometry and computational grid for the column are generated with Gambit 2.4.9 [52]. Gambit is a preprocessor of the commercial Ansys Fluent fluid dynamics solver for generating geometry and mesh. The grid was generated with Gambit as the graphic interface allows to generate more complicated geometries compared to `blockMesh` grid generation method in OpenFOAM. This method is used in later calculations.

Across the column diameter approximately 30 computational cells were set in the coarsest grid, which should be an appropriate amount for a laminar case [28]. The coarse grid has 231732 cells. In Fig. 21 the schematic figure and an overview of the column of the case is presented. A closer look at the wiperbands with a grid structure is shown in Fig. 17. The structured packed bed is modelled using zones of porous media (Fig. 22).

The porous zones in this case are defined in the grid easily by splitting the grid volume in Gambit correspondingly to generate the porous zone volumes. The zone width corresponds to the flow channel width in the packing, which is 13.4 mm. The channel width is calculated from the whole packing width and the amount of flow channel layers in the experimental set-up of Owens [24]. After splitting the volumes, the porous zones are added to one of the four porous zone groups, accordingly to their flow direction. The characteristics i.e. local coordinates and coefficients for Eqn. (70) for each porous zone is defined in file `constant/porousZones` by the user. The porous zones are made by hand due to the small-scale of the apparatus.

First, calculations were made using two grids: with and without wiper bands around the packed bed. It was noticed that in this kind of small scale laboratory column they significantly affect the pressure loss. Therefore, the calculations were conducted with a grid with wiper bands to model the column as realistically as possible.

Wiperbands are described by cuts at locations of the wiperbands in the grid. The modelled wiperband does not allow a flow through itself, which corresponds to the actual effect the wiperbands have on the flow. The wiperbands seal the gap between packing and the column inner wall, as the packing is pushed into column which can be seen from the Fig. 20.

The packed bed of this column consists of two packing layers, where the upper one is rotated horizontally 90° with respect to the lower layer. Each packing layer is 305 mm of its height. This can be seen in Fig. 22. At the packing joints, non-conformed meshes were used i.e. the grid lines were not continuous after rotation. This part utilised the Arbitrary Mesh Interface (AMI) algorithm of OpenFOAM to create 90° change in the grid [55]. AMI is a technique that allows a flow through disconnected mesh domains. The domains can be stationary or move relative to one another.

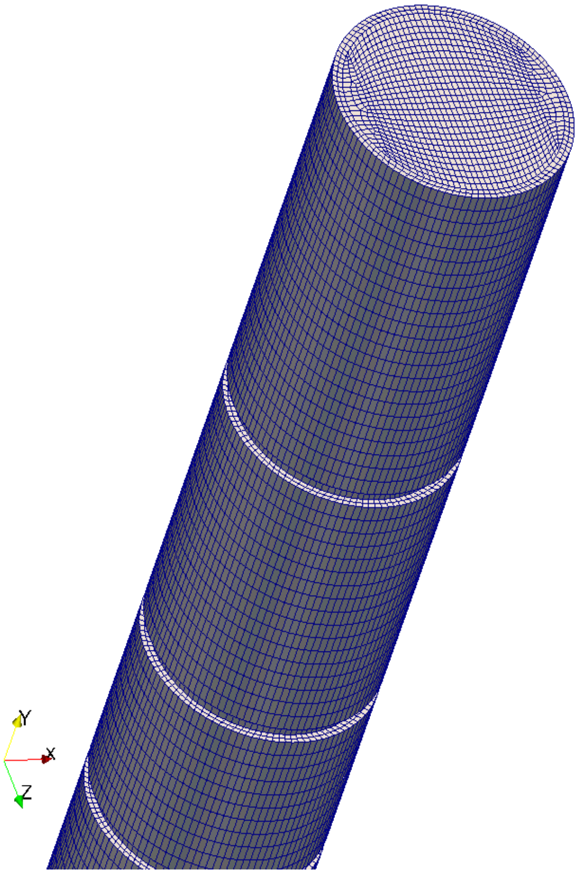


Figure 17: *Closer view of the column outlet in the Owens study.*

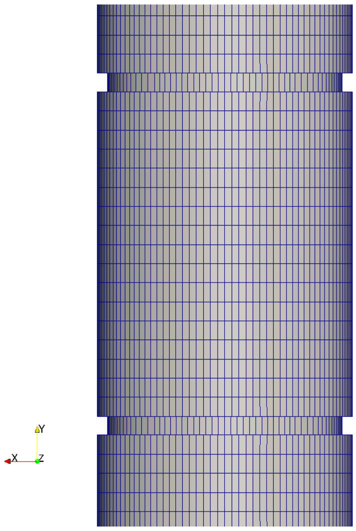


Figure 18: *Grid in the lower bed part. Wiper bands in the grid are modelled as walls.*

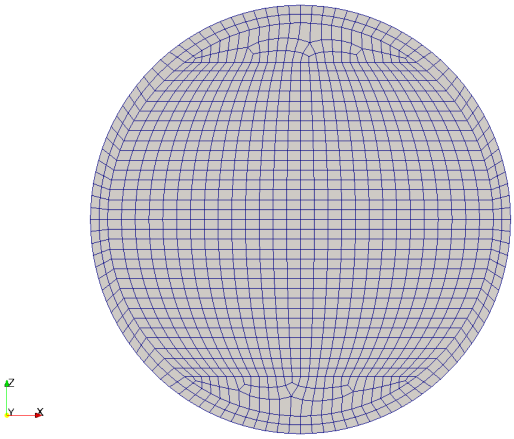


Figure 19: *Grid from the viewpoint above of the column.*

7.3.2 Porous medium definition

Flow aligned to Mellapak 250Y channels in the packing has a lower resistance than in the directions perpendicular to the packing metal sheets. This can be taken into account by the coefficients in equation (70) in corresponding directions. The porous media option in OpenFOAM is taken into account with the aid of the Darcy-Forchheimer equation Eqn. (70) through the source term in case of the explicit porosity treatment or through the matrix coefficients in case of the implicit porosity treatment.

For defining the coefficient in Eqn. (70), a least-squares curve fitting was applied on the experimental data from Ref. [24] presented in Fig. 50. This analysis gave, however, too large values for the viscous and inertial loss terms, because the

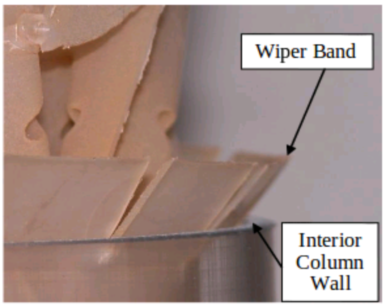


Figure 20: *Wiperband around the packing exceeds the column diameter.*

fitting could be made only in one dimension. Thus the experimental pressure drop curve in Fig. 50 cannot be predicted with the coefficients obtained. The coefficients obtained with aid of least-squares curve fitting could probably hold for the system with isotropic porosity. Since here the packed bed is modelled with an improved anisotropic porosity, the curve fitting is more complicated and coefficients are iteratively determined to satisfy the experimental curve.

The viscous term dominates the flow through the porous media when $Re < 100$ according to Chapt. 4.4. In this study, the flow Reynolds numbers according to the hydraulic diameter of the packing are in a range of $Re = \{562 - 3096\}$, thus it can be deducted that the inertial term dominates the flow through the porous media. The parabolic shape of the curve also indicates that the second order term i.e. inertial loss term could predict the pressure drop across the packed bed in relation to the gas flow velocity. This also holds with the results shown later. Therefore, only the inertial term is taken into account from the Darcy-Forchheimer Eqn. (70).

In OpenFOAM, the local coordinates are defined per zone for the porous media. The local coordinates are noted $(e1, e2, e3)$. The inclination angle of a corrugated sheet in the Mellapak 250Y structured packing is 45°, so the local coordinates are defined to be inclined 45° as compared to the global coordinates. In OpenFOAM, the local coordinates $(e1, e2, e3)$ and the coefficients of equation (70), d and f, are specified in the `constant/porousZones` file in the following way:

```
porosity1
{
    coordinateSystem
    {
        e1  (0.70710678 0.70710678 0);
        e2  (-0.70710678 0.70710678 0);
    }

    Darcy
    {
        d   d [0 -2 0 0 0 0 0] (0 0 0);
        f   f [0 -1 0 0 0 0 0] (-1000 3.5 -1000);
```

```
    }
}
```

Coordinate $e3$ is defined by a right hand rule from the vectors $(e1, e2)$. The values of coefficients d and f are defined in the directions of $(e1, e2, e3)$ where the coordinate $e2$ is chosen to be the direction of the flow channels in the structured packing. The coefficients of f in the directions $(e1, e3)$ are set to 1000 times the coefficient in the direction $e2$, while they describe the direction perpendicular to the packing sheet wall. The value of 1000 was taken from Raynal et al. [23] who studied structured packed beds with a three-scale approach and applied also porous media for the large scale computation (Chapt. 6.2). Factor f in the direction of the channels $f = 3.5$ was iteratively estimated from CFD calculation results reported in this work, by comparing the simulated pressure loss across the packing against the measured pressure loss values of Owens. A value for f_{e2} for the porosity model in OpenFOAM was thus obtained to be used for calculations of the large-scale column. The validity of the coefficient must be tested for the larger scale columns, too.

7.3.3 Boundary conditions

A schematic figure of the geometry of the column used in calculations for Owens case is shown in Fig. 21. It is shown that the gas flows in from the bottom of the column, where the inlet velocity is defined. The mean inlet velocities u_m are defined in Tab 4. The gas flows through the packed bed and finally out from the top of the column. The pressure is defined at the outlet of the column to be $p = 0$ Pa (or actually $\frac{p}{\rho} = 0\ \mathrm{m}^2/\mathrm{s}^2$ in OpenFOAM in case of incompressible flow). This is a gauge pressure i.e. pressure relative to the local atmospheric pressure. Thus the absolute pressure is obtained from gauge pressure by adding the value of the atmospheric pressure [30]. The kinematic viscosity has a value of $\nu = 1.56 \cdot 10^{-5}\ \mathrm{m}^2/\mathrm{s}$.

Table 4: *Reynolds number according to the hydraulic diameter for Mellapak 250Y structured packing Re_{dh} and according to the column diameter Re_D with different gas capacity factor F_S Eqn.* (5) *and corresponding mean velocity u_m in case of constant density.*

$F_S[(\mathrm{Pa})^{\frac{1}{2}}]$	0.61	0.915	1.22	1.53	1.83	2.14	2.44	2.75	3.05	3.36
$u_m[\mathrm{m/s}]$	0.56	0.855	1.14	1.43	1.71	2.0	2.28	2.57	2.85	3.14
Re_{dh}	562	843	1124	1410	1686	1972	2249	2535	2811	3096
Re_D	5329	7994	10659	13370	15988	18700	21318	24029	26647	29358

7.3.4 Fully developed inlet velocity profile

A fully developed flow profile at the inlet of the column was derived to make sure that the flow at the inlet is fluid dynamically realistic. The real column geometry in the experiments of Owens was such that there were an inlet duct at the bottom smaller than the column diameter, thus the inlet velocity profile would be more like a paraboloid than a bulk flow already at the inlet. The velocity values are largest at the centre of the inlet where the gas flows in. The fully developed velocity profile in a pipe can be presented as a paraboloid [31]. The profile resembles paraboloid, since the friction of the walls forces a zero velocity at the walls. The velocity magnitude increases away from the wall reaching its maximum value in the middle of the pipe. The equation for the paraboloid is formed so that the mass flux through the pipe corresponds the volume of the paraboloid. The mass flux is calculated with the aid of a gas capacity factor F_S, since the density ρ_G and the radius of the pipe R are known

$$F_S = u_m \sqrt{\rho_G} \tag{94}$$

From Eqn. (94) the mean velocity u_m is obtained. According to the mean velocity u_m the mass flux is

$$\dot{m} = \rho_G u_m A = \rho_G u_m \pi R^2 \tag{95}$$

On the other hand, the mass flux is the density multiplied with the integral of the velocity over the pipe cross section

$$\dot{m} = \rho \int\limits_0^{2\pi} \int\limits_0^{R} u(r) r dr d\varphi \tag{96}$$

As it is known that $u(r)$ is of a form

$$u(r) = ar^2 + br + c \tag{97}$$

Eqns. (95) and (96) were combined and the constants a, b, c were solved. The resulting equation for the parabolic inlet flow is

$$u(r) = -\frac{2u_m}{R^2} r^2 + 2u_m \tag{98}$$

In OpenFOAM, a flow profile defined in Eqn. (98) was set as an inlet boundary condition using a `funkySetFields` utility of `swak4Foam`, which is an external utility for OpenFOAM [54]. The file defining the profile `funkySetFieldsDict` is presented in Appendix C.

7.3.5 Selection of the turbulence model

In this section the selection of the turbulence model for the Owens column simulations is discussed. The hydraulic diameter of the packing is defined in equation (3). According to this equation and the values of specific area a and void fraction ϵ from Tab. 1 the hydraulic diameter for the Mellapak 250Y structured packing is $d_h = 0.0154$ m. The Reynolds number for the flow in the packing according to the hydraulic diameter d_h and mean velocity u_m can be calculated from

$$Re_{dh} = \frac{u_m d_h}{\nu} \tag{99}$$

The values of the Reynolds number for the Mellapak 250Y structured packing with a reference length of a hydraulic diameter $d_h = 0.0154$ m and Reynolds number for the column with reference length of column diameter $D = 0.146$ m at the measured velocities are presented in Tab. 4.

For internal flows, the critical Reynolds number above which the flow is turbulent, is $Re_{cr} \approx 2100 - 4000$ [30]. It is seen from the values of Re_D in Tab. 4, that according to the column Reynolds number Re_D the flow would be turbulent. According to the packing Reynolds number Re_{dh}, however, the flow is laminar at least in the lower mass flow rates. In some parts of the packing there is probably some turbulence. This turbulence, however, can not be modelled based on the global flow. This is deducted from the simulation results in Chapt. 7.3, since the SST $k - \omega$ turbulence model applied in calculations predicts probably too high turbulence values in the packing.

The SST $k - \omega$ model was chosen to be applied in the calculations since it is more or less a standard choice nowadays and because of good earlier experience with it. The properties of the turbulence models were explained in Chapt. 4.3.1. However, with the applied turbulence model turbulence cannot probably be modelled well in the packing area, since it overestimates the turbulence in the packing. This was noticed during the calculations shown later in Chapt. 7.3 and it is caused by the turbulent kinetic energy production term in Eqn. (50). The anisotropic porous media causes velocity gradients in the packing, which in turn increase the production of the turbulence kinetic energy, which is seen in Eqn. (50). Therefore, a modified turbulence model was created in OpenFOAM, where the flow in the porous media representing the packed bed was assumed laminar according to the Reynolds number in Tab. 4 and the flow in other parts of the column turbulent. Presumably the flow in the packing is turbulent to some extent, but in this case the flow is approximated as laminar. This was accomplished by setting a turbulent kinematic viscosity to zero $\nu_t = 0$ in the porous media. Therefore, a new user-defined turbulence model in OpenFOAM was implemented by modifying the existing SST $k - \omega$ model (see Appendix E). This model is in the current work is called "the modified SST k-ω" turbulence model. For example in Fluent fluid dynamics solver, a similar option to suppress the turbulent viscosity by setting $\nu_t = 0$ in porous media when appropriate, exists. Thus this approach was applied in this work too because of its simplicity.

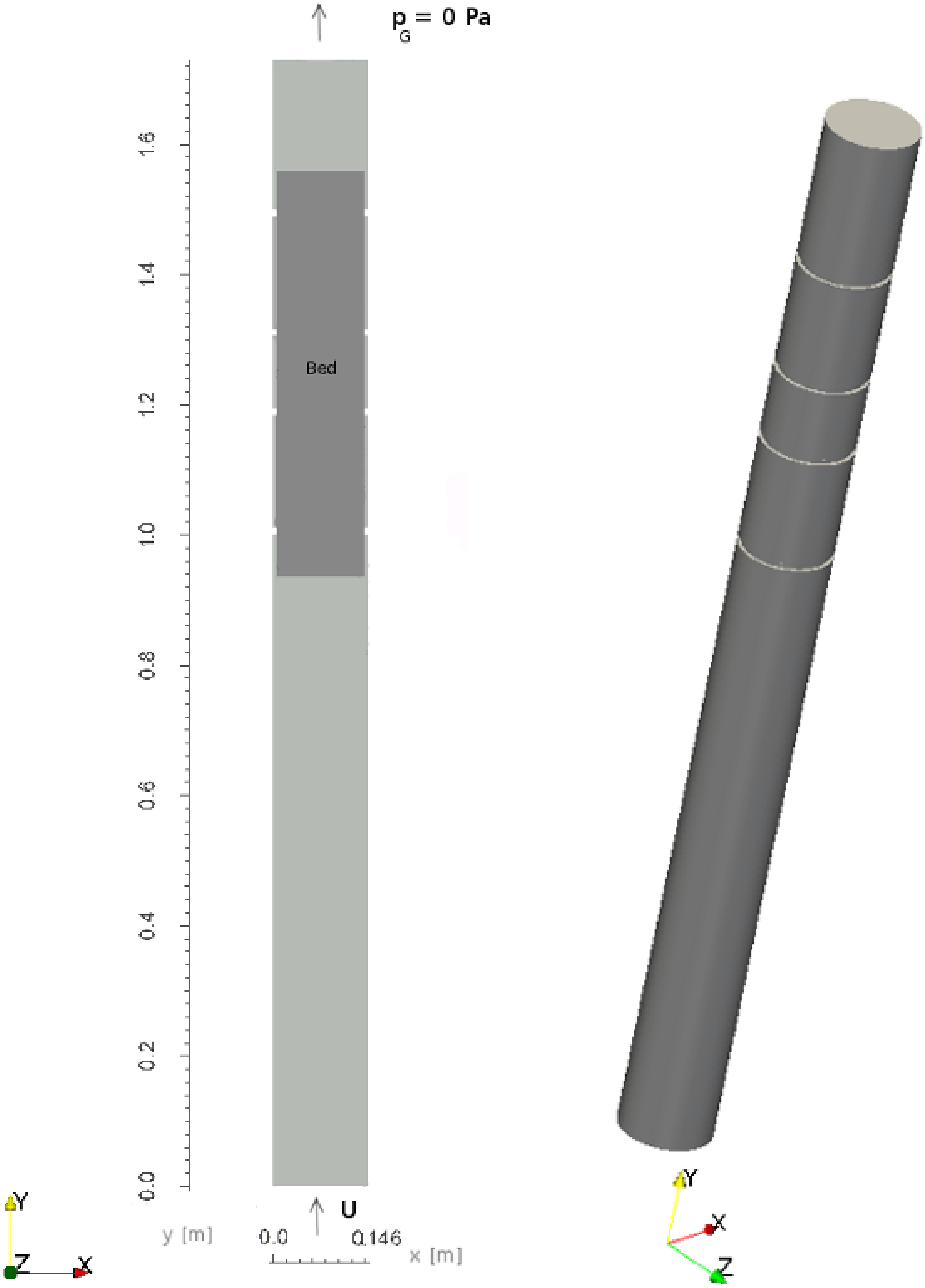


Figure 21: *Schematic figure of the geometry of the column used in calculations for Owens case.*

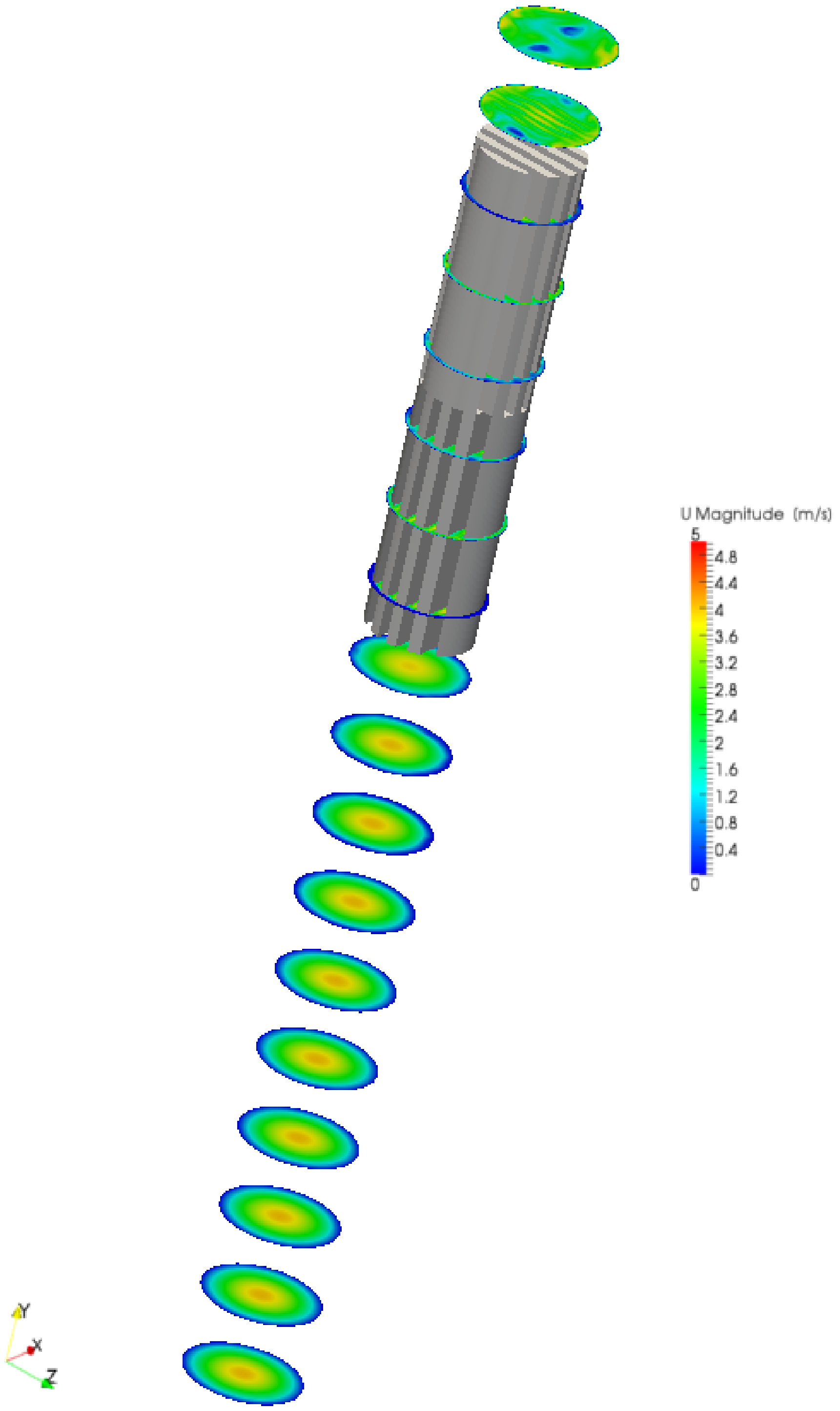


Figure 22: *Slices from the column. The packed bed which is modelled as a porous media can be seen as a grey block. Every second porous zone/channel in the bed is presented in the figure to show the structure more clearly.*

7.3.6 Grid independency study

Calculations with a coarse and a refined grid were made to investigate the effect of the grid on the results. The coarse grid has 231, 732 cells and the refined grid 1, 853, 856 cells. In Fig. 23 the pressure profile for both grids along the y-coordinate along the centre axis of the column is presented. In Fig. 24, velocity profiles at different cross sections are presented. The slight differences between the coarse and fine grids can be seen in Fig. 24. The velocity profile of the coarser grid in 24a is slightly wider and lower, which is because of the greater numerical diffusion induced by the coarser grid. Also, near the walls the velocity profile for the fine grid is smooth but for the coarse grid a clear change in profile curve direction is noticed. This is due to the wall function used, while the wall function uses information of the boundary layer form and the boundary layer is not exactly solved. Wall functions were briefly discussed in Chapt. 4.3.3 However, the results show that essentially the calculated values are the same for both grids thus the coarse grid can be used for the calculations.

Refining the mesh was conducted with the `refineMesh` utility in OpenFOAM [49], which automatically splits the cells in every direction.

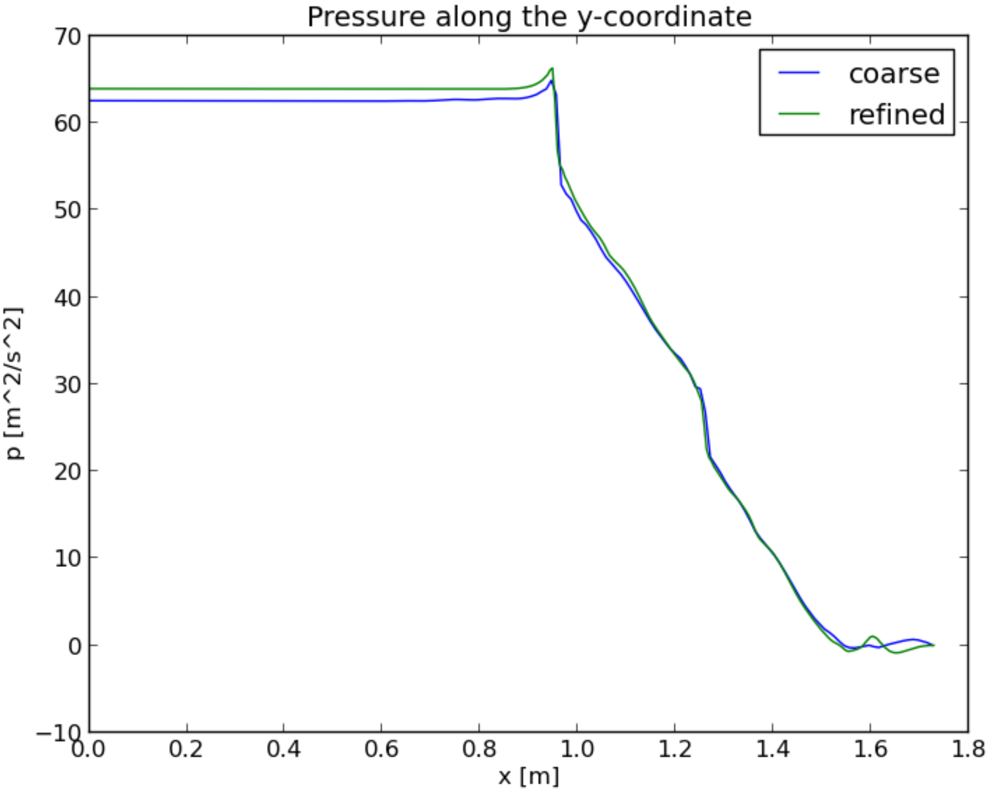
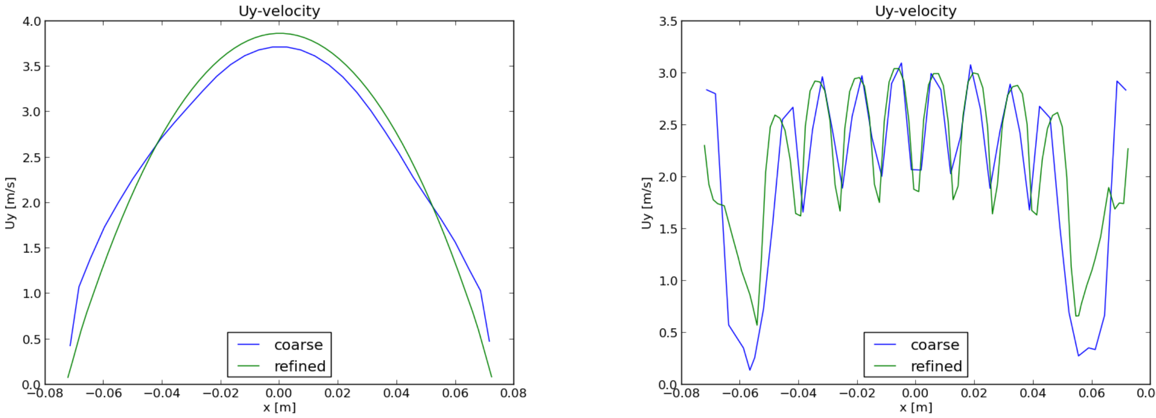


Figure 23: *Pressure profile along the column height.*



(a) Velocity along the x-coordinate in the middle of the column at height $y = 0.9$ m, i.e. just before the packed bed.

(b) Velocity along the x-coordinate in the middle of the column at a height $y = 1.4$ m i.e. in the upper part of the packed bed.

Figure 24: Velocity profiles of various cross sections of the column for the coarse and fine grids.

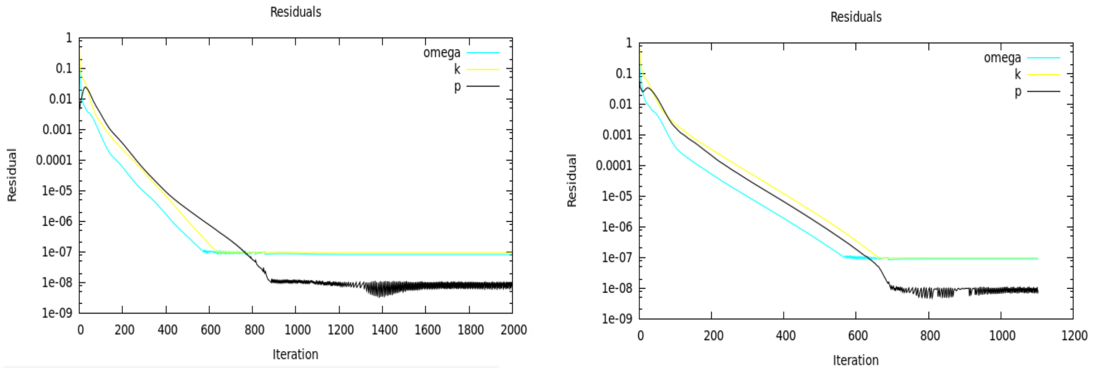
7.3.7 Results

Table 5: *The cases presented in Figs. 31 - 44.*

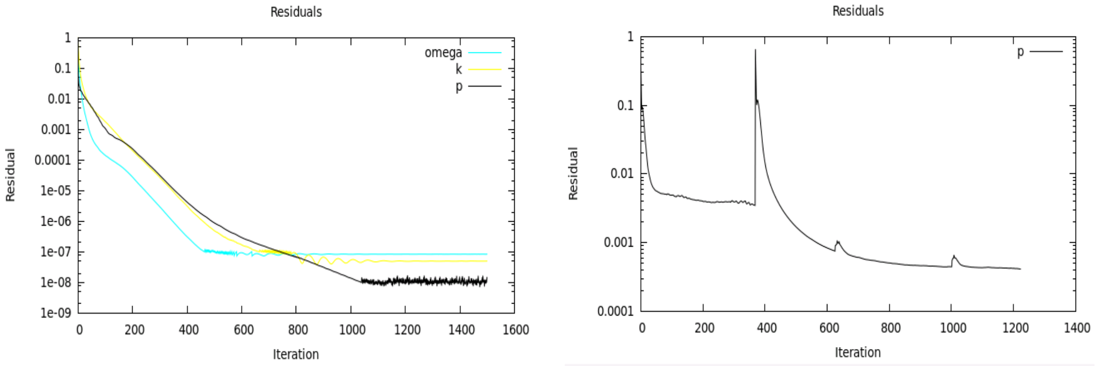
(a)	Turbulent case with a bulk flow inlet. The SST $k - \omega$ turbulence model is applied.
(b)	Turbulent case with a fully developed inlet flow. The SST $k - \omega$ turbulence model is applied.
(c)	Turbulent case with a fully developed inlet flow, the modified SST $k - \omega$ turbulence model is applied.
(d)	Laminar case with a fully developed inlet flow.

In this chapter the computational results conducted with the coarser grid with $231,732$ cells are presented. The simulation cases are presented in Tab. 5. In Figs. 31 – 44 the cases are referred with the corresponding letters in the Tab. 5. The simulations were made with all the velocities presented in Tab. 4. In Figs. 26 – 48 results only with $u = 2$ m/s are presented. The convergence history of each simulation case is presented in Fig. 25.

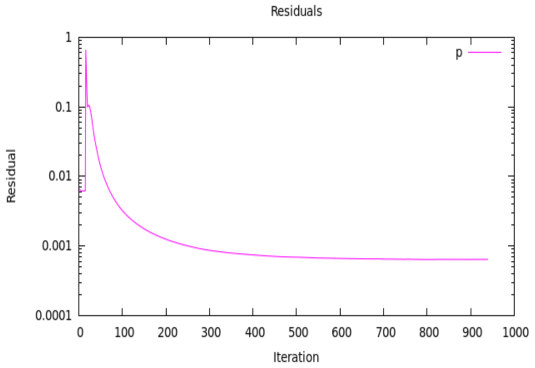
In Figs. 26 – 30 an overview of the flow situation in the column is given. The presented figures are from case (d). It gives though a good understanding of all the computed cases. In Fig. 26 slices from the whole column are shown to give an overview of the packing orientations and the flow field in the column. The area of the packed bed is shown with the grey bands, which present the every second porous zone in each layer. In Fig. 27 one cross section of the column in the packing is presented with vectors coloured by the velocity in the x-direction. It can be seen that the zones of porous media are directing the flow well, as if there were channels along which the gas could flow. In Figs. 28 – 30 the flow field in the different parts of the packing is shown with vectors. It can be seen from Fig. 28 how the gas flows in the column directly upwards until in the packing the gas flows along the channels, i.e. through the towards resistance. Fig. 29 demonstrates how the upper packing layer is rotated by 90° with respect to the lower layer, and how the flow direction changes respectively. In Fig. 30 the outflow from the packing is shown and it can be seen how the gas velocity fluctuates after the packed bed.



(a) *Coarse mesh:* 231, 732 *cells.* (b) *Coarse mesh:* 231, 732 *cells.*



(c) *Coarse mesh:* 231, 732 *cells.* (d) *Coarse mesh:* 231, 732 *cells.*



(e) *Case (d) with refined mesh:* 1, 853, 856 *cells.*

Figure 25: Convergence history of each calculated case.

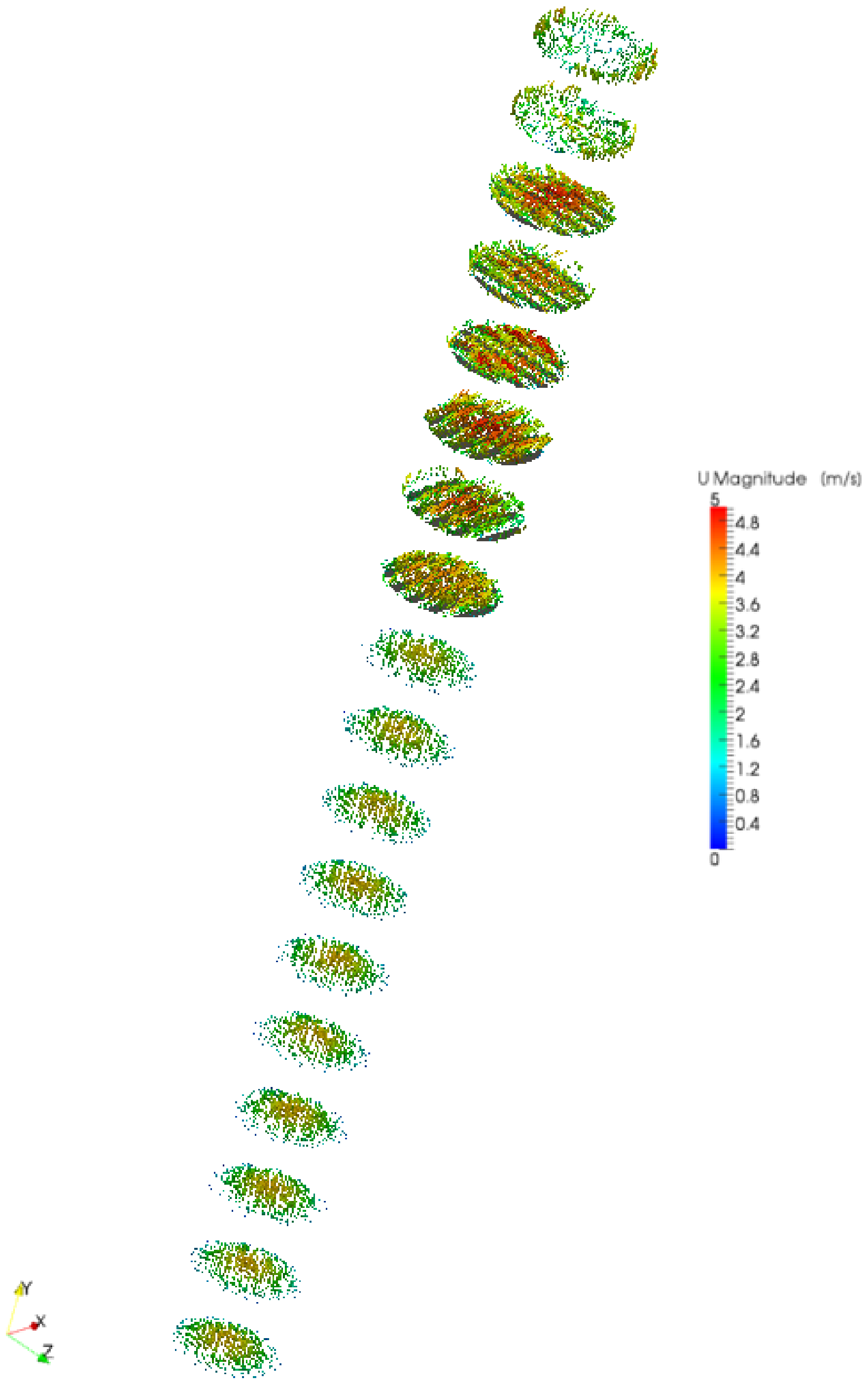


Figure 26: *Case (d): Cross sections of the column with vectors presenting velocity magnitude. In the cross sections of the packing the porous zones are shown: every second porous zone in the bed is presented as a grey slice.*

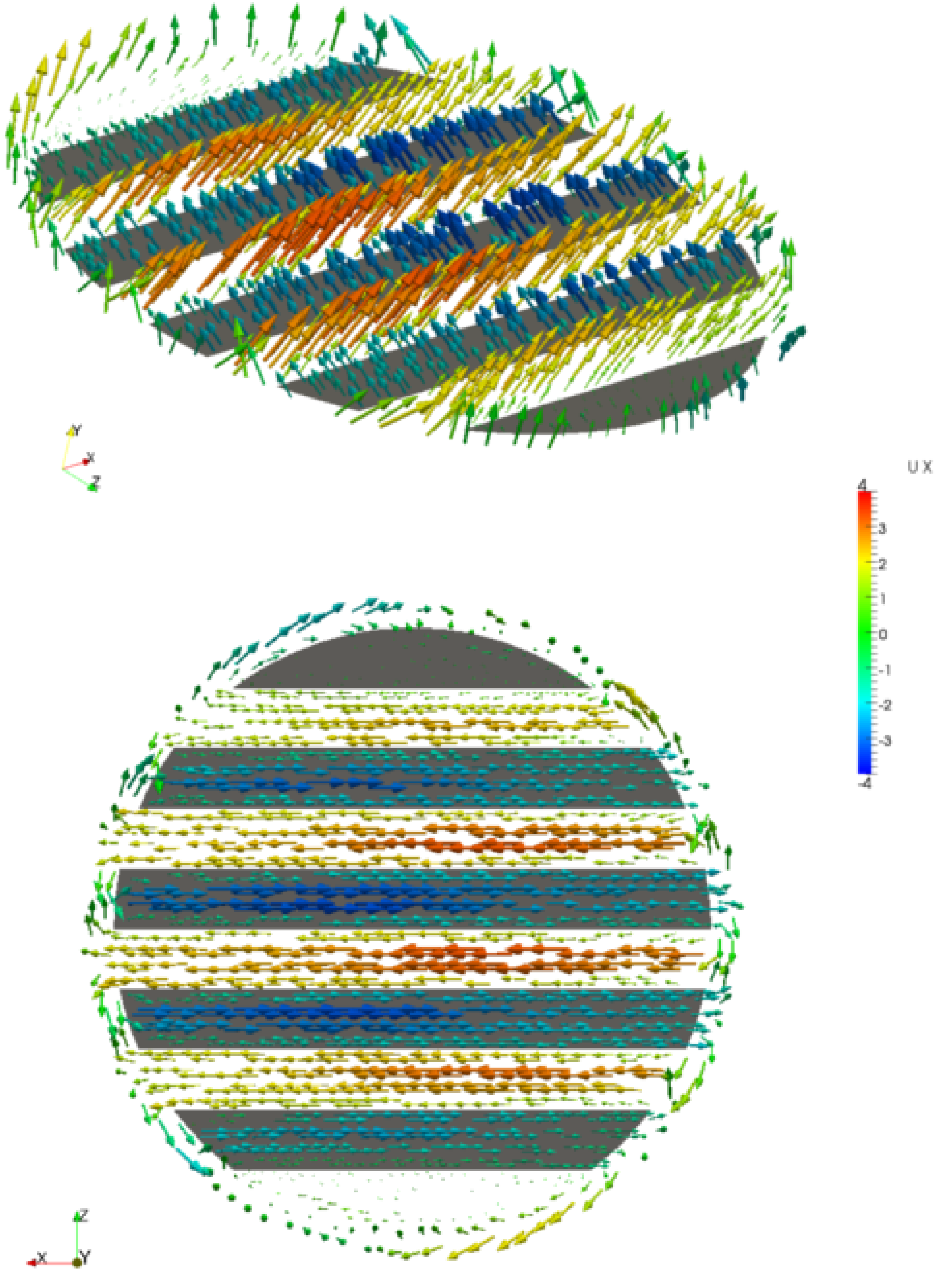


Figure 27: *Case (d): Cross section of the column in the packing with vectors presenting velocity in the x-direction. Every second porous zone in the bed is presented as a grey slice.*

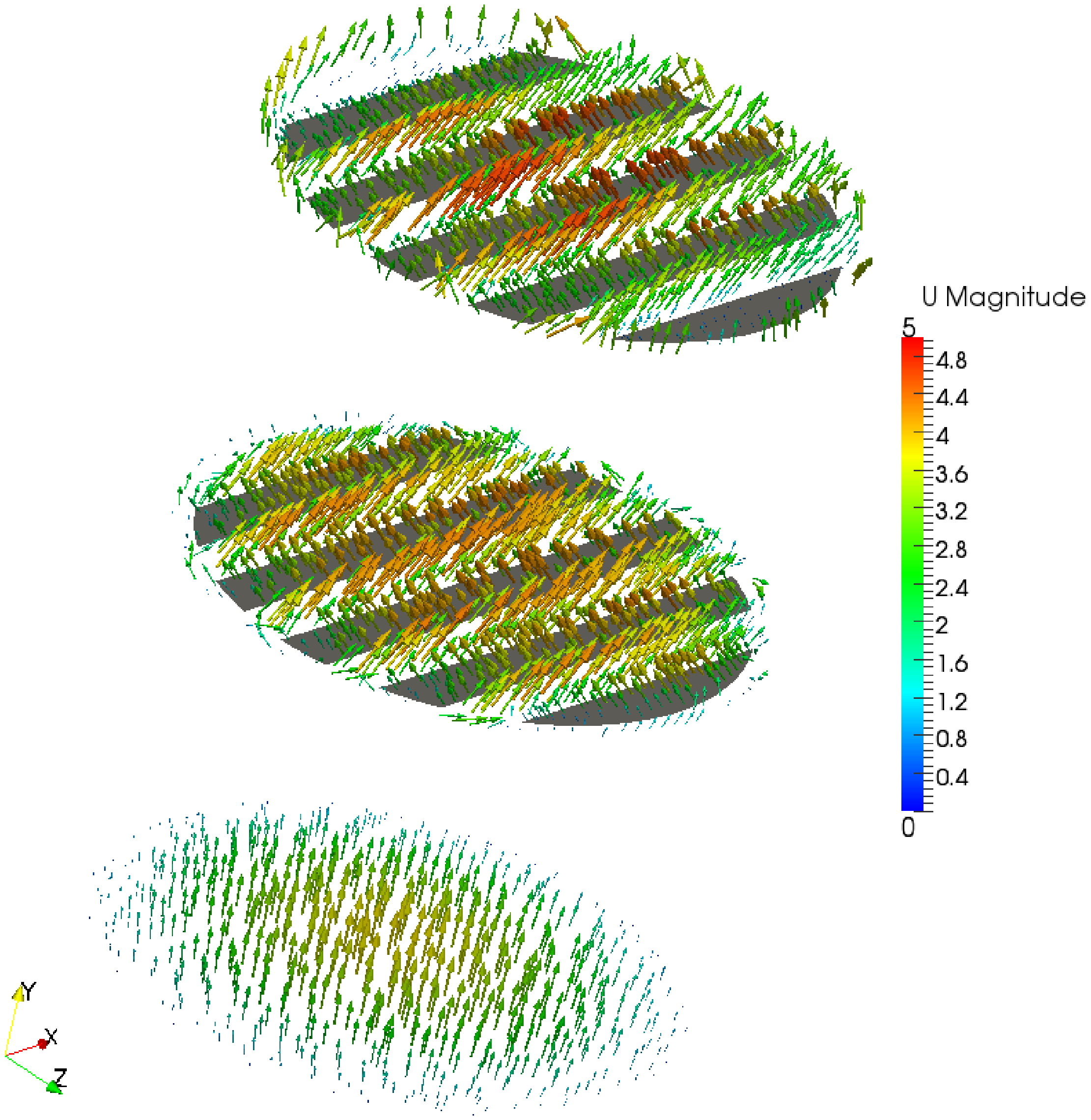


Figure 28: *Case (d): Cross sections of the column at the inlet of the packing. In the packing the gas flows along the channels. Every second porous zone in the bed is presented as a grey slice.*

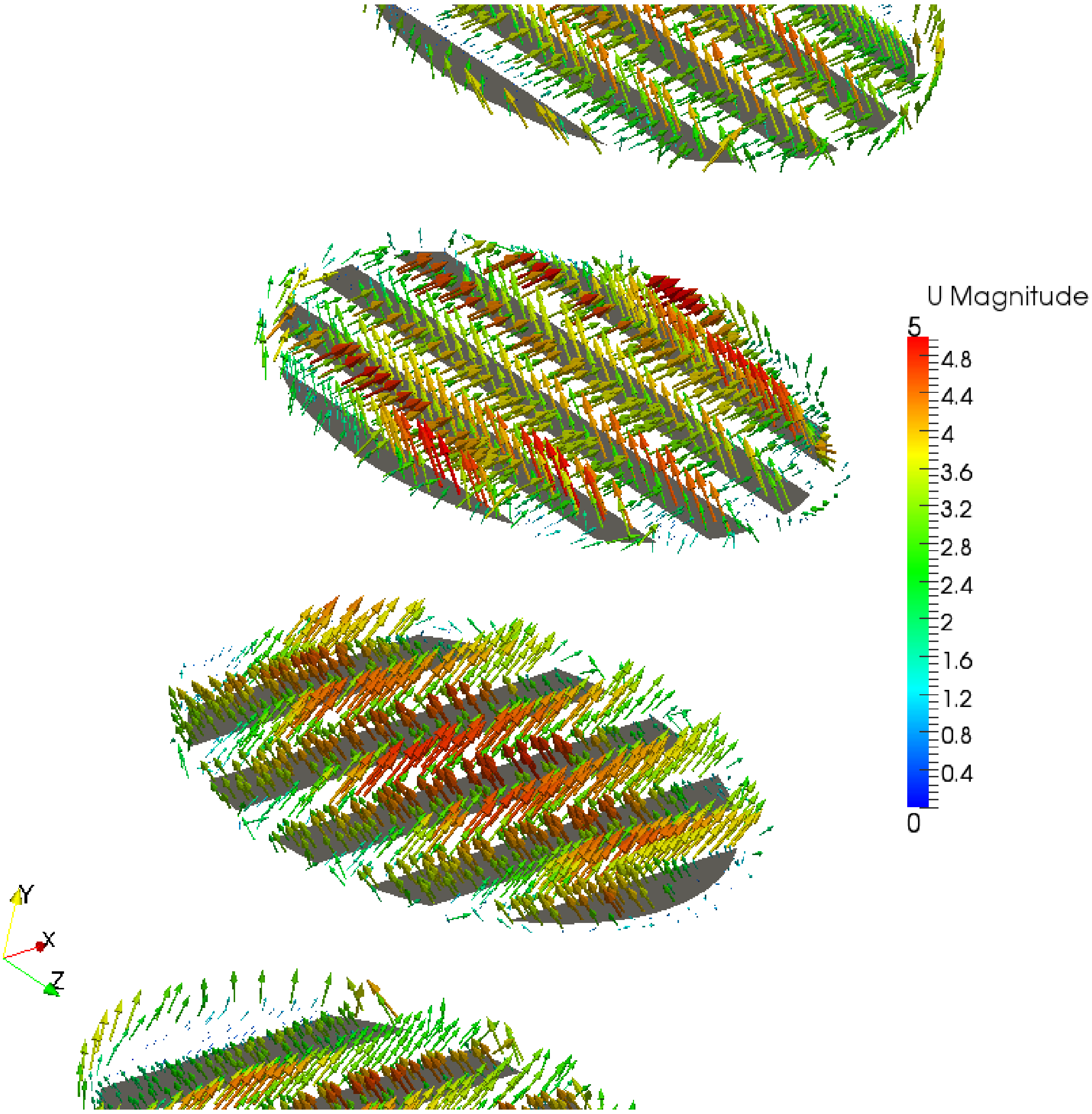


Figure 29: *Case (d): Cross sections of the column in the middle of the packing. The lower packing part ends and the upper packing part starts. The upper packing part is rotated by 90°. Every second porous zone in the bed is presented as a grey slice.*

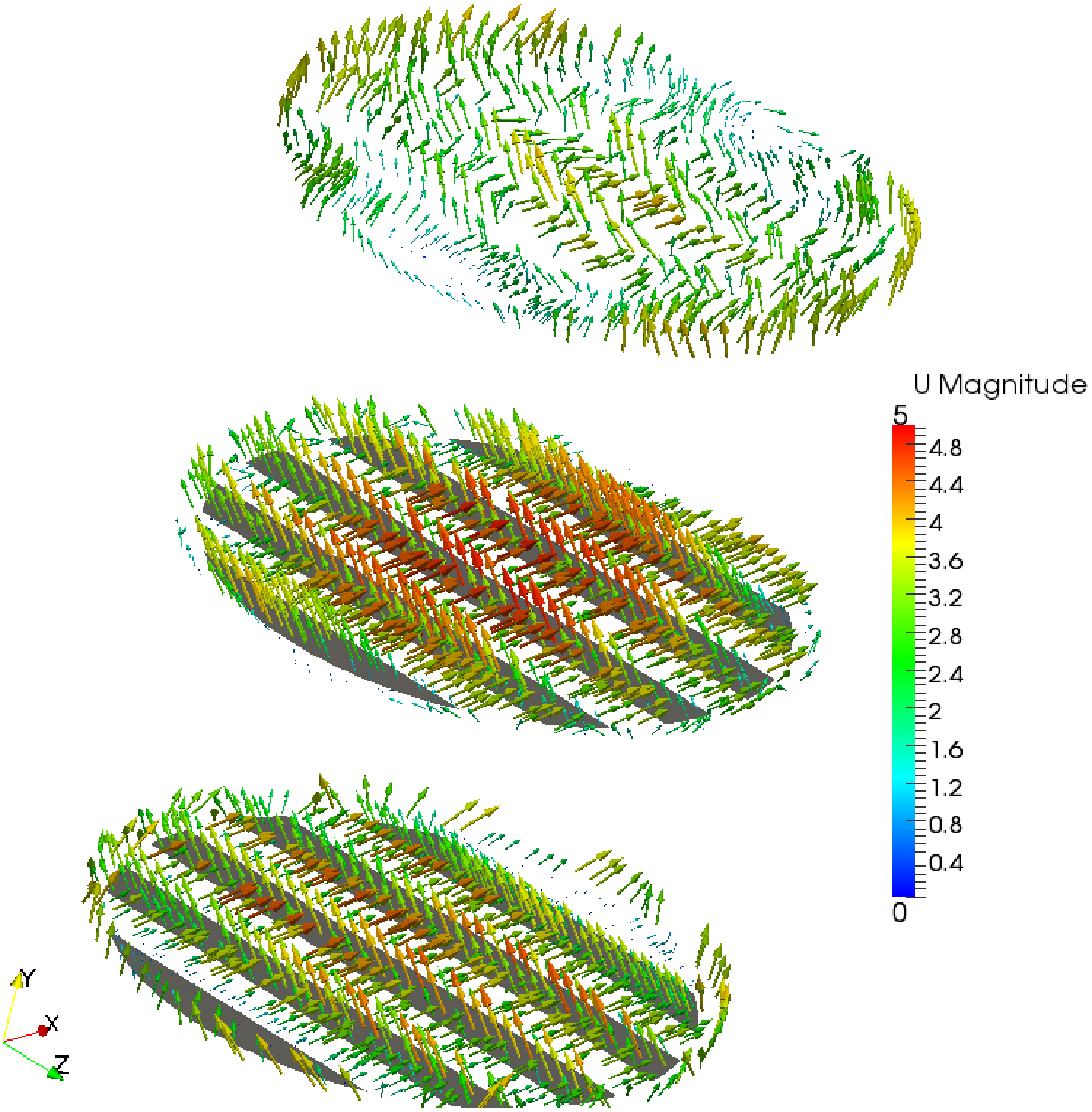


Figure 30: *Case (d): Cross sections of the column at the outlet of the packing. Every second porous zone in the bed is presented as a grey slice.*

Contour plots on vertical cross-sections

In Figs. 31 – 44 the quantity fields of different cases are presented on the $z = 0$ plane i.e. in the middle of the column. It is seen that the gas flow becomes more tortuous when the gas enters the packed bed. The gas flows in a direction of the flow channels, because of the directed resistance in the anisotropic porous medium presenting the packed bed. In the direction of the flow channels, the flow accelerates due to the continuity. Figs. 31 and 32 show the velocity magnitude at the cross section in different cases. It is seen that the wiperbands affect the flow by increasing the flow velocity by narrowing the column. In this calculation the most important quantity is the pressure loss in order to obtain correct resistance coefficients for the bed. In a small column like in this case, the wiperbands have presumably a relatively larger effect on the flow and in the pressure loss than in the larger scale columns. Therefore, because the pressure loss is needed to be modelled as accurately as possible, it is essential to take the wiperbands into account. It is seen that in the wholly turbulent cases Figs. 31*a* and 31*b* the velocity magnitude differences smooth out in the bed more than in the case 32*c*, which is the modified turbulent case i.e. laminar in the bed and in the case 32*d* which is calculated as wholly laminar. This occurs because of the diffusion caused by the turbulence. In Figs. 31*b* – 32*d* a fully developed flow profile at the empty part of the column below the bed is seen. Whether the flow is fully developed or a bulk flow, does not seem to affect on the flow field in the bed. In turn, whether the turbulence is taken into account or not, seems to have a significant effect on the flow field. In Fig. 32*d* a laminar boundary layer before the bed is seen, as the boundary layer starts to fluctuate before the bed. This may occur because according to the Reynolds number of the empty column Re_D the flow is turbulent, but in Fig. 32*d* the flow is calculated as laminar.

In Figs. 33 – 38 the velocity components in x-, y- and z-directions are presented. In Figs. 33 – 34 the velocity in x-direction is shown. Due to the current viewpoint in the xy-plane the velocity differences in porous zones presenting flow channels are not seen. A slight effect on the flow induced by wiper bands is seen in Figs. 33 – 38 and in the laminar bed and the wholly laminar calculations the fluctuations above the bed are seen. The grey parts in Figs. 36*c* and 36*d* are velocities in a negative y-direction i.e. downwards. Also here the laminar boundary layer fluctuations in Fig. 36*d* in the wholly laminar calculation are seen. These fluctuations in a larger scale may cause that the computations do not converge with the laminar flow assumption. This also indicates that the free region in the column is most probably turbulent, as is deducted from the Reynolds number in Tab. 4. In Figs. 37 and 38 the orientation of the upper porous zones along the z-direction representing the packing sheets is clearly seen as stripes in the upper part of the column presenting velocity in z-direction. The lower part has the porous zones in the x-direction so the velocity in the z-direction cannot be seen in these cross sections.

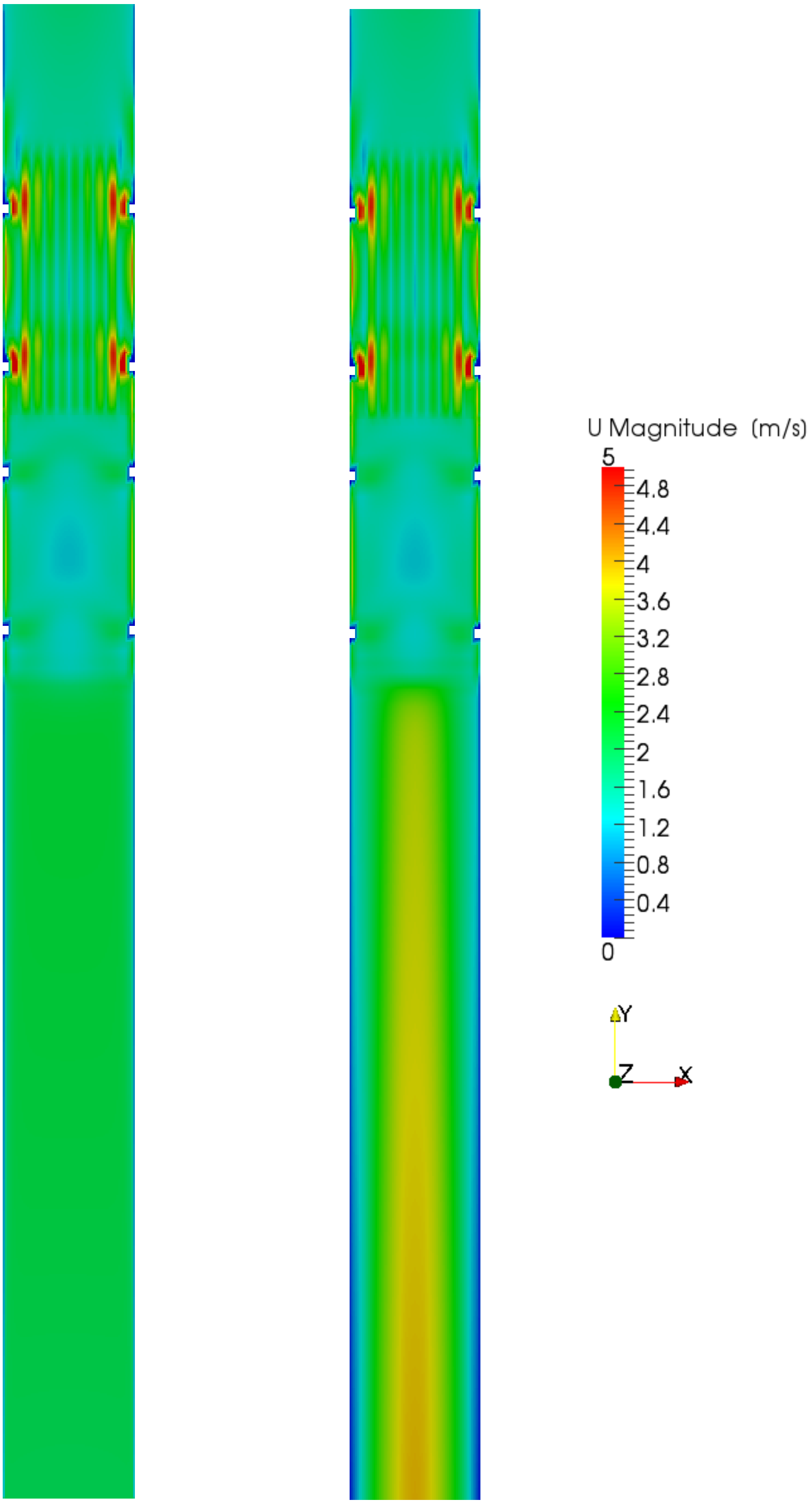
From Fig. 39*a* and 39*b* it is seen that there is turbulence in the packed bed and no or much less turbulence in the other parts of the column. According to the Reynolds numbers referred to the channel hydraulic diameter d_h in Tab. 4 the turbulence level in the packing area is clearly overestimated, since the free part of the column should be more turbulent than the packed bed. This is caused by the fact that the SST k-ω turbulence model includes the turbulence production term which is contains a velocity gradient of the global velocity field. Hence the changes in the flow direction i.e. velocity gradients due to the channels represented by anisotropic porous zones are interpreted as turbulence by the turbulence model. The anisotropic porous media does not contain real channels with walls which would damp the turbulence to some extent. The turbulence is predicted from the global flow. The anisotropic porous medium causes that velocity direction changes in the porous medium. These changes in directions cause velocity gradients and they increase the turbulence via the turbulence model and the turbulence is therefore overestimated.

In Fig. 41*a* and 41*b* the turbulence kinetic energy dissipation ω has the largest value at the packing area. Fig. 42*c* shows also a large value for ω at the packing area since it is not set to zero in the turbulence model. This has, however, no effect on the fluid flow, because $\nu_t = 0$.

In Figs. 43 and 44 the values of turbulent kinematic viscosity ν_t is presented. The difference between a bulk flow inlet and the fully developed inlet flow can be seen, since the bulk flow inlet results in an overestimation of turbulent kinematic viscosity in the column inlet.

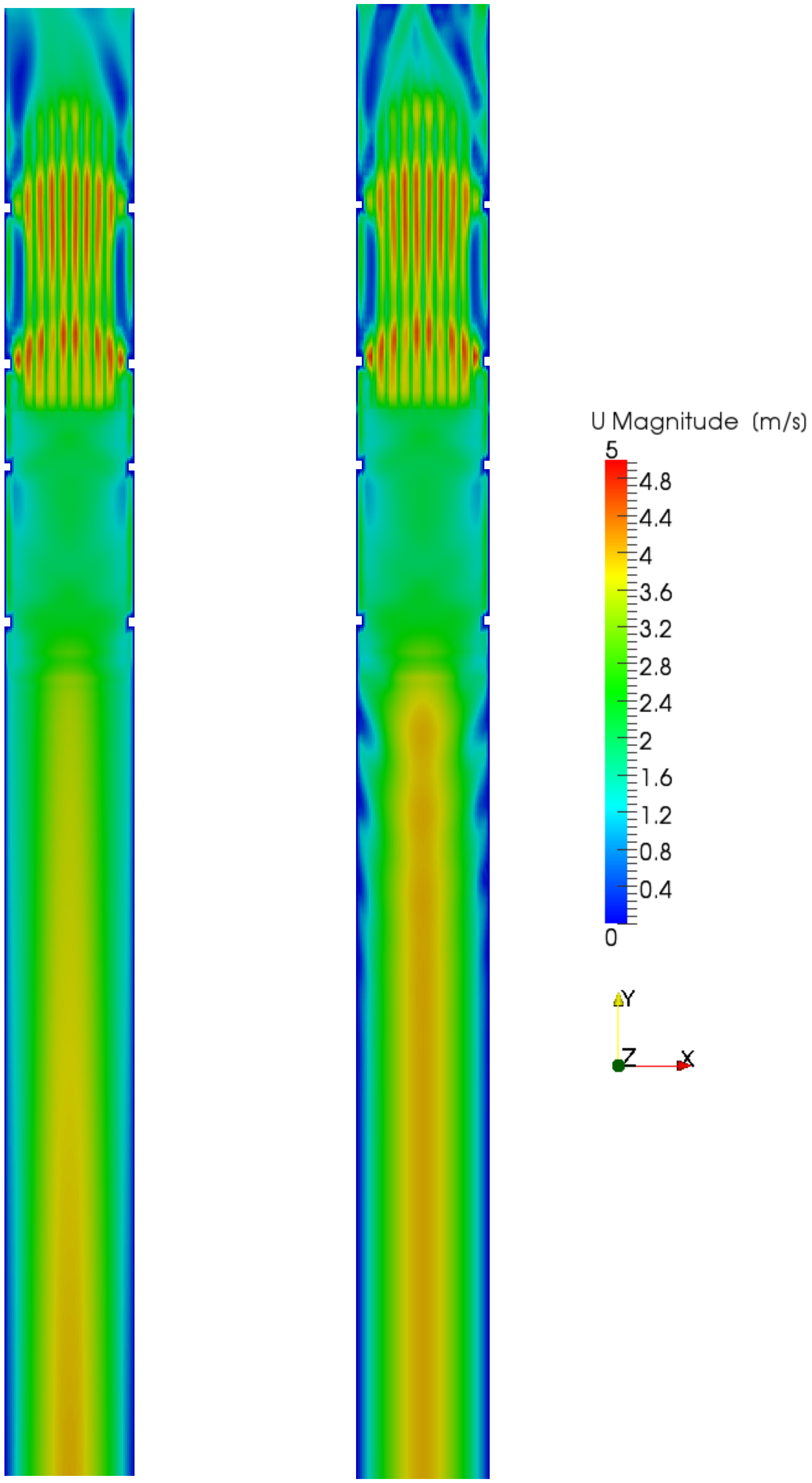
Contour plots on horizontal cross-sections

In Figs. 45 – 48 a velocity magnitude in cross sections at the different heights in the cases (a)–(d) are presented. The cross sections correspond to: $y = 0.9$ m is before the bed, $y = 1.1$ m is in the lower bed half, $y = 1.225$ m is exactly at the interface of the two bed halfs, $y = 1.4$ m is in the upper bed half, $y = 1.56$ m is at the outlet of the bed, and $y = 1.6$ m is above the bed. The velocity inlet profiles in different cases can be clearly seen as well as the changing the flow direction because of the two packing layers rotated 90° with respect to each other. It is also seen that in the cases including turbulence, the velocity differences smooth out rapidly because of the diffusion, corresponding to the laminar cases.



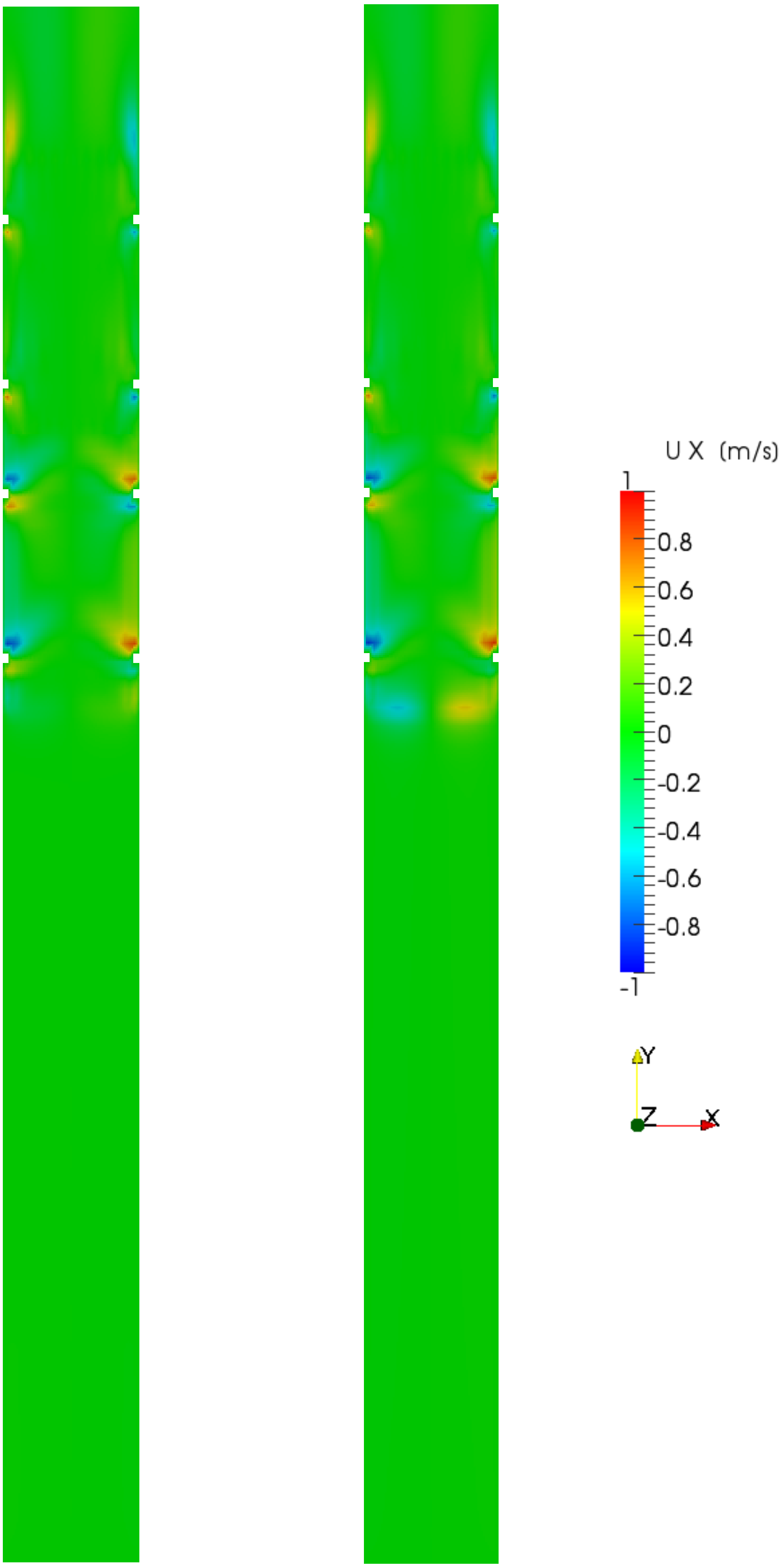
(a) (b)

Figure 31: *Cases (a) and (b). Velocity magnitude* (m/s) *at* $z = 0$.



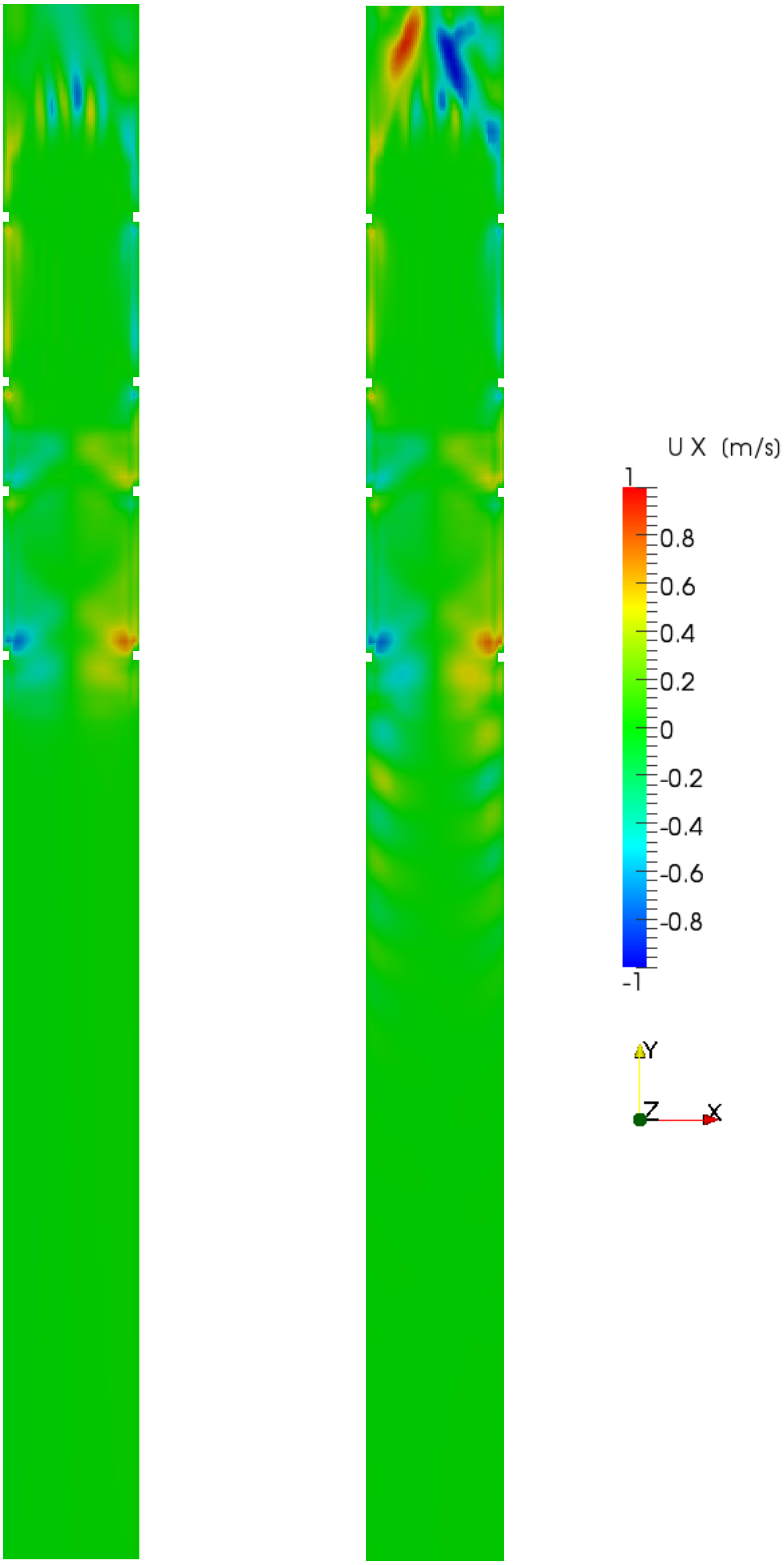
(c) (d)

Figure 32: *Cases (c) and (d). Velocity magnitude* (m/s) *at* $z = 0$.



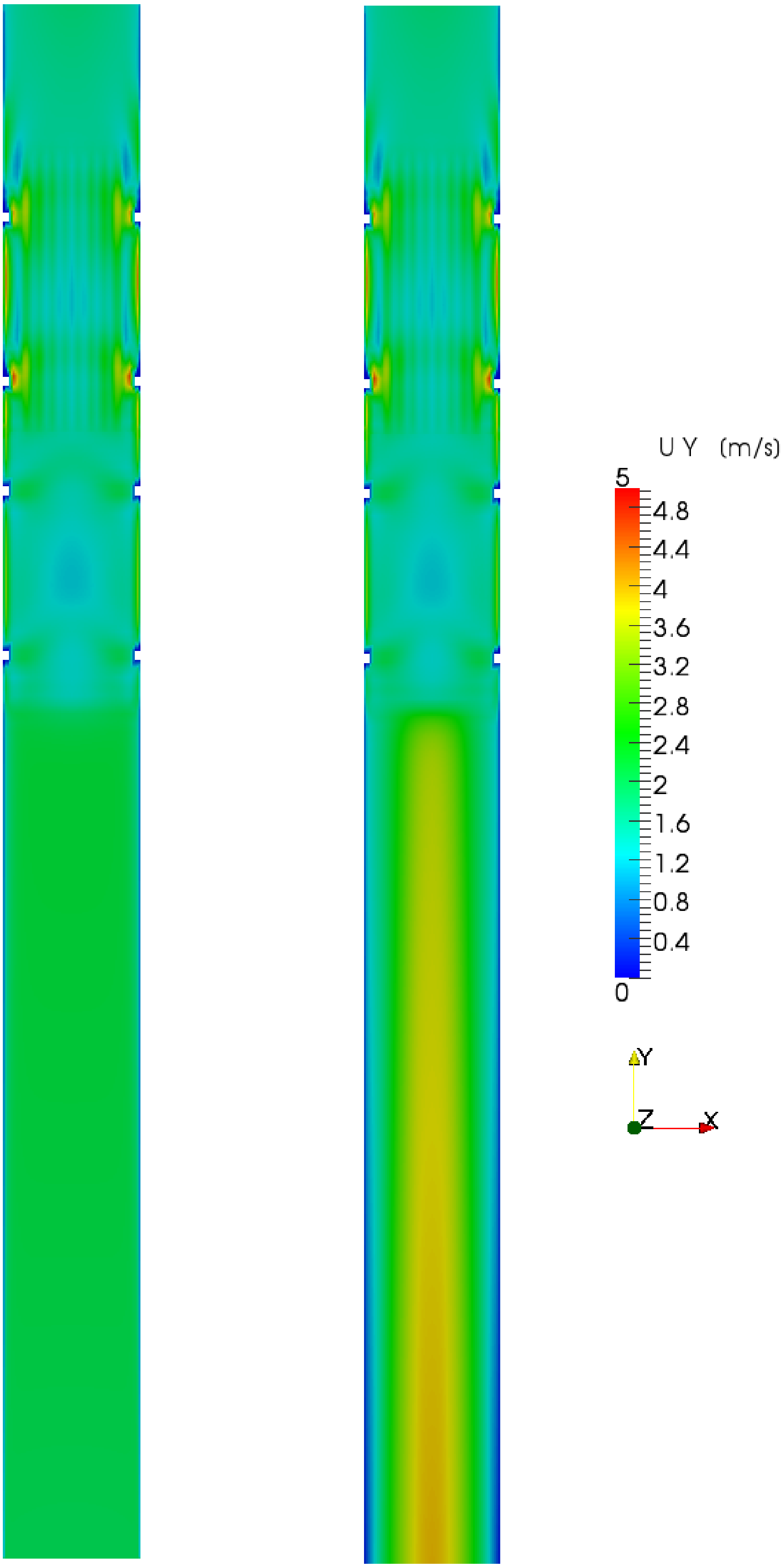
(a) (b)

Figure 33: *Cases (a) and (b). Velocity in the x-direction* u (m/s) *at* $z = 0$.



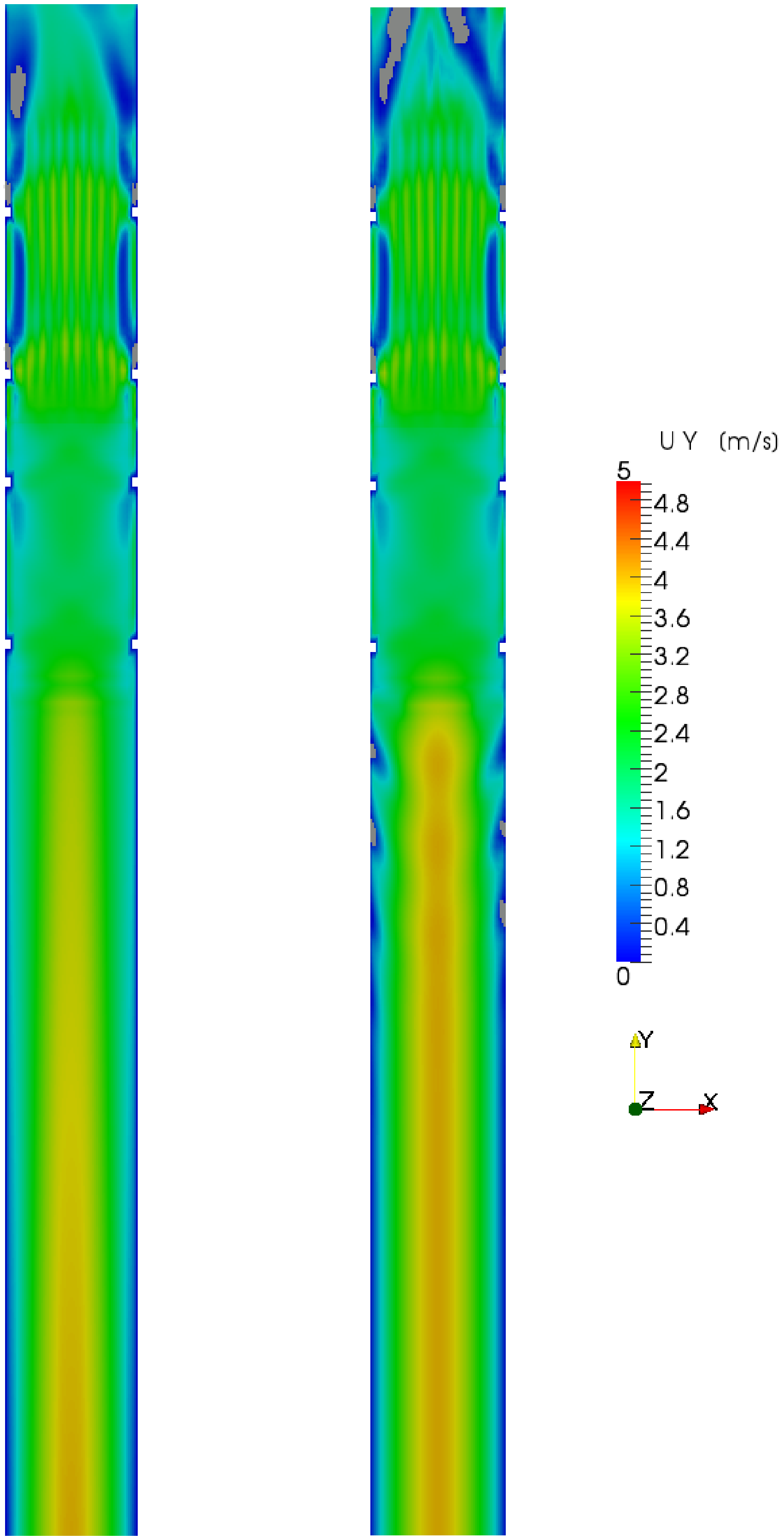
(*c*) (*d*)

Figure 34: *Cases (c) and (d). Velocity in the* x*-direction* u (m/s) *at* $z = 0$.



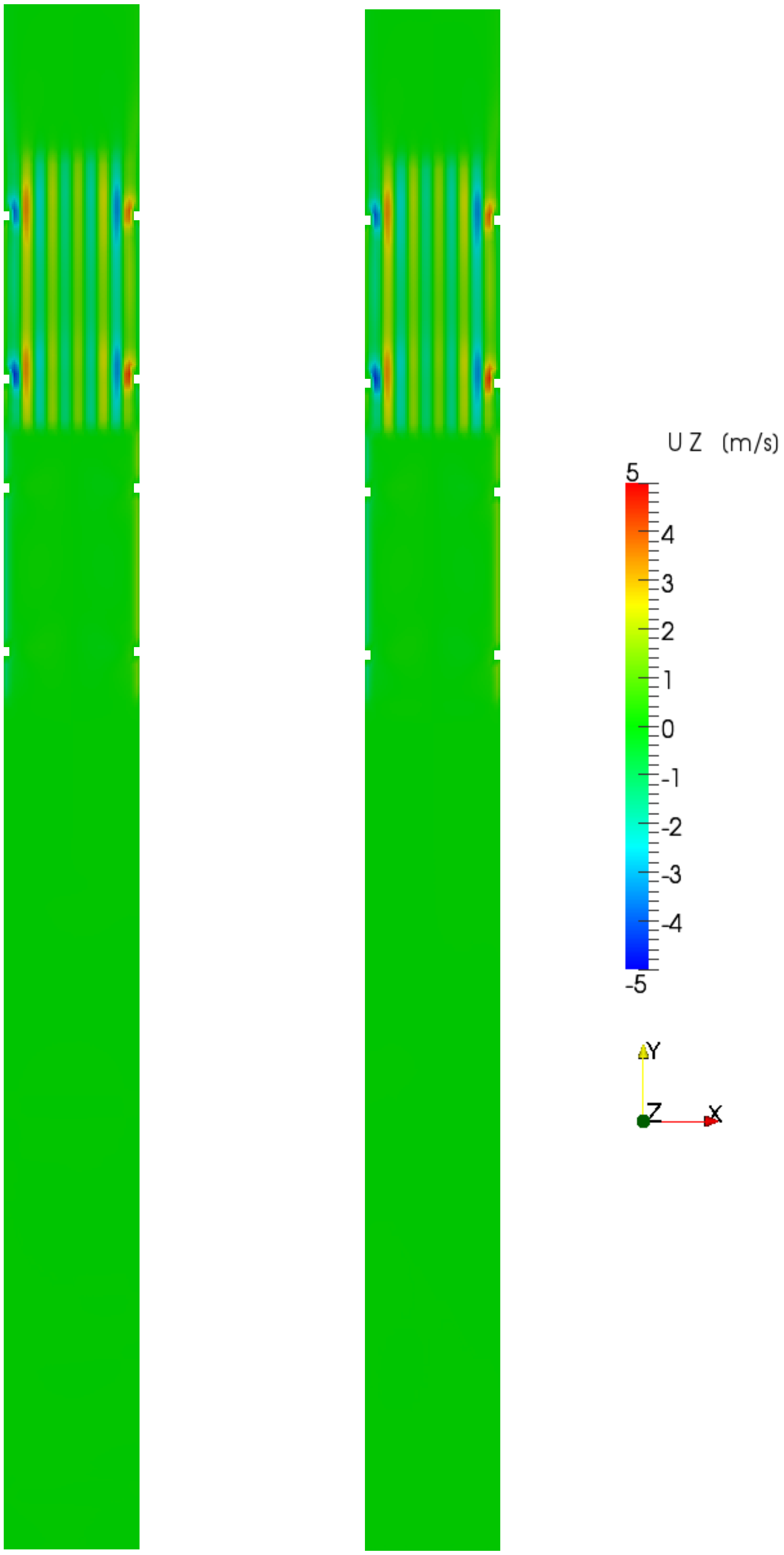
(a) (b)

Figure 35: *Cases (a) and (b). Velocity in the y-direction* v (m/s) *at* $z = 0$.



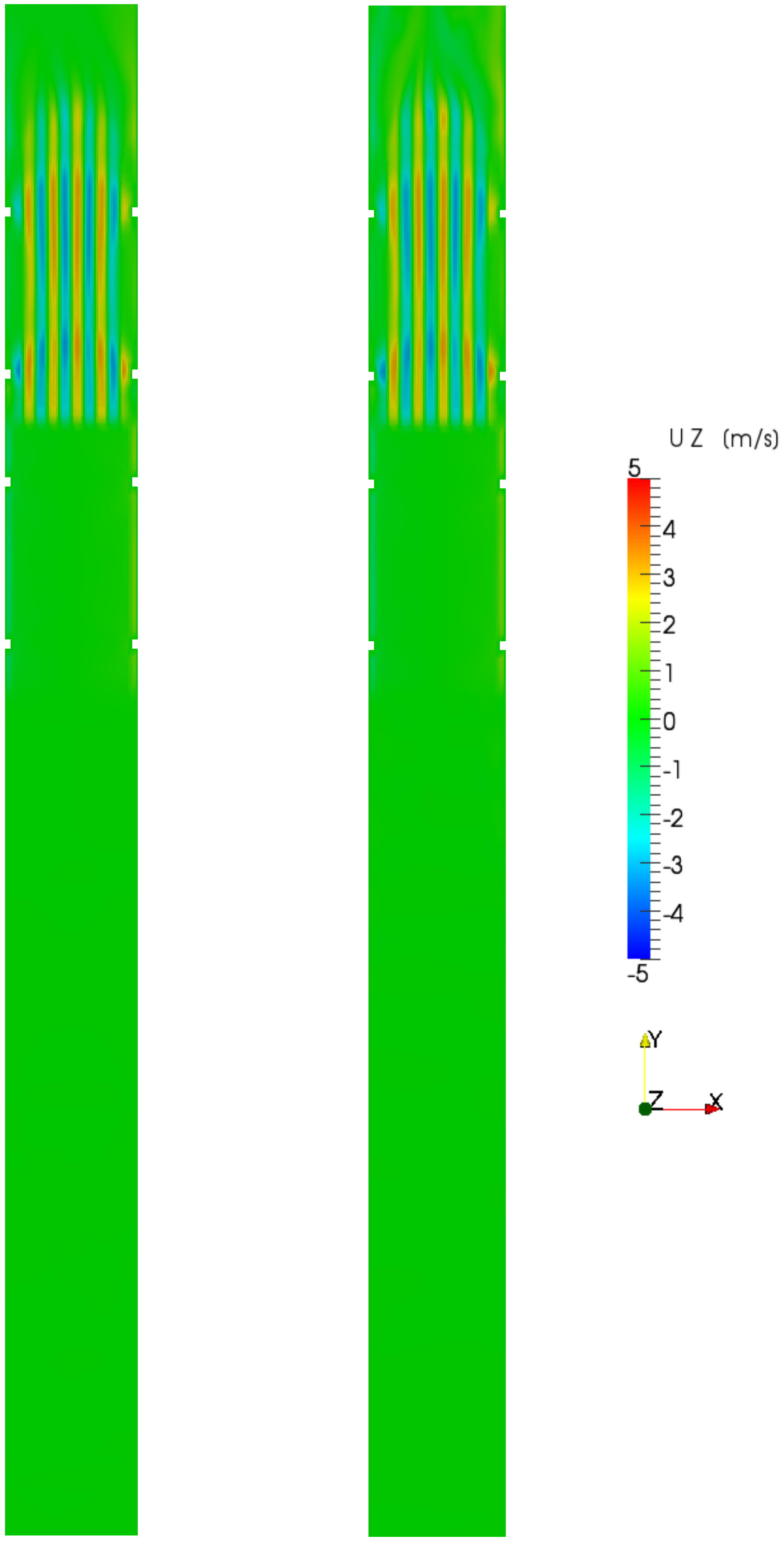
(c) (d)

Figure 36: *Cases (c) and (d). Velocity in the y-direction* v (m/s) *at* $z = 0$.



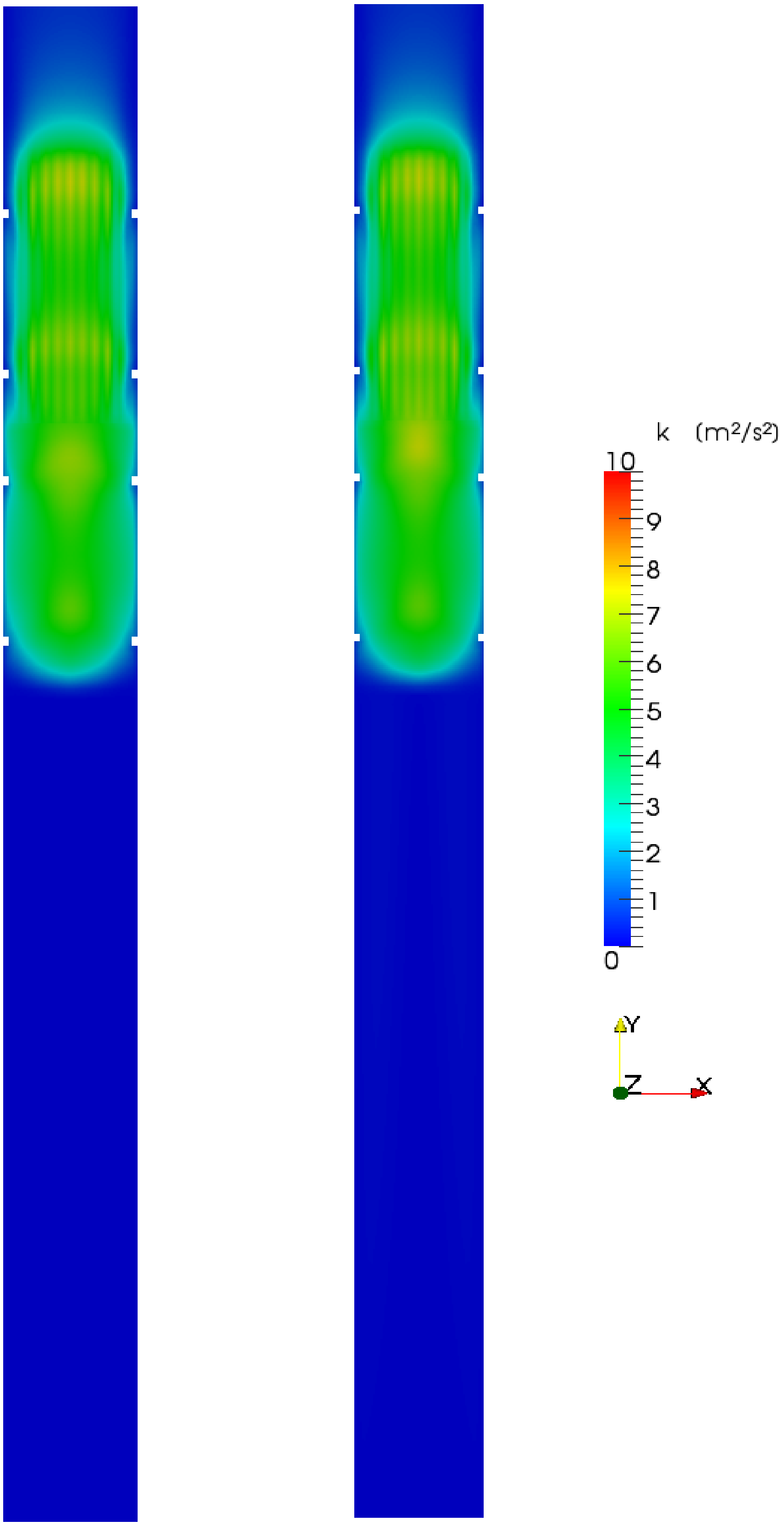
(a) (b)

Figure 37: *Cases (a) and (b). Velocity in the z-direction w* (m/s) *at* $z = 0$.



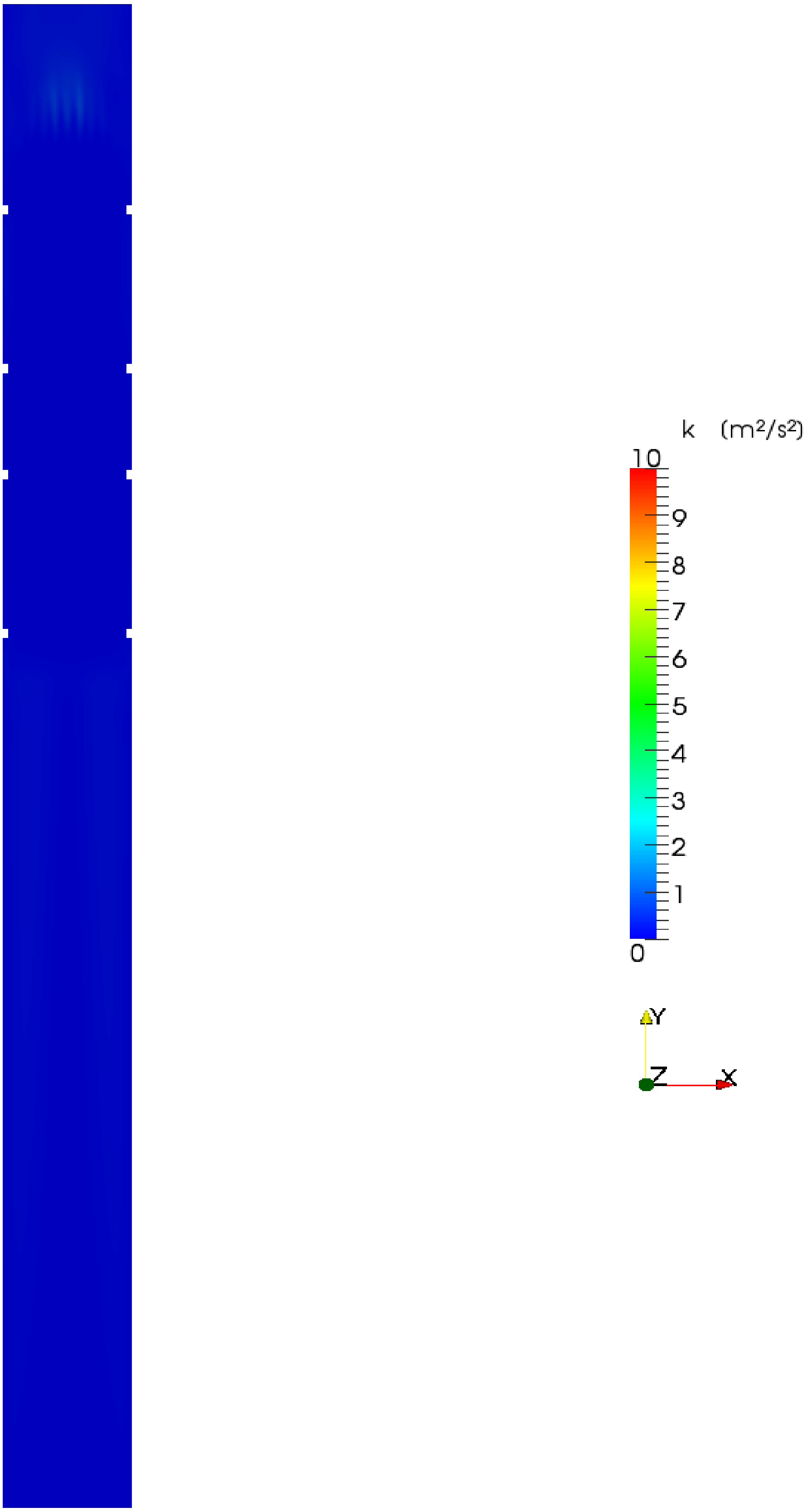
(*c*) (*d*)

Figure 38: *Cases (c) and (d). Velocity in the z-direction* w (m/s) *at* $z = 0$.



(a) (b)

Figure 39: *Cases (a) and (b). Turbulence kinetic energy* k $(\mathrm{m}^2/\mathrm{s}^2)$ *at* $z = 0$.



(c)

Figure 40: *Case (c). Turbulence kinetic energy* k ($\mathrm{m}^2/\mathrm{s}^2$) *at* $z = 0$.

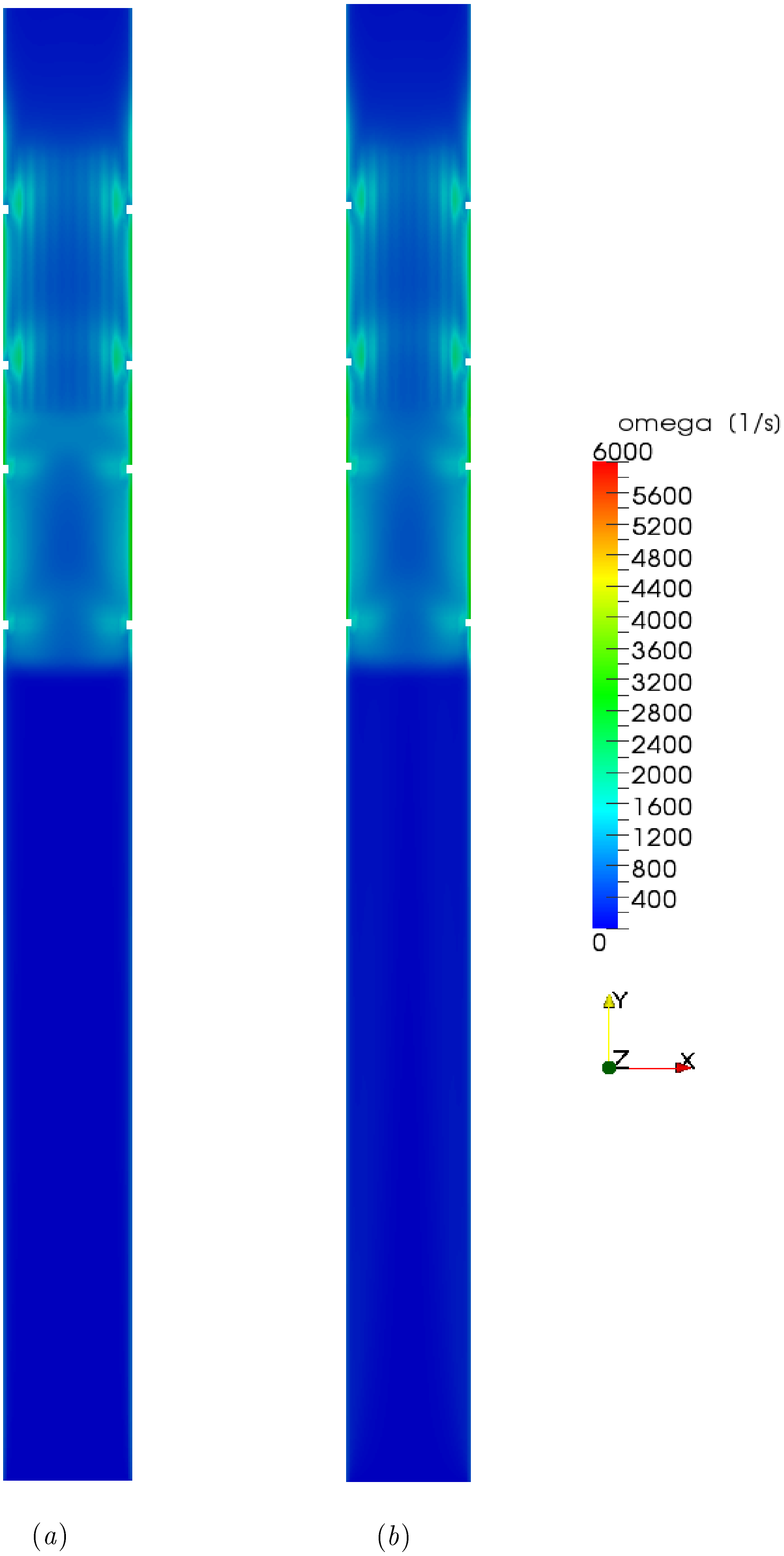
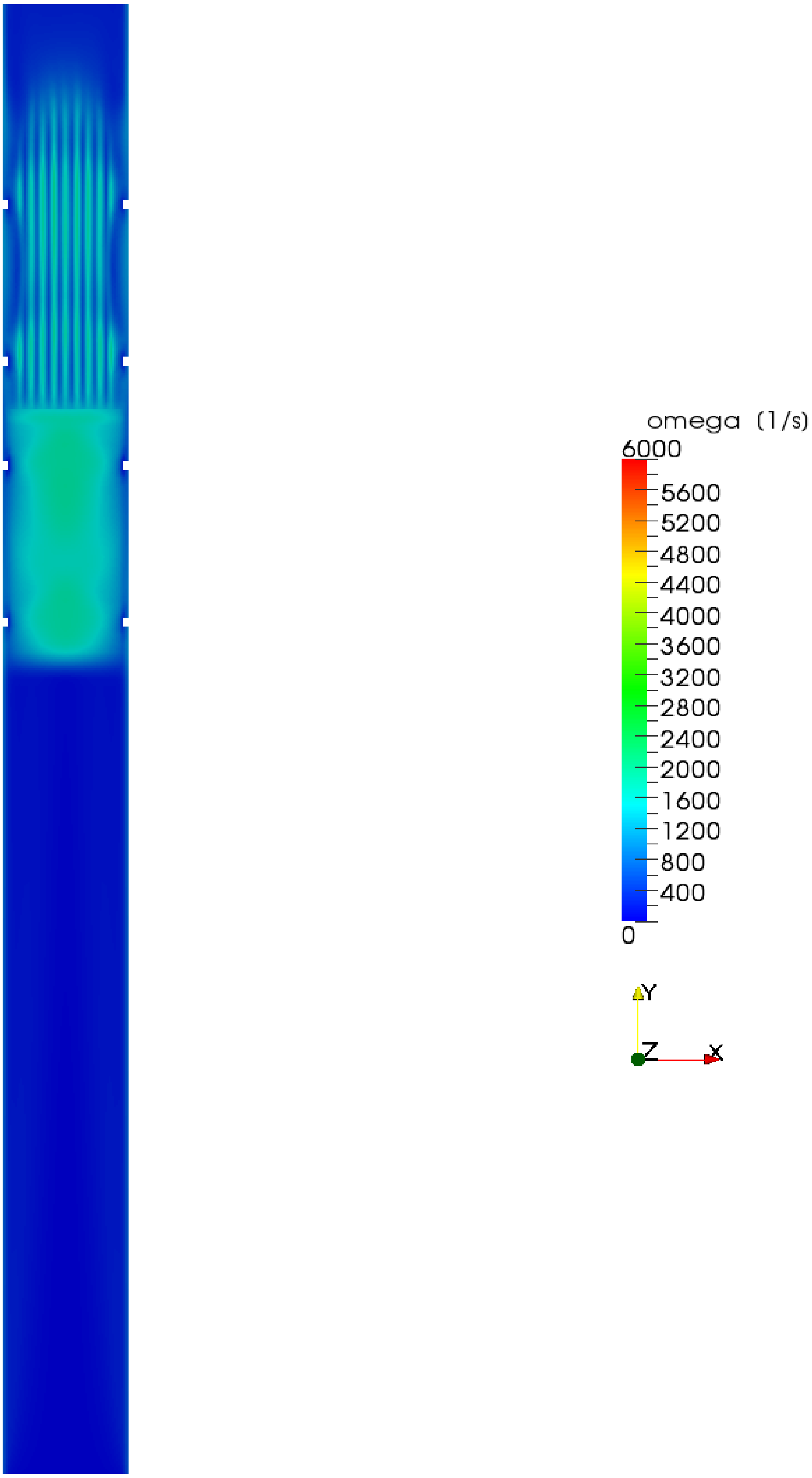
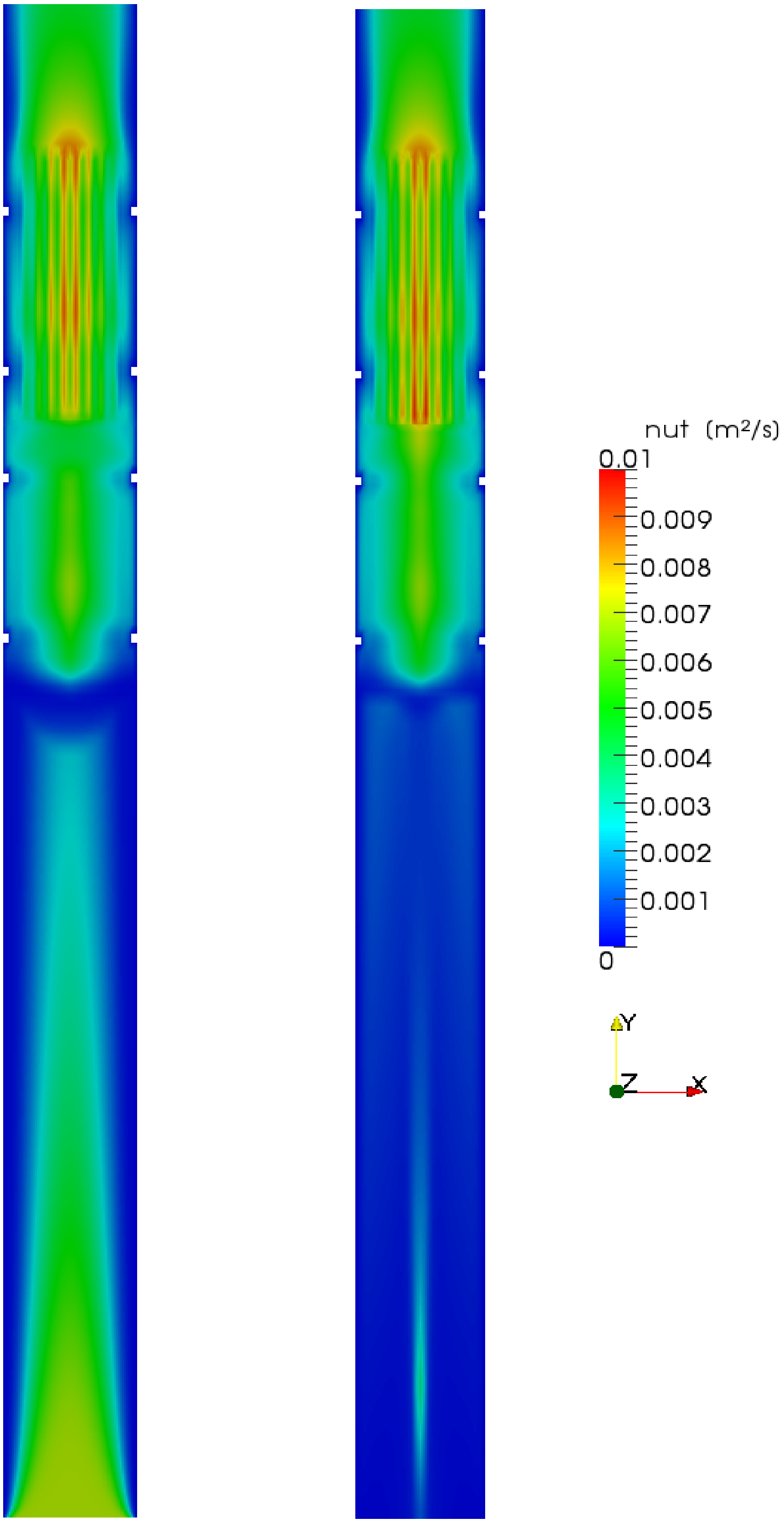


Figure 41: *Cases (a) and (b). Specific dissipation of the turbulence kinetic energy* ω (1/s) *at* $z = 0$.



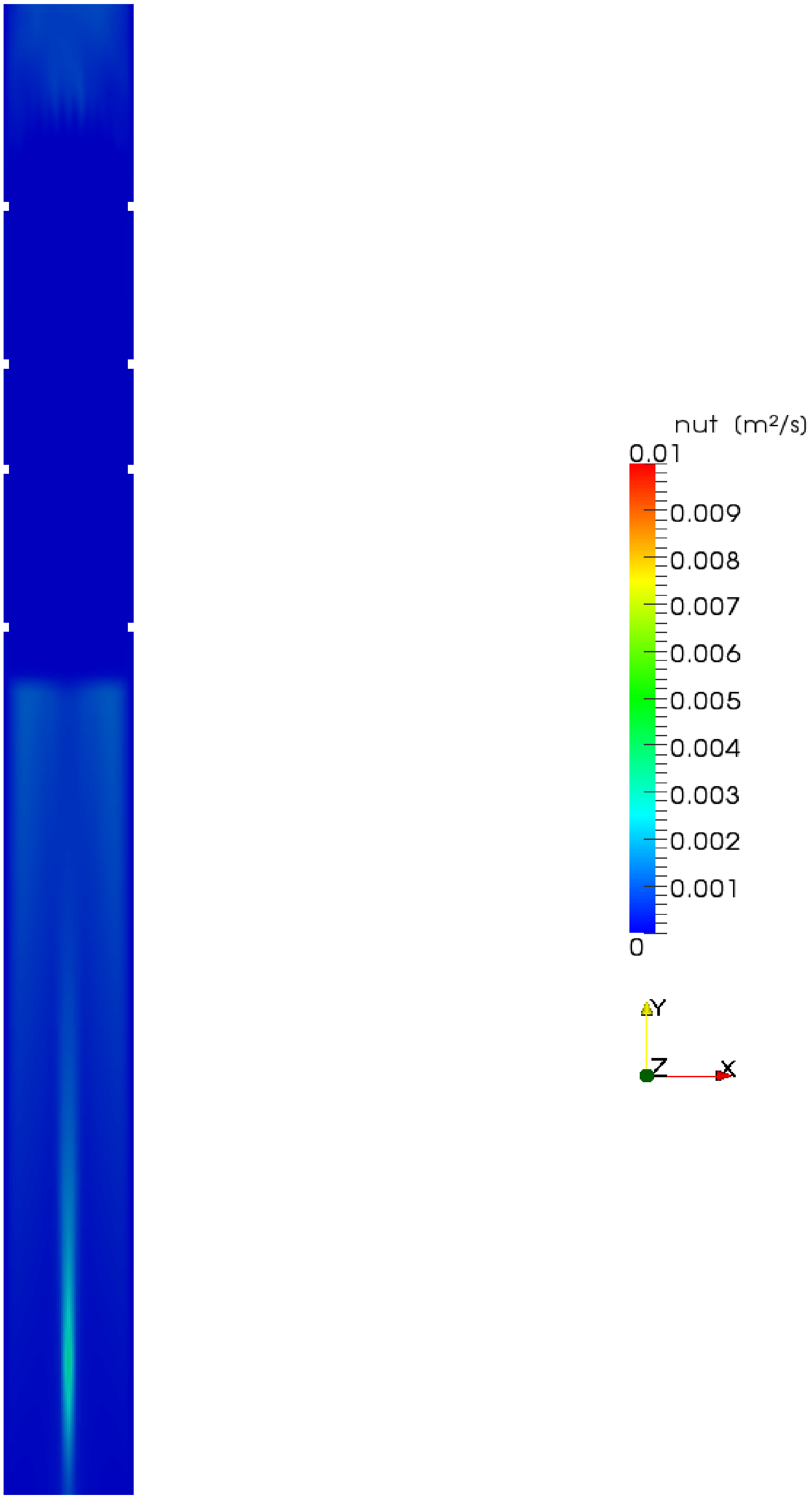
(*c*)

Figure 42: *Case (c). Specific dissipation of the turbulence kinetic energy* ω (1/s) *at* $z = 0$.



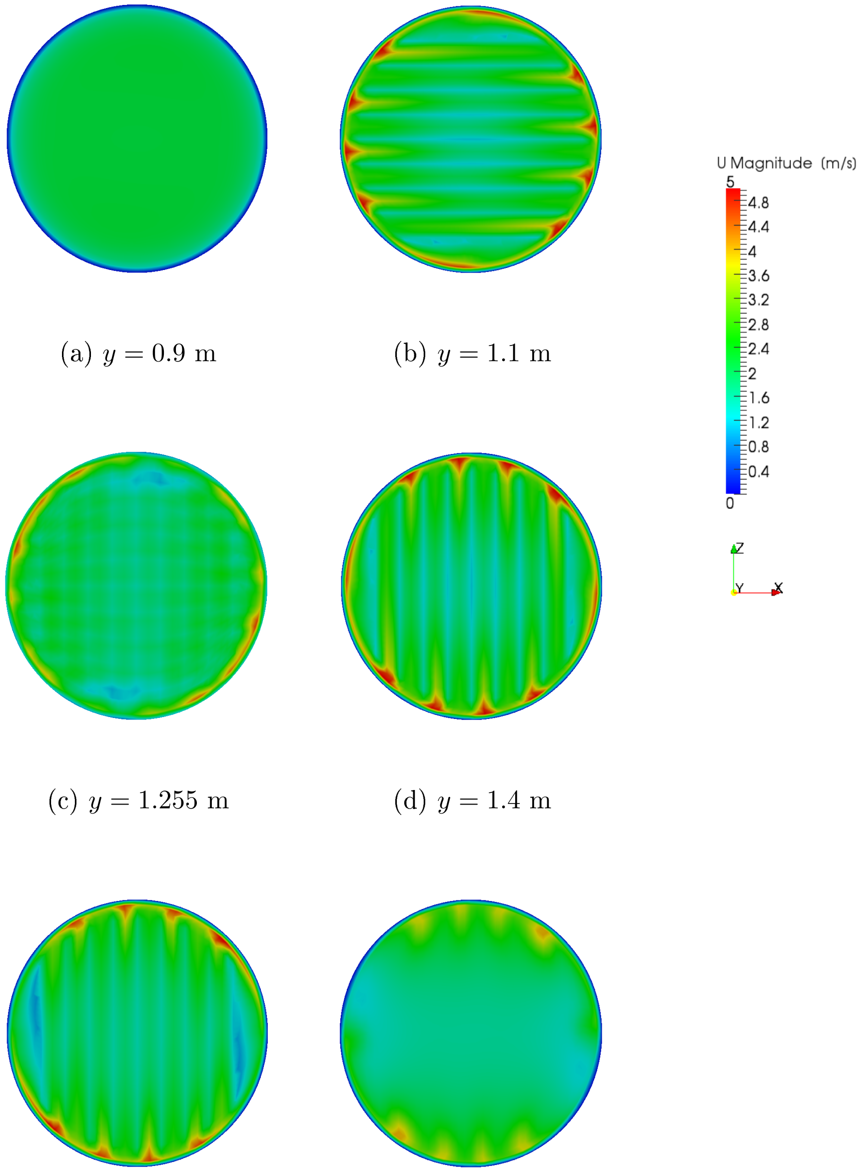
(a) (b)

Figure 43: *Cases (a) and (b). Turbulent kinematic viscosity* ν_t (m^2/s) *at* $z = 0$.



(*c*)

Figure 44: *Case (c). Turbulent kinematic viscosity* ν_t (m^2/s) *at* $z = 0$.

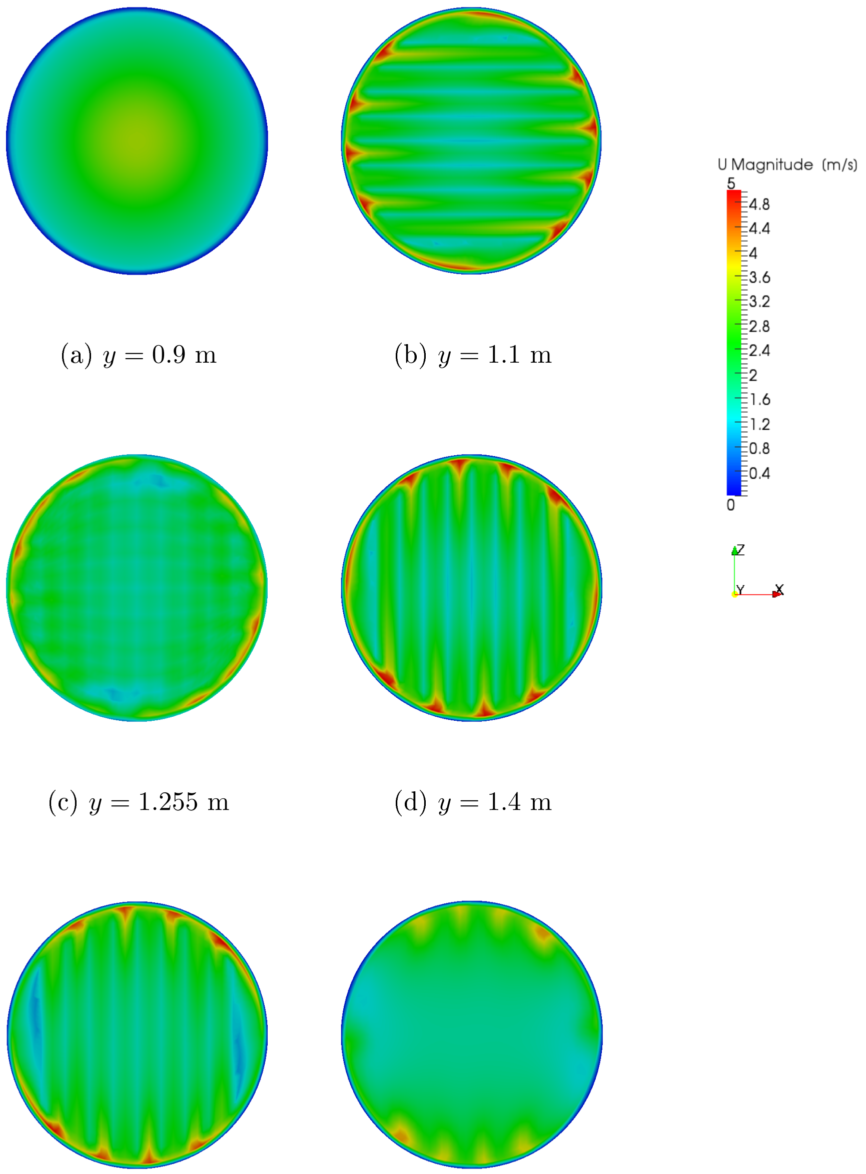


(a) $y = 0.9$ m (b) $y = 1.1$ m

(c) $y = 1.255$ m (d) $y = 1.4$ m

(e) $y = 1.56$ m (f) $y = 1.6$ m

Figure 45: *Case (a). Velocity magnitude* (m/s) *in case of laminar flow and plug inlet flow in various cross sections. The SST $k-\omega$ turbulence model is applied.*



(a) $y = 0.9$ m

(b) $y = 1.1$ m

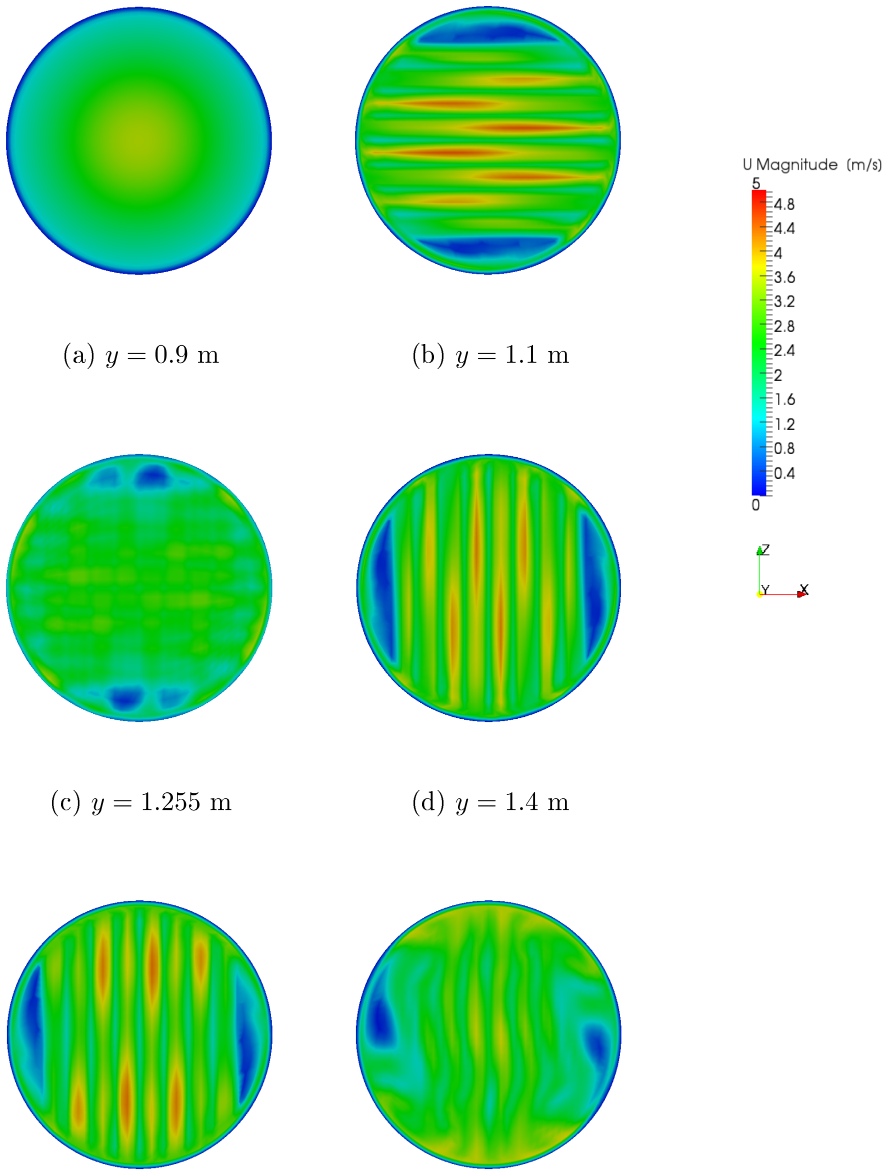
(c) $y = 1.255$ m

(d) $y = 1.4$ m

(e) $y = 1.56$ m

(f) $y = 1.6$ m

Figure 46: *Case (b). Velocity magnitude* (m/s) *in case of turbulent flow and fully developed inlet flow in various cross sections. The SST* $k - \omega$ *turbulence model is applied.*



(a) $y = 0.9$ m

(b) $y = 1.1$ m

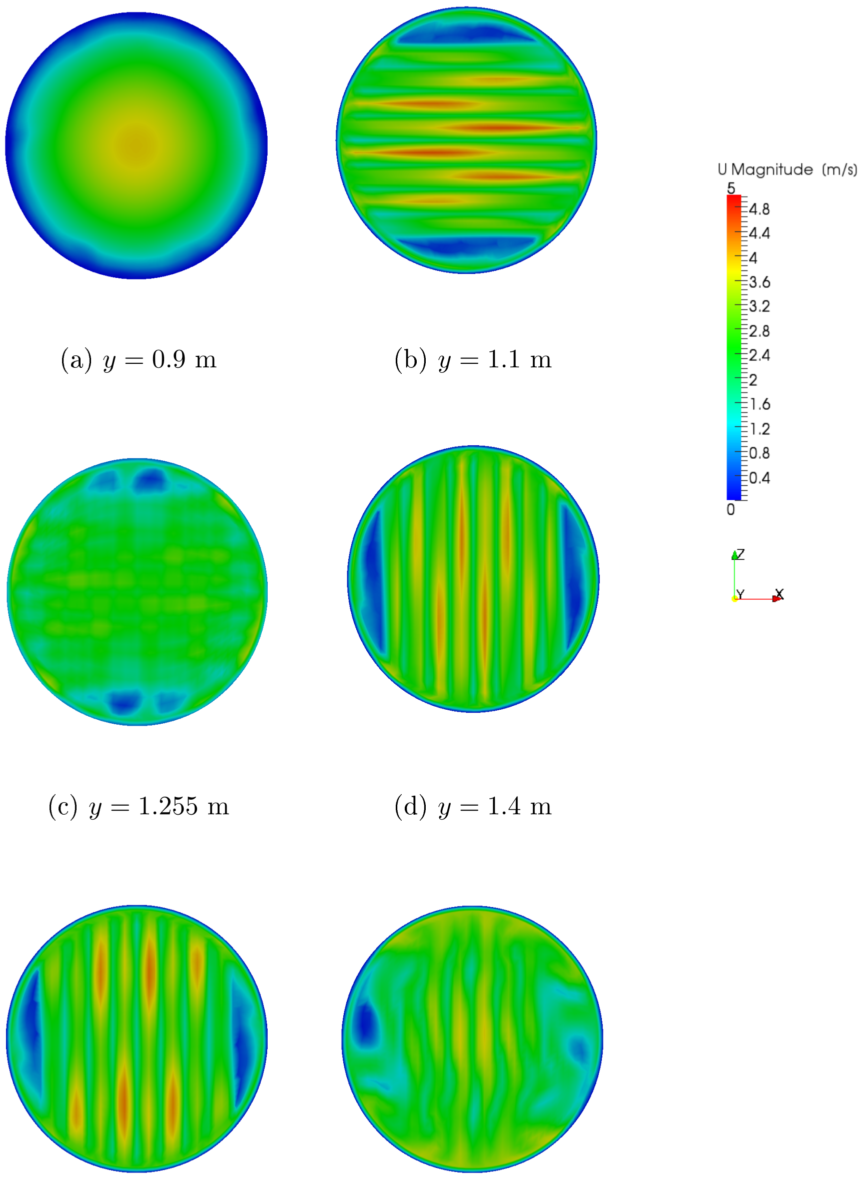
(c) $y = 1.255$ m

(d) $y = 1.4$ m

(e) $y = 1.56$ m

(f) $y = 1.6$ m

Figure 47: *Case (c). Velocity magnitude* (m/s) *in case of turbulent flow and fully developed inlet flow in various cross sections. The modified SST $k - \omega$ turbulence model applied.*



(a) $y = 0.9$ m

(b) $y = 1.1$ m

(c) $y = 1.255$ m

(d) $y = 1.4$ m

(e) $y = 1.56$ m

(f) $y = 1.6$ m

Figure 48: *Case (d). Velocity magnitude* (m/s) *in case of laminar flow and fully developed inlet flow in various cross sections.*

Pressure

In Fig. 49 it can be seen that the turbulent cases (a) and (b) have much higher a pressure loss as compared to the similar cases, where the whole calculation is laminar (c) and where the modified SST $k - \omega$ model is used (d). Because of the anisotropic porous model presenting the structured packing, the flow turns towards a lower resistance in the porous media. Therefore, there are velocity gradients in the packing, thus the turbulence production term increases. Therefore, turbulence kinematic viscosity ν_t increases and this increases the pressure losses in cases where the bed is modelled turbulent. The results presented here suggest, that the applied SST $k-\omega$ turbulence model cannot predict the turbulence in the packed bed realistically in its basic formulation. Therefore, the modified turbulence model was created in OpenFOAM, where the porous media representing the packed bed is assumed laminar zone. The modification was to set turbulent kinematic viscosity recently in $\nu_t = 0$ in the porous media, while the turbulence affects on the flow through this quantity. Case (c) in Figs. 32 – 44 represents the results computed with the modified turbulence model.

The simulation results of the cases (c) (laminar) and (d) (modified turbulent) are the most reliable. The packed bed causes the most of the pressure loss. Whether the free part of the column is modelled as turbulent (c) or as laminar (d) does not seem to have an effect on the pressure loss. The comparison between the pressure loss of these cases and the experimental results of Owens are presented in Fig. 50. The experimental values are obtained from Owens' dissertation [24]. The coefficient for the resistance of the channel was obtained by iteratively calculating the cases with different values of f_{e2} for the local coordinate direction $e2$. The best correlation is obtained with the value $f_{e2} = 3.5$ for the coefficient in Eqn. (70).

7.3.8 Conclusions

Based on the results presented in Chapt. 7.3 it is concluded that it is a good method to use the modified SST $k-\omega$ turbulence model for the rest of the work. The laminar calculations gave similar results for pressure loss across the bed. In case of modified SST $k - \omega$ model the bed is modelled laminar, so the turbulence model does not affect the flow field in the packing either. Thus in case of bed calculations, laminar model is also an appropriate choise.

The resulting coefficient for the inertial term f in (70) is $f = 3.5$ and this value is used in the rest of the work.

7.4 Gas maldistribution study: Case Olujic

In this chapter the ability of the created porous model for structured packed bed to predict gas maldistribution in the anisotropic beds is studied. Olujić made experiments on the gas maldistribution in a 1.4 m diameter column [27]. Initial maldistribution was obtained with the help of a plate below the bed. The experiment by Olujić provide experimental data for this work. Simulations for a similar column

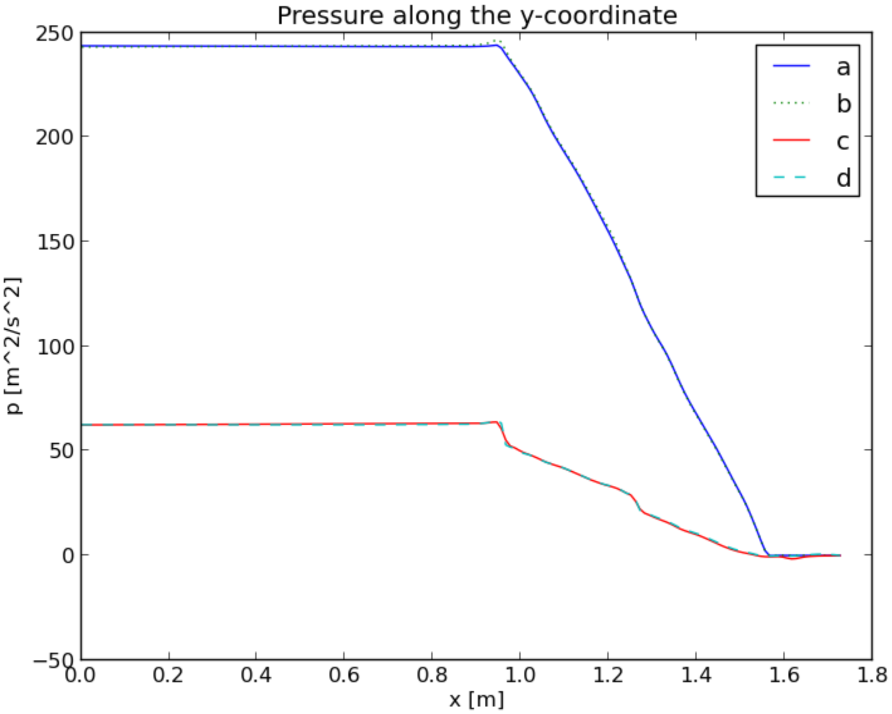


Figure 49: *Pressure profile along the column height. (a): Turbulent flow, a bulk flow at the inlet. (b): Turbulent flow and a fully developed inlet flow. (c): Turbulent flow and a fully developed inlet flow. Modified turbulence model. (d): Laminar flow, a fully developed inlet flow. Mean inlet velocity* $u_m = 2$ m/s.

as used in the experiments of Olujić are here conducted and simulation results are compared to experimental results.

Previously (Chapt. 7.3) the porosity model for the packed bed was tuned to produce the measured pressure losses. Here two of the test cases by Olujić are simulated to validate the ability of the current model to predict velocity distributions in a structured packed bed. A laminar calculation for the packed bed is applied for the two test cases according to the experience from the calculations in Chapt. 7.3. The laminar calculation of the bed corresponds to the modified SST $k-\omega$ turbulence model in the bed region, since there the bed is modelled laminar, too.

Two cases for studying the gas distribution produced by the model obtained in Chapt. 7.3 are selected from Ref. [27]. In the cases the inlet flow is partly blocked, once the left part of the inlet is blocked and once a circular area in the middle of the inlet is blocked. These cases are referred to as the chordal blockage and as the central blockage, respectively. The blocked area is half of the cross-section area in each case. The velocity field in the bed cross-section at five different heights was measured [27]. Both cases, the chordal blockage and the central blockage, were simulated with the anisotropic and isotropic porosity model and results are compared to experimental results. The anisotropic model is the model described in Chapt. 7.3

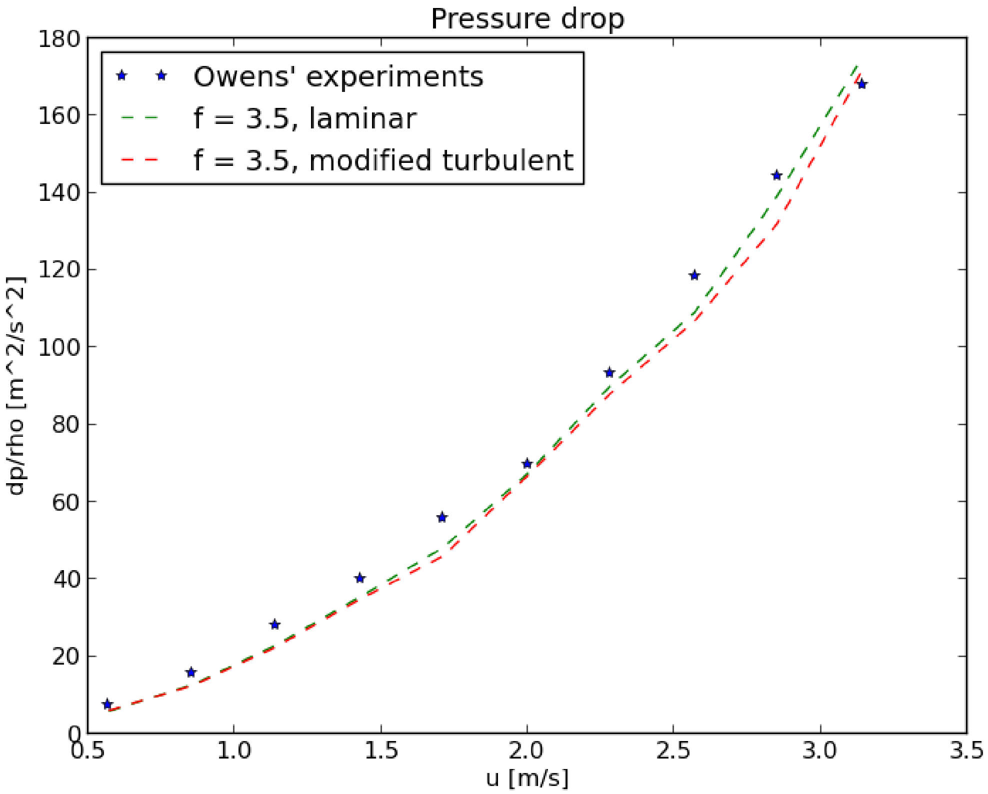


Figure 50: *Pressure drop as a function of the mean inlet velocity. Coefficient f is the inertial term coefficient in equation* (70)*. The results of the Owens experiments are compared to two simulation cases, modified turbulent (c) and laminar (d).*

and the isotropic model refers to the model often used in packed bed modelling [53]. The coefficients for the isotropic porosity model were set so that the same pressure loss across the bed resulted as in the calculations with anisotropic porosity model. As explained in Chapt. 7.3, in the anisotropic porosity model the factor $f_{e2} = 3.5$ and $f_{e1} = f_{e3} = 1000 \times f_{e2}$. In the isotropic case all the factors are identical and the corresponding pressure loss is obtained with the values $f_{e1} = f_{e2} = f_{e3} = 90$.

7.4.1 Grid generation

The packing height in this case was 200 mm and the bed consisted of 5 packing layers. Grid for the case was generated with `blockMesh` and `snappyHexMesh` mesh generation utilities [49]. First, a simple rectangular grid was generated with `blockMesh` with a uniform cell size of 10 mm. The porous zones are defined in the grid with the aid of a Python script, which adds cells to different porous zone groups according to the location of the cell. The Python script generates a `.setSet`-file, which is an utility in OpenFOAM to manipulate cell-, face- or point sets or zones interactively [57]. This utility finally produces the porous zones when the `.setSet`-file is run in

OpenFOAM. The porous zones represent the corrugated packing sheets with flow channels, like in case of Chapt. 7.3. Examples of the input files for `blockMesh` and `snappyHexMesh` utilities are presented in Appendix F.

After generating the porous zones, STL-surfaces which define the cylinder were made with FreeCad software and the final grid was obtained with `snappyHexmesh`. In the final grid there are $230,4754$ cells. The chordal blockage and central blockage at the inlet were defined with the `topoSet` utility of the OpenFOAM. In Fig. 52 every second porous zone in the x-direction is presented. The porous zone layers with channels in the z-direction are between the three shown layers with channels in x-direction.

The reason why the grids are generated in a different way in Chapts. 7.3 and 7.4 is that in Chapt. 7.3 the grid is small and porous zones are easily defined by hand in Gambit. In Chapt. 7.4 the grid is significantly larger and the amount of the porous zones is so high that definition of the porous zones is not reasonable by hand thus the porous zones are defined with the Python code.

7.4.2 Boundary conditions

In Fig. 51 a schematic figure of the column used in simulation of the Olujić case presented in Chapt. 7.4 from Ref. [27]. The gas capacity factor in Ref. [27] is determined to be $F_S = 2$ Pa$^{1/2}$. According to Eqn. (5) and with constant density $\rho = 1.2$ kg/m^3, the mean velocity at the inlet of the column is

$$u_{col} = \frac{F_S}{\sqrt{\rho}} \approx 1.83 \text{ m/s} \tag{100}$$

and since the chordal blockage and central blockage plates block half of the cross-section area, according to the continuity the inlet velocity to the bed is approximately two times the inlet velocity to the column

$$u_{bed} = \frac{A_{col}}{A_{bed}} u_{col} = \frac{1}{2} u_{col} \approx 3.7 \text{ m/s} \tag{101}$$

It is shown that the gas flows in from the bottom of the column at a velocity of $u = 3.7$ m/s and flows out from the top of the column. At the outlet the gauge pressure is $p = 0$ Pa. The kinematic viscosity of the gas is $\nu = 1.56 \cdot 10^{-5}$ m^2/s.

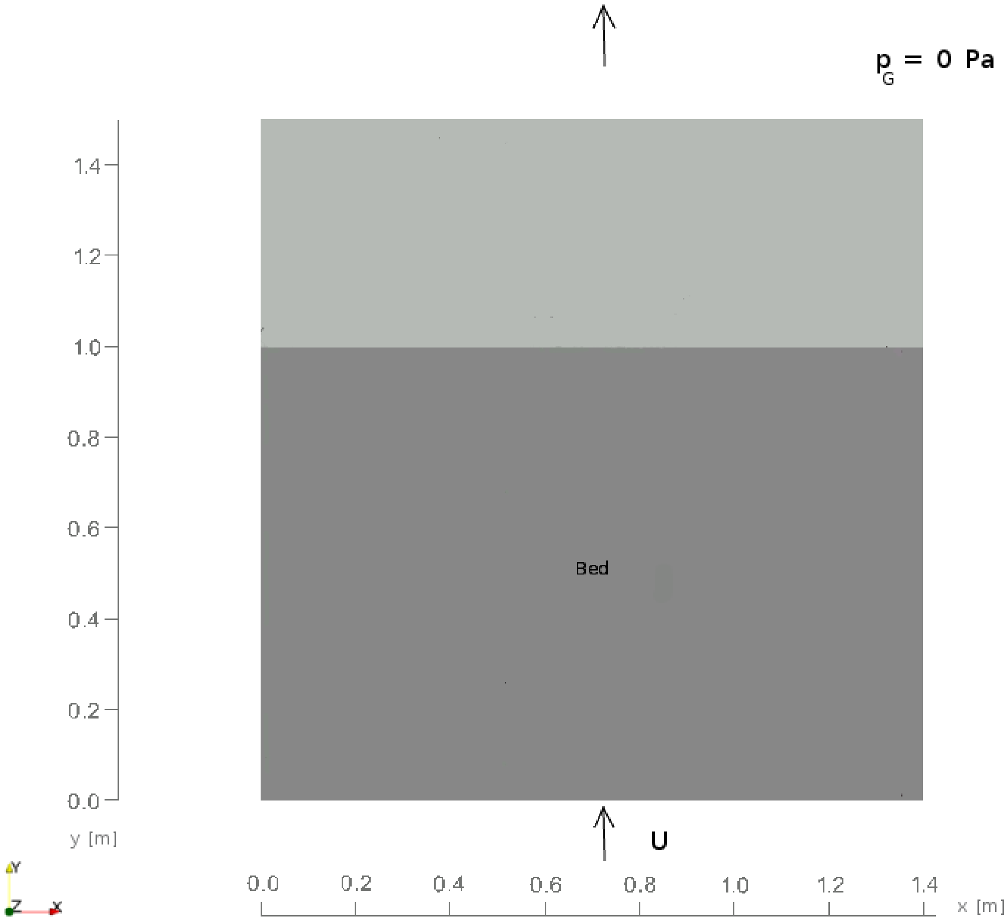


Figure 51: *Schematic figure of the column used in a simulation which is compared to Olujic experimental results. The scale is in meters* (m) *The darker grey area* $y = 0.0 - 1.0$ m *is the packed bed containing five layers of packings, each rotated* 90° *with respect to the lower layer. A light grey area* $y = 1.0 - 1.5$ m *presents the empty space in the column after the packed bed.*

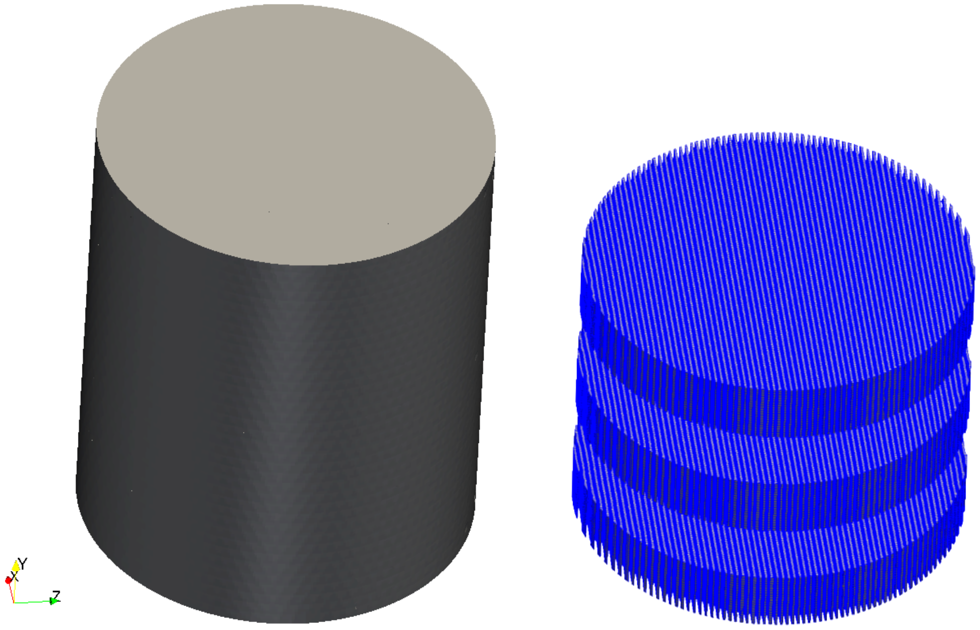


Figure 52: *The packed bed in column (left) presented schematically in Fig. 51 consists of five layers of which three are presented (right). Every second porous zone in the x-direction in three beds are presented.*

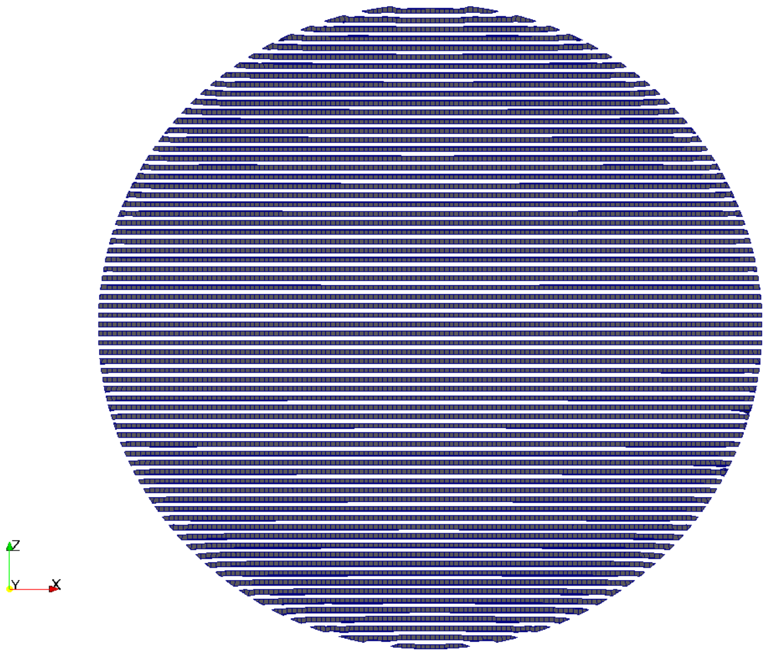


Figure 53: *Every second porous zone in the x-direction from viewpoint above the column.*

7.4.3 Results

The results for both the chordal blockage and the central blockage cases are presented in this chapter. First, the chordal blockage case is studied. The convergence history of the case can be seen in Fig. 54. It is noticed that the calculation is well converged. In Fig. 55 the measured data and two simulation results, one with the anisotropic and one with the isotropic porosity model, are presented. When the anisotropic (b) and isotropic (c) model results are compared to the measured results (a) in Fig. 55 it is seen that the anisotropic model corresponds better to the experimental results. According to the measurements the initial maldistribution created with the chordal blockage does not smooth out in the packed bed of height of approximately 1 m and this behaviour is also seen in the anisotropic simulation results. The simulation results with the isotropic porosity model predict that the gas maldistribution would smooth out after approximately 0.4 m which is not true according to the experiments.

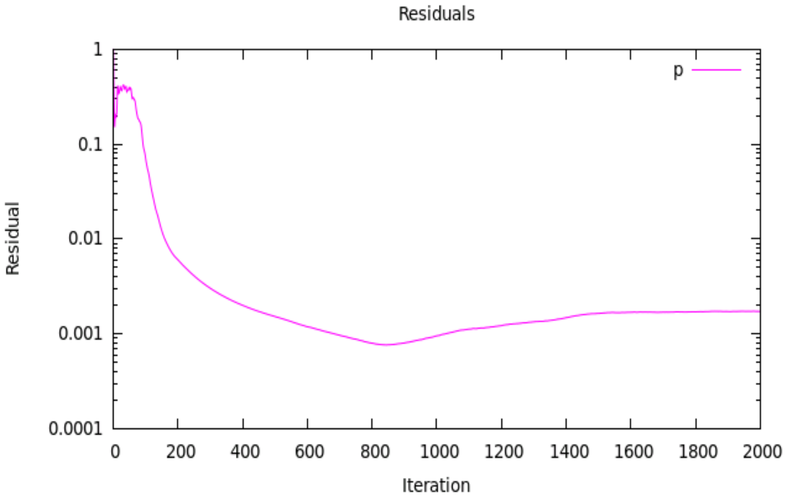


Figure 54: *Convergence history in a case of a chordal blockage.*

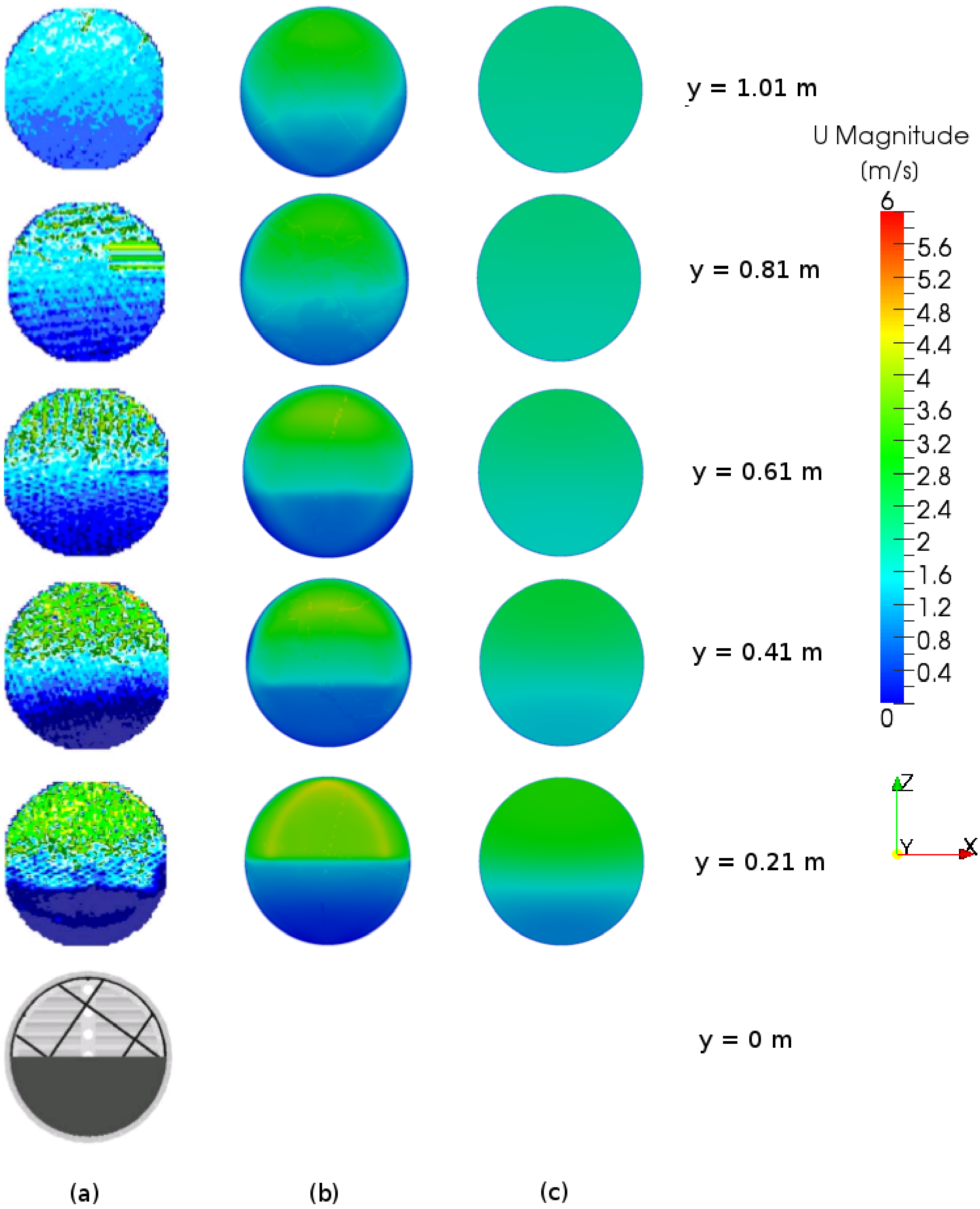


Figure 55: *Velocity magnitudes at different cross sections in case of the chordal blockage. The inlet velocity is* $u_{in} = 3.7$ m/s. *(a): Measured velocities from Ref. [27], (b): Simulation results in case of the anisotropic porosity model, (c): Simulation results in case of the isotropic porosity model.*

Second, the central blockage case is studied. The convergence history of the case can be seen in Fig. 56. It is noticed that the calculation is well converged. In Fig. 57 the measurement results and two simulation results, one with the anisotropic and one with the isotropic porosity model, are presented.

When the anisotropic (b) and isotropic (c) model results are compared to measured data (a) in Fig. 57 it is seen that the anisotropic model corresponds better to the experimental results. According to the measurements the initial maldistribution created with the central blockage smooths out in the packed bed of a height of approximately 0.61 m and this behaviour is also seen in the anisotropic simulation results. The simulation results with the isotropic porosity model predict that the gas maldistribution would smooth out after approximately 0.21 m which is too rapid smoothing according to the experimental results.

According to the both simulated cases, the chordal blockage and the central blockage, anisotropic model seems to model better the gas flow in a packed bed than the isotropic model.

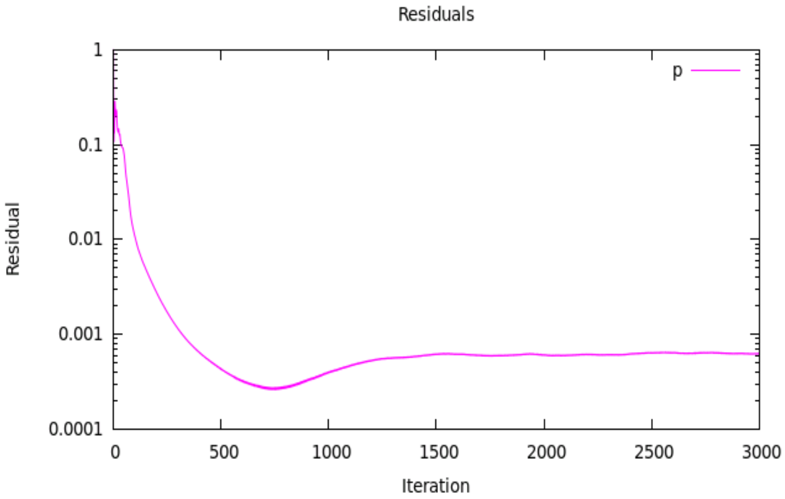


Figure 56: *Convergence history in a case of a centre blockage.*

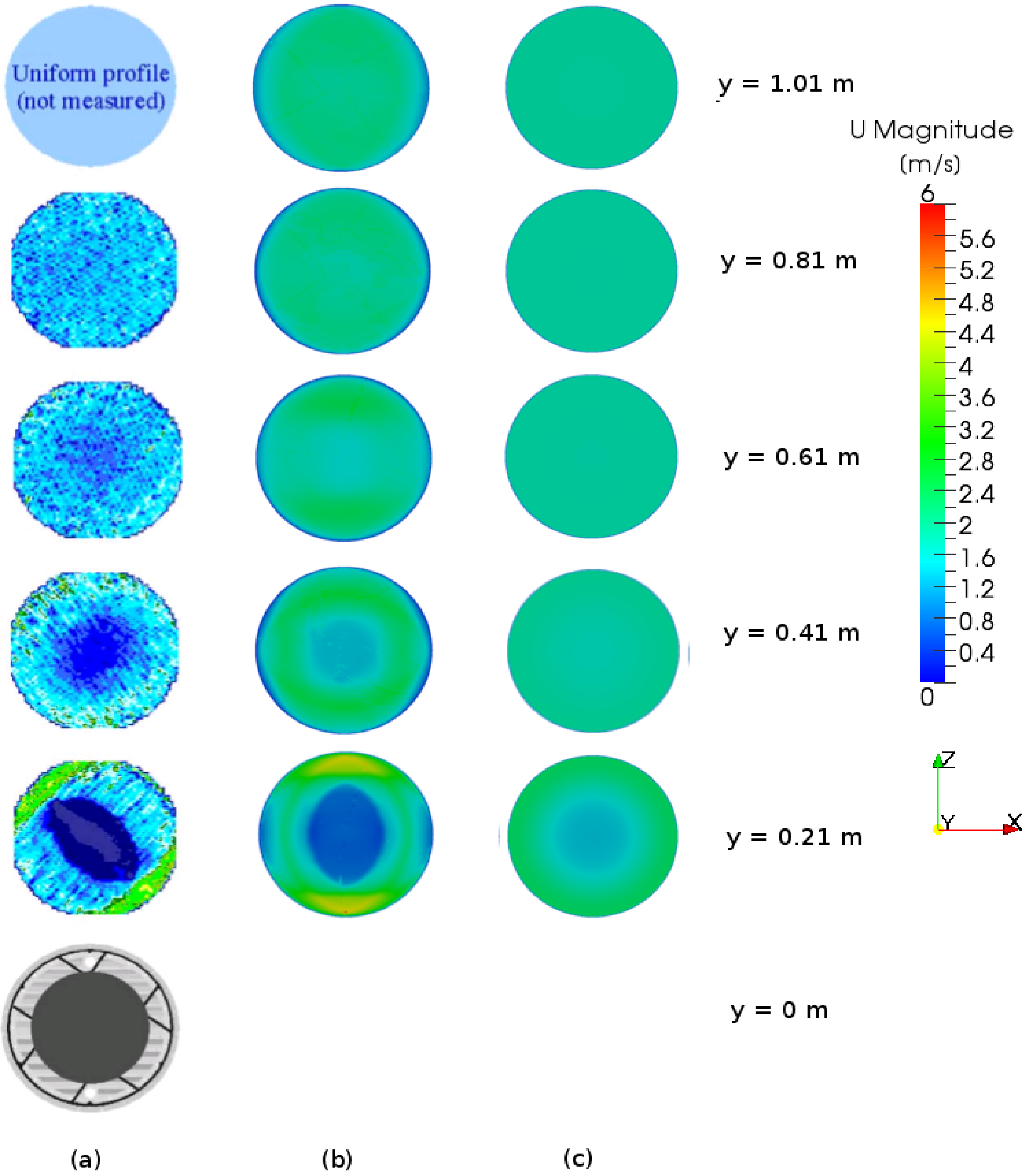


Figure 57: *Velocity magnitudes at different cross sections in case of the central blockage. The inlet velocity* $u_{in} = 3.7$ m/s. *(a): Measured velocities from Ref. [27], (b): Simulation results in case of the anisotropic porosity model, (c): Simulation results in case of the isotropic porosity model.*

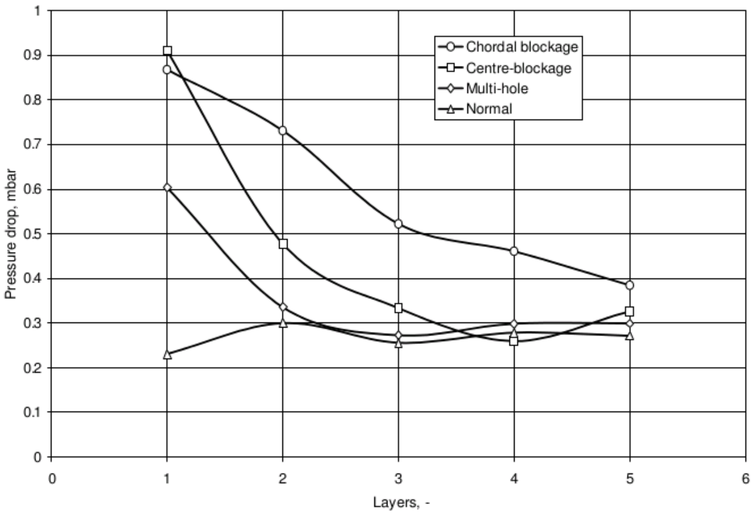


Figure 58: *Pressure drop per packing layer for the packed bed in case of different maldistribution forms at a gas capacity factor of* $F_S = 2$ $\mathrm{Pa}^{1/2}$ [27].

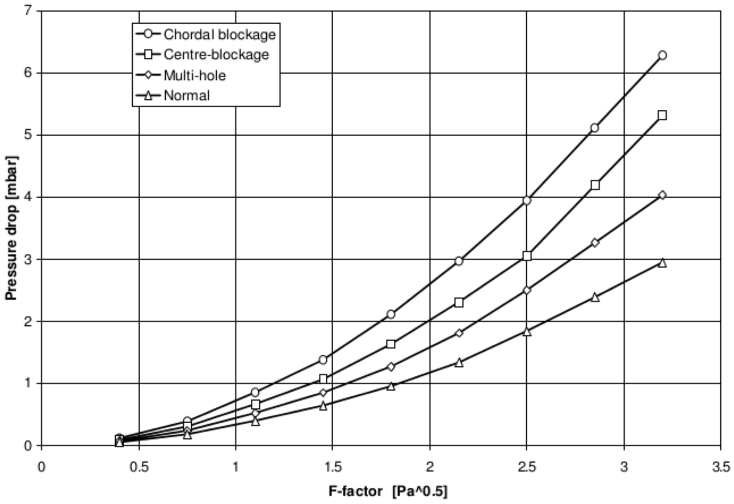


Figure 59: *Pressure drop across the bed in case of different maldistribution forms as a function gas capacity factor* F_S [27].

Pressure losses across the bed according to the Olujić experiments are presented in Figs. 58 and 59. The total pressure drop across the packed bed in experiments and simulations is shown in Tab. 6.

The differences between the pressure drop values may occur because the packing in the experiments of Olujić [27] was different than in experiment of Owens [24], against which the porosity model is tuned. The packing used in the Olujić experiments is Montz-pak B1-250 and in the experiments of Owens the packing is Mellapak 250Y. The packings are very similar and have same geometrical characteristics, except that Mellapak 250Y is made of perforated sheets of metal and Montz-pak

Table 6: *Pressure drop results across the packed bed from experiments [27] and simulations.*

p [Pa]	Chordal blockage	Central blockage
Experiments	380	320
Simulation	260	200

B1-250 of non-perforated sheets. Therefore, the pressure drop is higher in case of the experiments, since the non-perforated sheets do not let gas flow through the packing walls, thus the flow path is more tortuous for the gas in Montz-pak B1-250. In real columns having a two-phase flow and hence irrigated packing, most of the holes in the sheets of Mellapak 250Y would be filled with liquid and the geometry would correspond to the geometry of the Montz-pak B1-250. However, the experiments of Owens [24] were conducted only with gas and, therefore, there is a difference between the pressure losses across the bed.

In Ref. [27] the coefficient of variation CV (Eqn. (13)) values are calculated in every layer to as a measure of the maldistribution. In Tab. 7 the values of the coefficient of variation from experiments [27] and from the simulations are presented. It is seen that the values in simulations with the anisotropic porous media are in the correspond almost the values in experiments. The flow in column with chordal blockage smooths out slower than in the case with the central blockage, which was also seen in Figs. 55 and 57. The results from the simulations with an isotropic porous media differ significantly from the values in anisotropic porous media. Thus the maldistribution is not captured with an isotropic porosity, but anisotropic porosity gives corresponding results to the experiments.

Table 7: *The coefficient of variation (CV) in different layers in bed from experiments [27] and simulations.*

y [m]		0.0	0.21	0.41	0.61	0.81	1.01
Experiments	Chordal blockage	1.28	0.82	0.68	0.53	0.44	0.24
	Central blockage	1.10	0.64	0.38	0.34	0.32	–
Simulations	Chordal blockage	0.93	0.77	0.61	0.58	0.46	0.46
	Central blockage	0.86	0.63	0.29	0.23	0.18	0.21
	Chordal blockage, isotr.	0.47	0.38	0.23	0.12	0.07	0.05
	Central blockage, isotr.	0.47	0.18	0.04	0.007	0.003	0.02

7.5 Industrial scale case: Maldistributed inlet velocity profile

The anisotropic porosity model for the Mellapak 250Y packed bed is calibrated in Chapt. 7.3 with respect to pressure losses. In Chapt. 7.4 the model is validated from the velocity distribution point of view. This model is next applied on an industrial scale column.

The CO_2 absorption columns have often a large diameter and, therefore, the flow field is not easily predictable. The aim is to study gas flow distribution in a large scale industrial column with the model which was formed and presented in the previous Chapts. 7.3 and 7.4.

7.5.1 Grid generation

The computational resources available for the current work allow computations for approximately 100 million cells within a reasonable computational time. With this maximum number of cells, for example a grid for $10\times2\times2$ m with cells of $1\times2\times1$ cm could be used. In the packing one channel between corrugated metal sheets is approximately 1 cm wide. These channels are modelled as a porous zone, thus each porous zone has a width of 1 cm which poses an upper limit for cells in the x- and z-directions.

The grid of a size $10 \times 8 \times 2$ m with periodic boundary conditions could present the rectangular industrial scale column. The cross sections of absorption columns in practice can be either round or rectangular. The grid which is proposed corresponds to a column with a rectangular cross section.

The grid is generated in the same way as in Chapt. 7.4, by first creating the grid with the `blockMesh`-utility, then selecting the porous zones with the Python script to `.setSet`-file and finally creating porous zones by running the `.setSet`-file in OpenFOAM. The `snappyHexMesh`-utility used in Chapt. 7.4 is now not needed since the geometry is simply a rectangle. The packing height in this case was 300 mm and the bed consisted of 27 packing layers.

7.5.2 Boundary conditions

In Fig. 60 a schematic figure of the column used in simulation of an industrial case is presented. It is shown that the gas flows in from the bottom of the bed and flows out from the top of the column. At the outlet the gauge pressure is $p = 0$ Pa. The kinematic viscosity of the gas is $\nu = 1.5 \cdot 10^{-5}$ m^2/s. At the inlet a non-even inlet velocity profile is applied to find out whether maldistribution smooths out in the packed bed or not. The inlet velocity profile is defined as a piecewise continuous function

$$v = \begin{cases} 0, & \text{if } 0 \text{ m} < x < 4 \text{ m} \\ 2x - 8, & \text{if } 4 \text{ m} \leq x \leq 6 \text{ m} \\ 4, & \text{if } 6 \text{ m} < x < 10 \text{ m} \end{cases} \tag{102}$$

The inlet velocity profile is shown also in Fig. 62 by the blue solid line. The walls in xy-plane have periodic boundary conditions.

7.5.3 Results

The convergence history of the case is plotted in Fig. 61. In Fig. 62 velocity profiles at different heights in the column of a total height of 8 m are presented. The mean

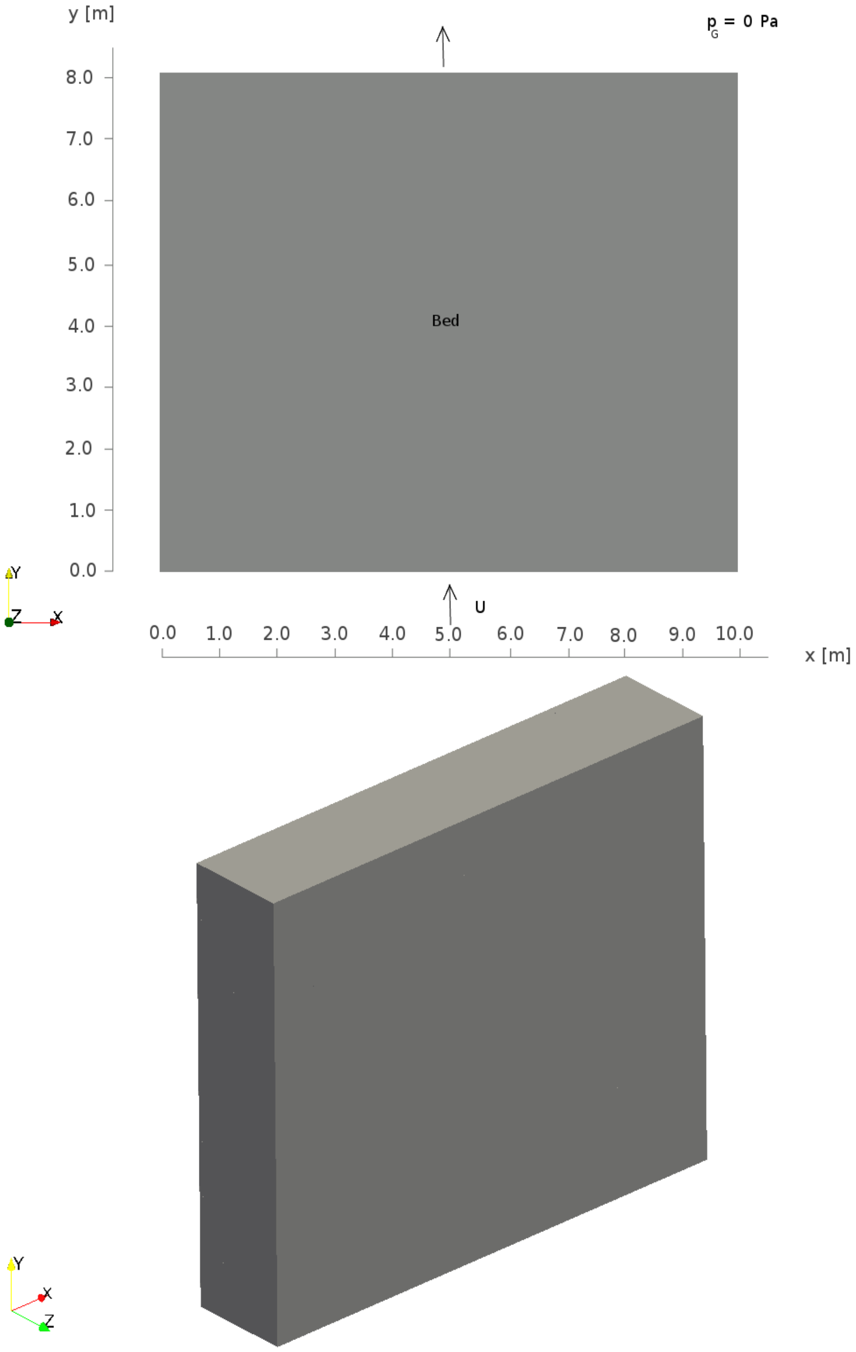


Figure 60: *A schematic figure of the industrial scale bed (upper figure) and a slightly rotated figure of the bed (lower figure).*

velocity in the y-direction is 2 m/s according to the typical operation conditions in a CO_2 absorption column. In this case the velocity profile at the inlet ($y = 0$ m) is assumed to be biased on the right side of the column so that on the left side velocity in y-direction is 0 m/s and on the right side the velocity in y-direction is 4 m/s, as is presented in Fig. 62. This presents a severe maldistribution in a column, caused for example by a bad inlet design. It is noticed that the maldistribution in a gas flow starts to smooth out along the packed bed height, but a significant maldistribution remains still after the height of $y = 8$ m: the range of the velocity magnitudes is from 1 m/s to 3 m/s.

In Fig. 63 the velocity magnitude in the packed bed cross-section $z = 1.0$ m, i.e in the middle of the computational domain in the z-direction, is presented with velocity vectors. The length of the velocity vectors is scaled according to the velocity magnitude. The vectors are plotted in the xy-plane, so only other flow direction of the channels in the xy-plane is shown. The bed layers are rotated 90° with respect to lower bed layer, so that every second bed layer is oriented in same direction. This can be clearly seen as stripes in Figs. 63 – 66 presenting the velocity fields of velocity magnitude and different velocity components. It is seen that the velocity maldistribution does not smooth out completely in the bed.

Fig. 64 presents the velocity in the x-direction in the bed cross-section. Near the inlet i.e. at the bottom of the bed there is a significant velocity component in the x-direction, but in this direction the velocity component starts to smooth out because the channels in the x-direction are perpendicular to the velocity profile gradient. Every second layer has its channels in x-direction and respectively, every second in the z-direction. Therefore, the velocity differences smooth out through convection along the channels in the x-direction.

In Fig. 65 it is noticed that some part of the gas flows downwards near the inlet on the left side of the column, where the inlet velocity is zero. The velocity component in the y-direction does not improve smoothing out because of its direction.

Fig. 66 presents the velocity field in the z-direction in the xy-cross-section. On the right side of the bed the packing channels in the z-direction can be seen as red and blue stripes while the gas flows in z-direction in those layers. In every second layer the packing channels are in the x-direction so there is no significant velocity component in the z-direction in those layers.

It is seen that the severe maldistribution in a gas flow in an industrial scale column does not seem to smooth out. The preferential flow directions may affect the flow so that the gas does not mix in the packing. On the other hand, this case was calculated as laminar, but the real flow in the packing would presumably contain turbulence to some extent. Turbulence increases diffusion so the gas flow differences could smooth out faster. However, the turbulence values in packing are not known, so the best and the most conservative guess in this situation is to assume laminar flow in the bed.

The pressure field in the bed is seen in Fig. 67. Pressure has its highest values were the flow velocity is highest, too. The average pressure at the inlet is approximately $p = 440$ Pa.

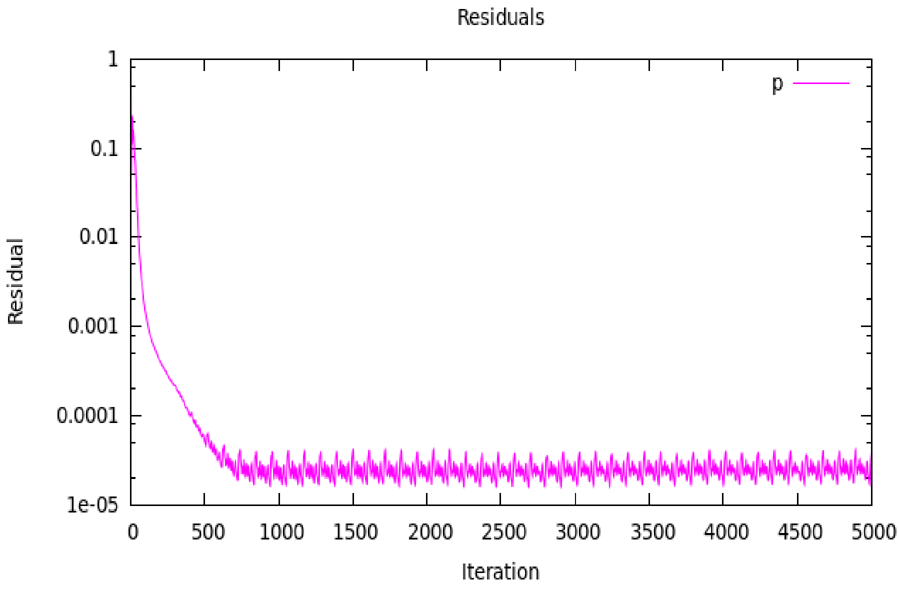


Figure 61: *Convergence history of the industrial scale case with the maldistributed inlet velocity profile.*

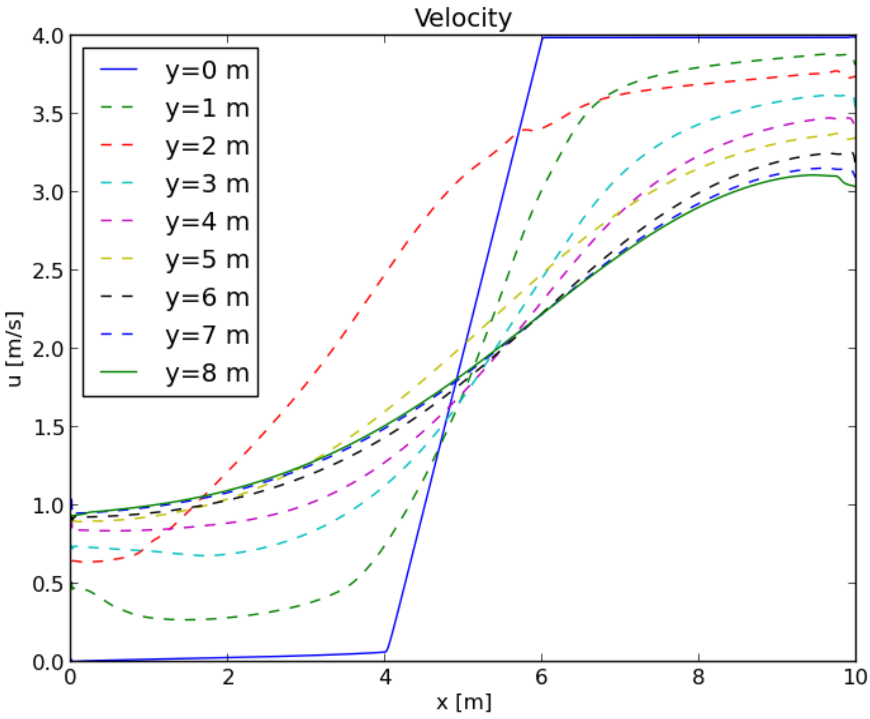


Figure 62: *Velocity profile* (m/s) *at different heights in a column of height of* 8 m.

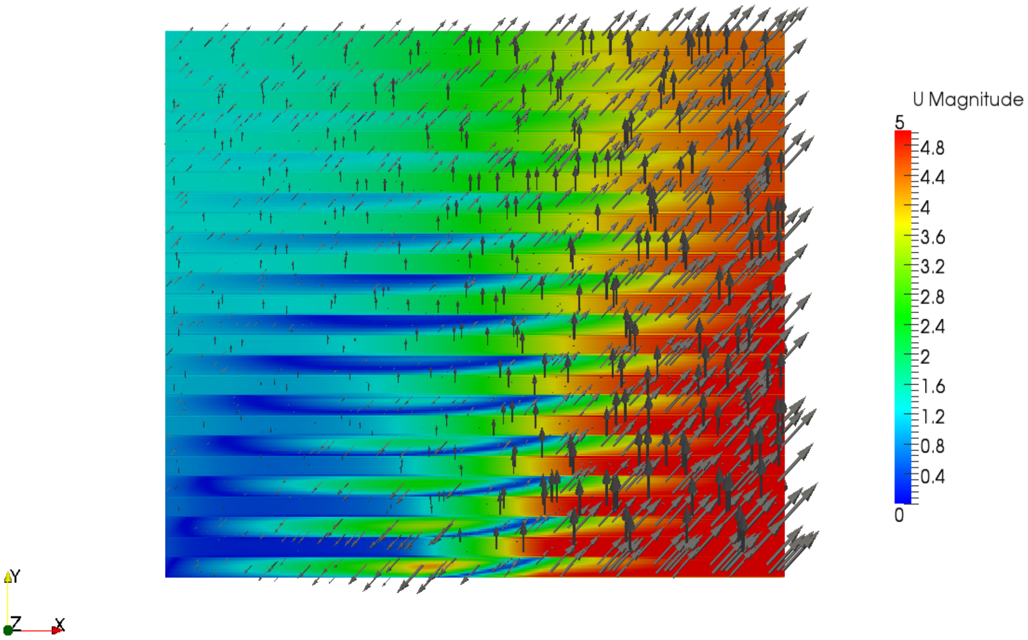


Figure 63: *Velocity magnitude* (m/s) *at the cross-section of* $z = 1.0$ m.

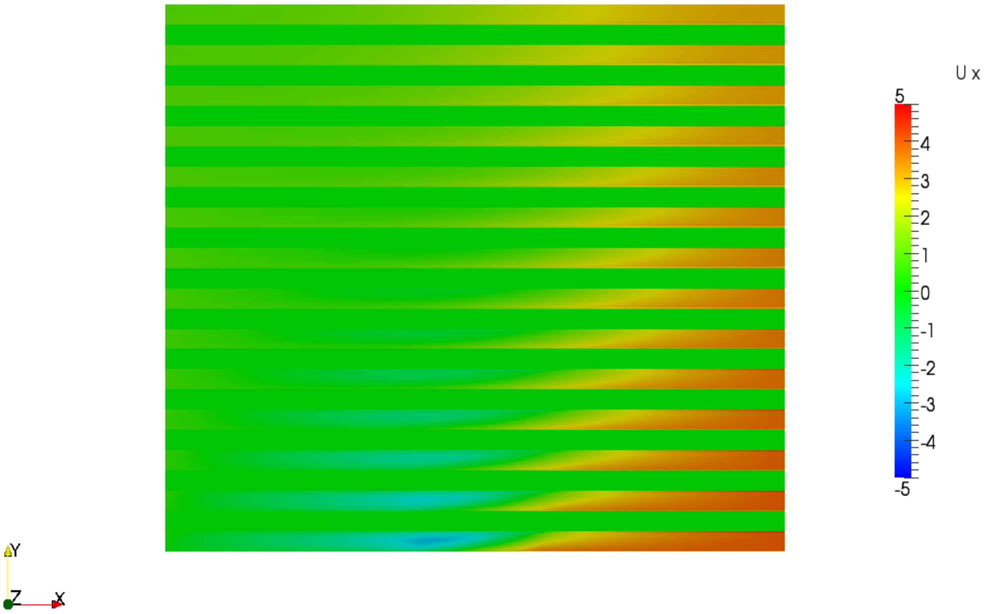


Figure 64: *Velocity in the* x*-direction* (m/s) *at the cross-section of* $z = 1.0$ m.

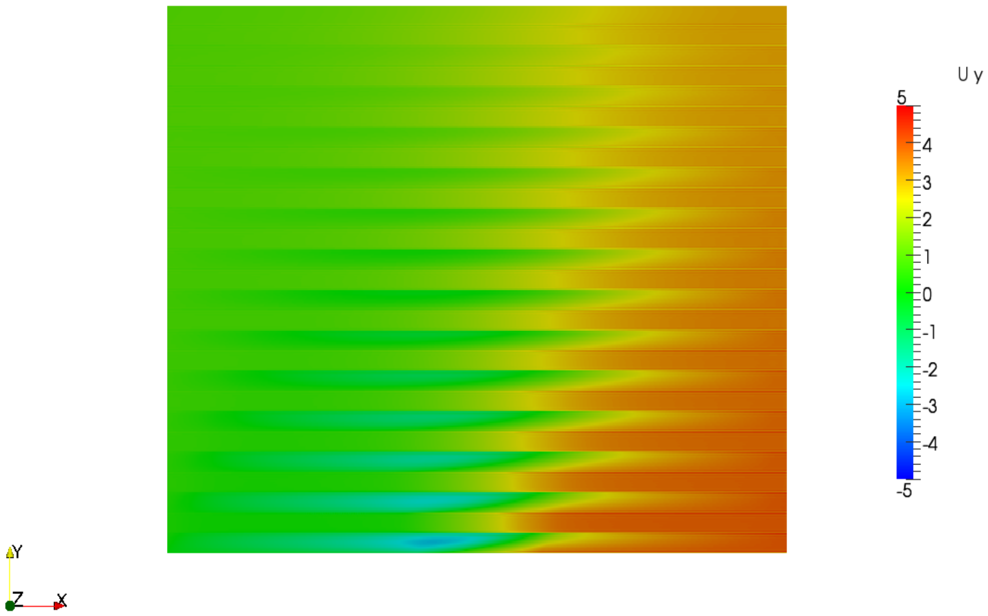


Figure 65: *Velocity in the y-direction* (m/s) *at the cross-section* $z = 1.0$ m.

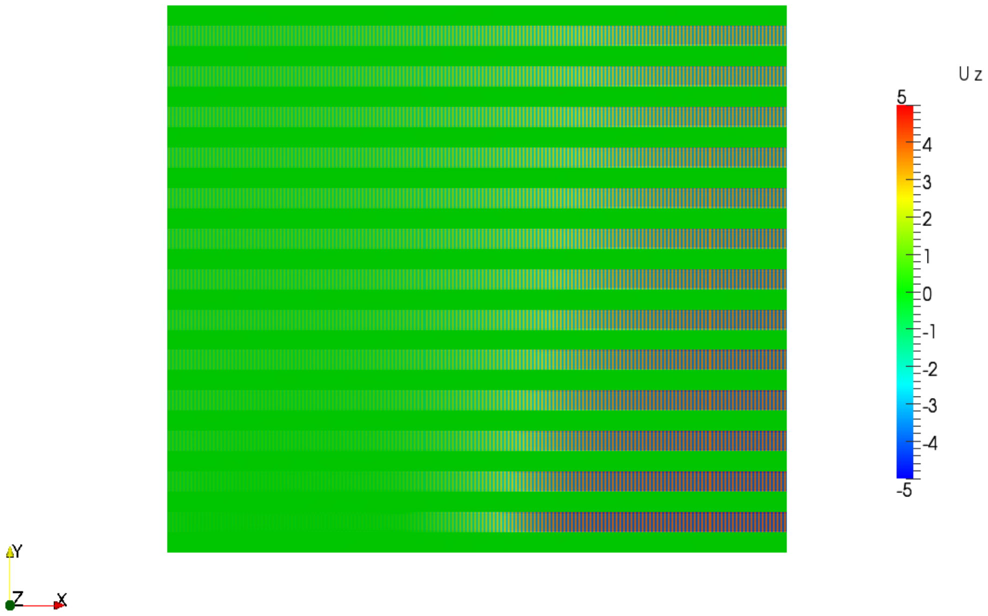


Figure 66: *Velocity in the z-direction* (m/s) *at the cross-section* $z = 1.0$ m.

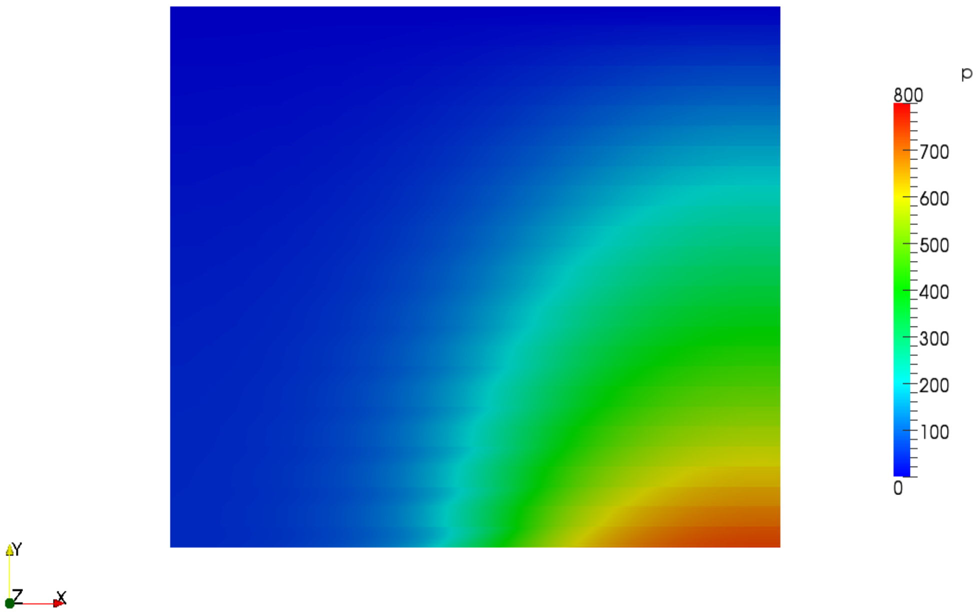


Figure 67: *Pressure field* (Pa/(kg/m^3)) *at the cross-section* $z = 1.0$ m.

7.6 Industrial scale case: Support beams at the inlet

In this chapter the grid is similar as in the case of Chapt. 7.5, except that five support beams for the packed bed at the inlet are taken into account. The same grid as in Chapt. 7.5 is used, but the support beams are formed with `createPatch`-utility in OpenFOAM [56]. The support beams are modelled as impermeable walls. The case was calculated in order to investigate, whether the support beams induce a maldistribution in the packed bed. The size of the support beams was estimated from the known absorber columns and the beams were assumed 200 mm wide.

7.6.1 Boundary conditions

The configuration of the inlet is shown in Fig. 68. The grey areas present the free area and the five white stripes are the support beams of the bed. The inlet velocity in the y-direction is $v = 2$ m/s. The kinematic viscosity of the gas is $\nu = 1.5 \cdot 10^{-5}$ m^2/s. The walls in the xy-plane have periodic boundary conditions.

Figure 68: *Schematic figure of the inlet of the column. Support beams are shown as white stripes at the inlet.*

7.6.2 Results

In Fig. 69 the convergence history of the computation is shown and it is seen that the computation is well converged. In Fig. 70 the velocity profiles at different heights of the column are presented and in Fig. 71 the velocity profiles are shown closer. At the inlet i.e. at height $y = 0$ m the locations of the support beams are clearly seen as zero velocity. The maldistribution due to the support beams smooths out after three packing heights, as is seen in Fig. 70 – 75 presenting the velocity fields at the cross-section $z = 1.0$ m i.e. in the middle of the column. According to these results the support beams do not induce significant maldistribution in the packed bed.

The pressure field in the bed is seen in Fig. 76. The pressure has its highest values were the flow velocity is highest, too. The average pressure at the inlet is approximately $p = 250$ Pa.

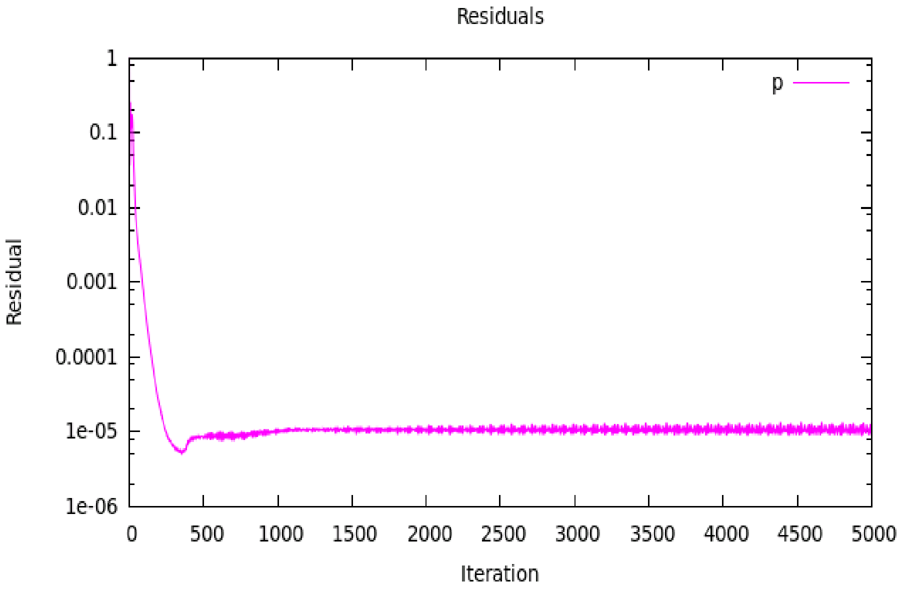


Figure 69: *Convergence history of the industrial scale case with the support beams at the inlet.*

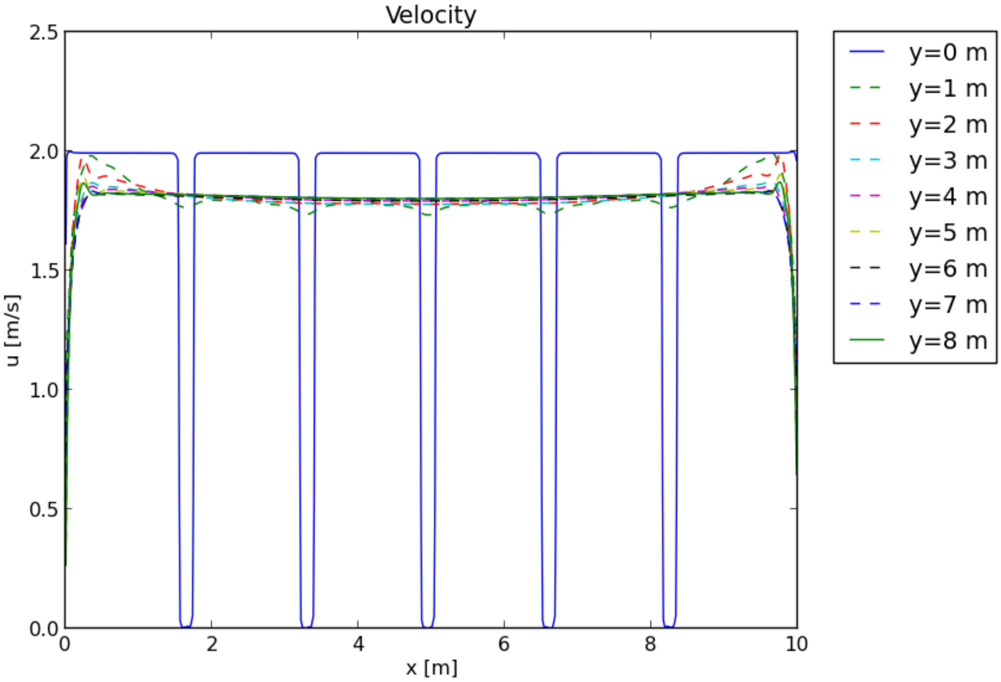


Figure 70: *Velocity profiles* (m/s) *at different heights in a column of height of* 8 m.

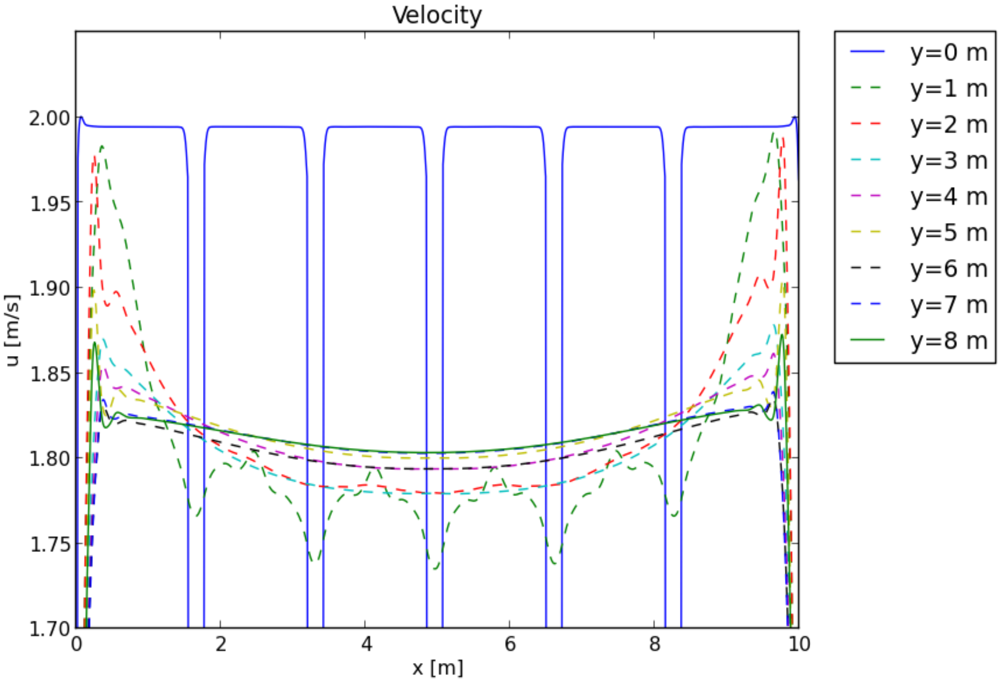


Figure 71: *Closer view of the velocity profiles* (m/s) *at different heights in a column of height of* 8 m*.*

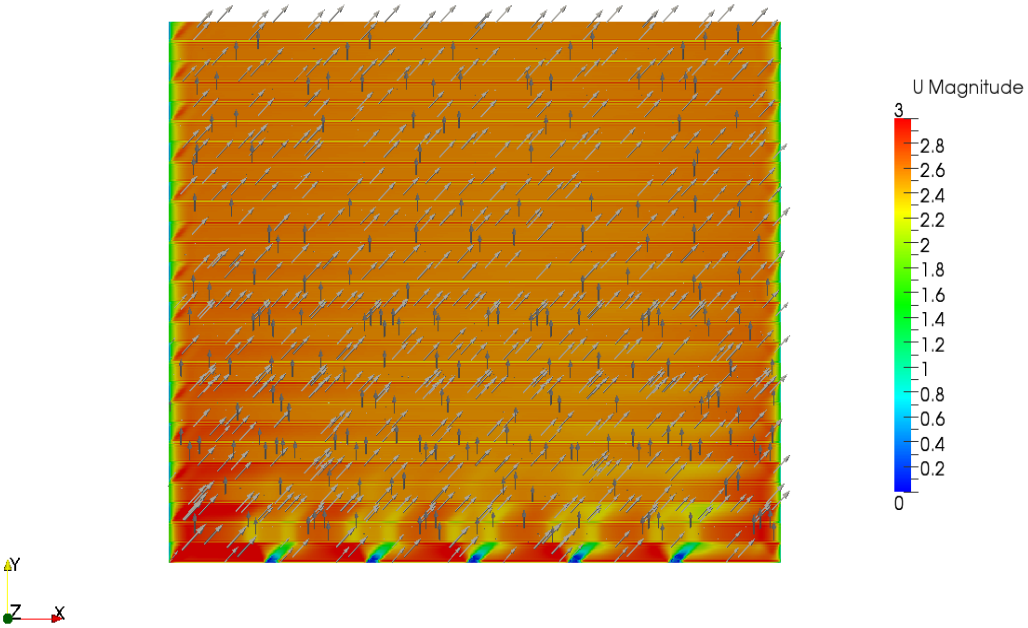


Figure 72: *Velocity magnitude* (m/s) *at the cross-section of* $z = 1.05$ m. *Velocity vectors are presented as grey arrows.*

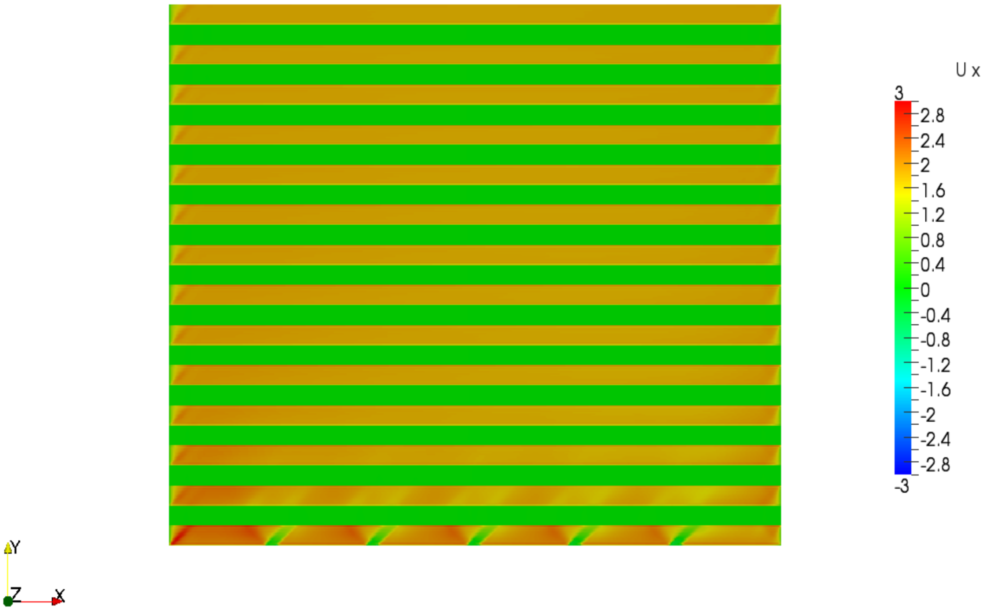


Figure 73: *Velocity in the* x*-direction* (m/s) *at the cross-section of* $z = 1.0$ m.

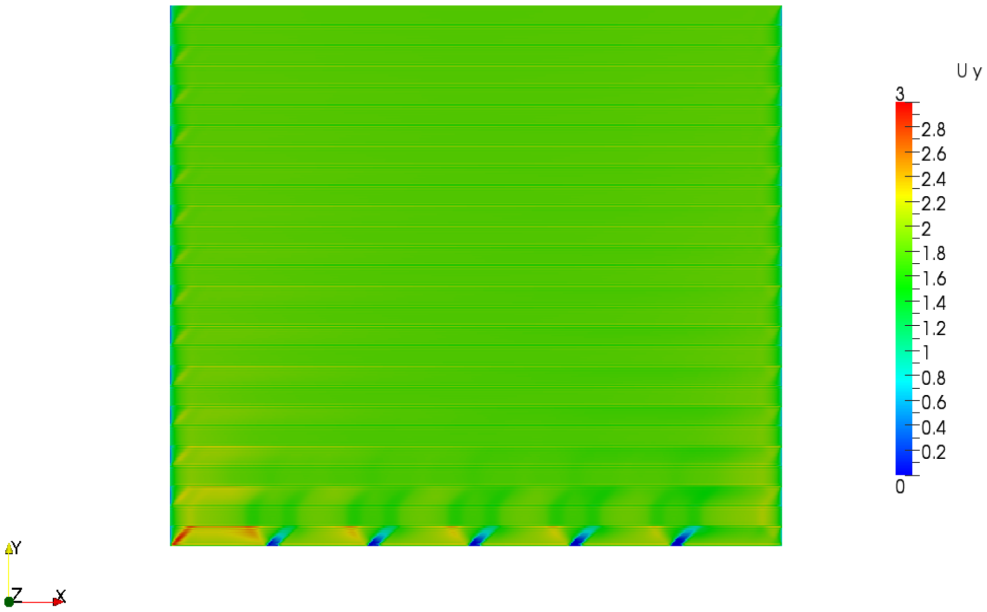


Figure 74: *Velocity in the y-direction* (m/s) *at the cross-section of* $z = 1.0$ m.

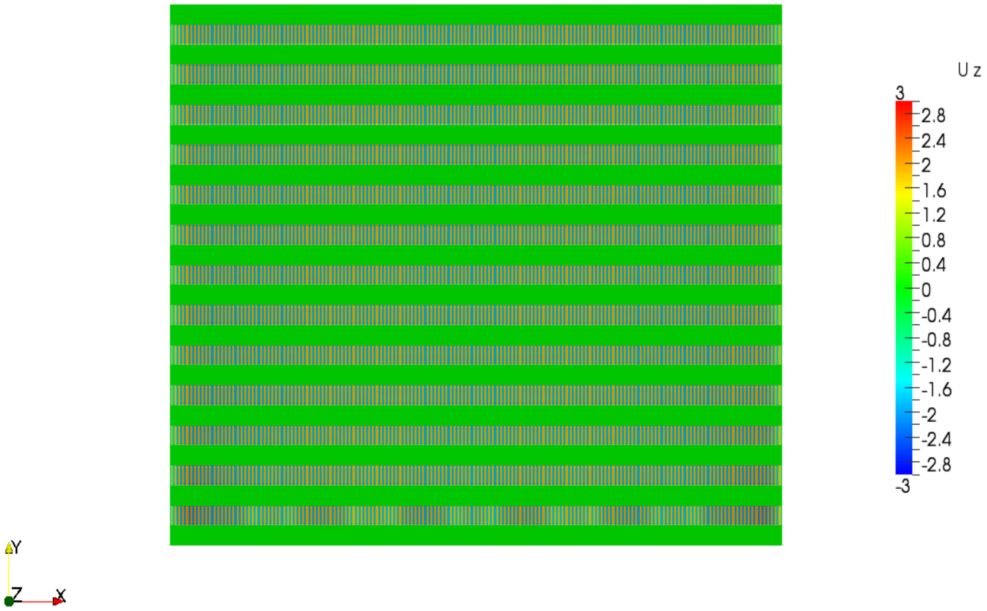


Figure 75: *Velocity in the z-direction* (m/s) *at the cross-section of* $z = 1.0$ m.

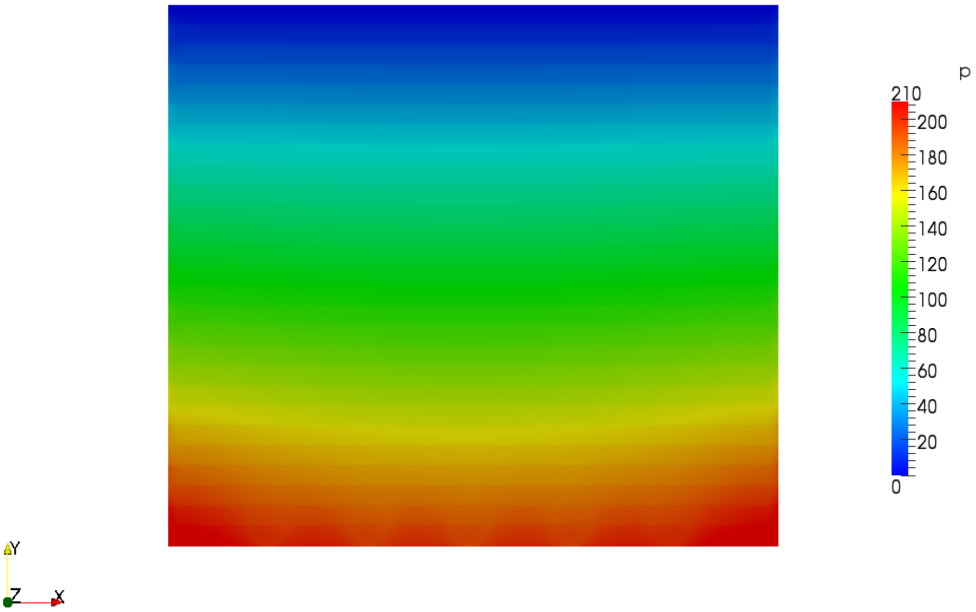


Figure 76: *Pressure field* (Pa/(kg/m^3)) *at the cross-section* $z = 1.0$ m.

7.7 Bed pressure drop

The porous media parameters were tuned according to the pressure drop across the bed with different inlet velocities in Chapt. 7.3. The model was then applied in larger column in Chapt. 7.4 and checked, whether the maldistribution can be captured. The pressure drop in this larger scale simulations, however, differs a little from the measured values as is seen in Tab. 6. The reasons for this were discussed in Chapt. 7.4.

The simulations in industrial scale were made for a maldistributed case and for the case, where the support beams below the bed are taken into account. In the maldistributed industrial scale case the average pressure drop in simulations is $p = 440$ Pa and in the case of support beams it is $p = 250$ Pa. When calculating the pressure drop across the bed with Sulzer's SULCOL 3.0.8 column sizing program [58], approximately the value $p = 900$ Pa is obtained (see Appendix G), which is also in line with the pressure losses in real industrial columns. SULCOL assumes a round cross-section for the column, so the geometry of the column in the SULCOL is not exactly equivalent to the geometry in simulations presented in Chapt. 7.5.1 with periodic boundary conditions. In this case for SULCOL, the diameter of the column is set to 10 m, as the width of the column in simulations is 10 m. The calculation procedure in SULCOL is based on the data from measurements in laboratory and industrial scale columns and thus the results from SULCOL are considered rather reliable. Therefore, it is deducted that the pressure drop in simulations may be too low and therefore the maldistribution is modelled very conservatively, as the higher pressure drop would also smooth out the maldistributed gas phase better than in current results.

The reason for the significantly smaller pressure drop in simulations may occur because of the lack of the turbulence. Turbulence induces pressure losses because of the turbulent diffusion. The flow in bed was assumed laminar according to the packing Reynolds number and because the calculation conducted with laminar flow in bed in Chapt. 7.3 seemed to produce good results when compared to the experimental results. The flow in bed is presumably still turbulent to some extent, and this has maybe larger effect in the large industrial scale column than in laboratory scale, in which the resistance factor was obtained. The right turbulence production should be known in order to have more reliable results. In the previous computations for columns the pressure loss is usually a known parameter and the resistance for the isotropic porosity presenting bed is tuned according to the pressure loss. This is an option also for the anisotropic porosity model. In that case it is assumed that all the resistance, including turbulence, is taken into account in the porous media resistance factor.

The industrial scale case with the maldistributed inlet was calculated again with a coefficient $f_{e2} = 10$ tuned according to the pressure loss $p = 900$ Pa. In Fig. 77 the velocity profiles at the different heights are presented and it is seen, that the differences between the velocity profiles in Figs. 62 and 77 are marginal. Hence, the used packing cannot completely smooth out the severely distorted gas inlet profile in a 8 m high bed.

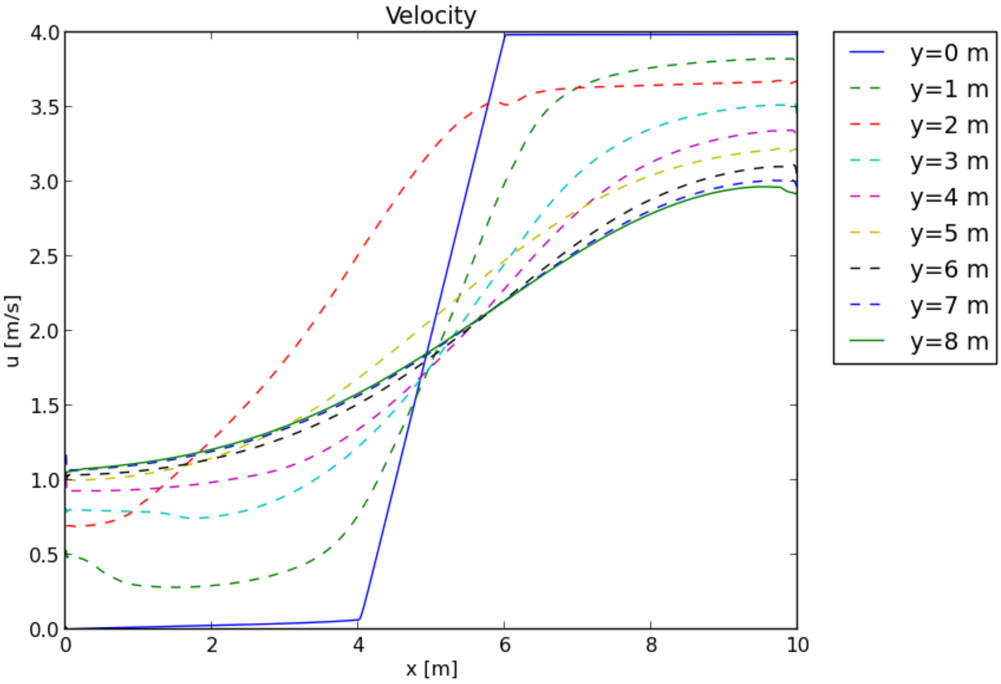


Figure 77: *Velocity profiles* (m/s) *at different heights in a column of height of* 8 m*.*

8 Recommendations for future work

In this work, the structured packed bed is modelled as an anisotropic porous media, as explained in Chapt. 7. In the new model it has been assumed that the resistance perpendicular to the metal sheets which form the packing is assumed to be 1000 times higher as the resistance in direction of the flow channels in the packing. Pressure loss measurements in different directions of the packing could provide important information of the resistance of the packing in different directions, which could benefit improving the porous model. The measurements could be conducted in a subsonic wind tunnel or in a channel built specifically for these measurements, for which the flow could be provided by a blower. The packing should be cut in directions perpendicular and parallel to the flow channels and additionally in some angles between them, if needed.

The flow field in a structured packed bed for example in an absorption column is a two-phase flow system consisting of gas and liquid phases. However, because of the restricted computational capacity and the large size of the columns in industrial scale, only a gas-phase flow by means of CFD was investigated in this work. In some previous studies, methods for taking additionally liquid phase into account are presented. The elementary cell model of Mahr and Mewes presented in Chapt. 6.1, allows taking the two-phase flow into account. The model is implemented in an existing CFD code, CFX 10.0. The applicability of the model for large scale computations could be further studied.

The effect of the liquid on the gas flow and the additional pressure loss due to the liquid flow in the packing can be investigated with the aid of VOF (Volume of Fluid) calculations. The results could benefit improving the parameters of the porous model, as is done in the study of Raynal presented in Chapt. 6.2 [23].

In this work, a flow in the porous media was assumed to be laminar according to Reynolds number of the hydraulic diameter of the flow channel in the structured packing (Tab. 4). However, flow in structured packing may be turbulent even if the flow channels presumably damp part of the turbulence by restricting the turbulence length scale to the diameter of the flow channels. Conventional turbulence models that are based on the global flow cannot model the anisotropic turbulence reliably and, therefore, in this work turbulence in the packing was suppressed by setting the turbulent kinematic viscosity to zero. Turbulence modelling in the packing could be improved with supplementary information on the turbulence in structured packing. Turbulence affects how the velocity differences smooth out and the pressure loss across the bed, as is seen in the results in Chapt. 7.3. Thus the appropriate modelling of turbulence is essential for the prediction of the flow field and pressure loss in a packed bed.

Investigation of the turbulence in the packing could be conducted with either CFD or experimentally. One option to study turbulence with CFD is direct numerical simulation (DNS), which is shortly explained in Chapt. 4.3.1. The computation size is limited in case of DNS, so that only few channels representing the packing structure could be modelled.

Turbulence can also be studied experimentally. The conventional wind tunnel measurements and Particle Image Velocimetry (PIV) could provide important data of the flow fields and turbulence. The wind tunnel measurements can be also conducted with a scaled (enlarged) model, and more accurate data of flow fields could be obtained. In the wind tunnel experiments the measurements with a hot-wire anemometer will give accurate data of velocity fluctuations and thus the turbulence measurements are possible [59].

The information of turbulence obtained from either numerical or experimental studies can be applied to investigate whether the turbulence is modelled correctly with conventional turbulence models or not. If not, the data can be used to modify existing turbulence models appropriate for structured packing. For example modifying turbulence production term according to the data on turbulence would also change the value of turbulent kinematic viscosity ν_t to a more appropriate value. This could be an option to model average turbulence correctly.

In this model, the flow was assumed to be incompressible. Taking compressibility into account necessitates using the solver for compressible flows in OpenFOAM, which takes also porosity into account. There are three appropriate solvers for compressible flow in porous media in OpenFOAM 2.1.1:

- `rhoPorousMRFSimpleFoam`: A steady-state solver for laminar or turbulent flow of compressible fluids with RANS turbulence modelling. Model includes implicit or explicit porosity treatment and moving reference frame option (MRF), which can be used for example to determine moving parts in the grid.

- `rhoPorousMRFPimpleFoam`: Transient solver for laminar or turbulent flow of compressible fluids. Model includes implicit or explicit porosity treatment and multiple reference frame option (MRF).

- `rhoPorousMRFLTSPimpleFoam`: Transient solver for laminar or turbulent flow of compressible fluids with local time-stepping for efficient steady-state solution. Model includes implicit or explicit porosity treatment and multiple reference frame option (MRF).

The porous model applied in this thesis in incompressible solver `porousSimpleFoam` could be applied to the presented compressible solvers, too. The main file defining porosity is `porousZones`, which in compressible solvers includes the temperature. The effect of the compressibility on the porous model should be tested against the experimental results.

9 Conclusions

The aim of this research was to create a practical CFD model for engineering purposes for a structured packed beds, which are used for example in absorption columns. The previous studies associated to fluid dynamics computations in packed beds were studied and a simple approach to model only the gas flow in the packed bed was selected. In case of large scale applications, such as industrial scale columns, modelling the exact geometry of the packing is computationally too expensive. Therefore, the structured packed bed is described as an anisotropic porous media. For fluid dynamics computations, an open-source CFD software, OpenFOAM 2.1.1, is applied. OpenFOAM 2.1.1 includes an option to calculate flows in porous media. The porous media is taken into account as a supplementary resistance in the momentum equation and modelled by the Darcy-Forchheimer equation (70), as explained in Chapt. 5.1. In OpenFOAM 2.1.1, the local coordinates for porous media can be defined by the user and the coefficients in Darcy-Forchheimer equation are defined separately for each local coordinate direction. Thus it is possible to take into account the flow channels and wall sheets in the structured packing through the coefficients of the porous media model.

The coefficients of the Darcy-Forchheimer equation for porous media are tuned according to the experimental results of Owens [24]. The calculations for the column similar to the column in Owens study are presented in Chapt. 7.3. Many simplifying assumptions were made in order to obtain the coefficients. It was assumed, that the gas density ρ is constant. It was also assumed, that the inertial term in Darcy-Forchheimer equation dominates in the flow through porous media, since the Reynolds number according to the hydraulic diameter of the packing flow channel is $Re = \{562 - 3096\}$ and according to [40] the transition from Darcy flow (viscous term dominates) to Darcy-Forchheimer flow occurs at the Reynolds number of order 10^2. Therefore, only the coefficient of inertial term, f, is defined for the packing. The coefficient f perpendicular to the sheets in the packing was assumed to be 1000 times the coefficient f in the direction of flow channels. Thus in practice the sheets are assumed impermeable. This approximation was also applied in study of Raynal [23]. The parameters of the packing depend now on many assumptions. Therefore, more investigations for example on pressure drop in different directions should be conducted in order to obtain more data on the parameters of the packing in different directions.

The coefficient f in the flow channel direction is obtained by iteratively calculating the flow through the column with the different velocities, which were also applied in Owens experiments. Finally, a coefficient f that satisfies the experimental curve, i.e. pressure loss as a function of a velocity, was obtained. The advantage to investigate the pressure loss in a small scale column of Owens presented in Chapt. 7.3, is that it can be assumed that the flow is rather evenly distributed, thus the maldistribution does not have an effect on the coefficient validation results. The packing used in Owens experiments was Mellapak 250Y, which is a common packing type in absorption columns.

After tuning the coefficients according to the pressure loss in Chapt. 7.3, the ability of the model to capture maldistribution in larger scale was studied. The Olujić study [27] presented in Chapt. 6.4 provides data of maldistribution experiments in packed beds. The packing in the pilot scale column was the Montz-pak B1-250, which has same geometric characteristics as Mellapak 250Y, except thatthe Mellapak 250Y consists of perforated sheets and the Montz-pak B1-250 of non-perforated sheets. The packing geometry is, however, not modelled exactly but approximated by the directional porous media approach, so the slight differences in packing details are not severe. In the simulation, lower pressure losses across the bed were obtained than in experimental results, and this may be because of the differences in the packing sheets. The study of Olujić included investigations of initial maldistribution. Two cases, chordal blockage and central blockage at the inlet of the bed were chosen. The simulations with a porous model for these two cases were conducted in Chapt. 7.4 and the results with an anisotropic porous model predicted the maldistribution in the packing according to the experimental results. In Chapt. 7.4, also isotropic porous model was applied in simulations for the two maldistributed cases, and it was noticed that in case of isotropic porous media, the maldistribution smoothed out too fast when compared to the experimental results. According to these results, the anisotropic porosity model seems to capture the maldistribution of the gas phase in structured packed bed. In order to obtain accurate pressure losses, probably a small scale pressure loss investigation must be made for the specific packing in consideration.

After the model validation, two simulations for an industrial scale bed of 10 m width were conducted. In the first simulation a severe maldistribution below the bed was assumed, so that the velocity differences at bed inlet are from 0 m/s to 4 m/s. The results indicated that this maldistribution does not smooth out completely even in a bed of height of 8 m. In the second industrial scale simulation, the support beams below the packed bed were taken into account as five impermeable walls. The beams induced local distributions to the velocity field near the inlet of the bed, but the distributions smoothed out approximately in the height of three packing layers. Thus it can be deducted that the support beams do not induce severe maldistribution to the packed bed.

The final outcome of this work is experience in using an anisotropic porous model. The model was systematically validated, compared to earlier models and applied an industrial scale cases to obtain new information on the flow in structured packed beds.

References

[1] www.cleen.fi/en, cited 23.4.2013

[2] CO_2 Capture And Storage -A Key Abatement Option. ISBN 978-92-64-04140-0

[3] Kulichenko, N., Ereira, E. Carbon Capture and Storage in Developing Countries: A Perspective on Barriers to Deployment. Washington D.C., USA: International Bank for Reconstruction and Development/The World Bank, 2012. ISBN 978-0-8213-9610-0.

[4] Energy And the New Reality 2 -Carbon-Free Energy Supply. Harvey, L.D.D.

[5] Teir, S., Tsupari, E., Koljonen, T., Pikkarainen, T., Kujanpää, L., Arasto, A., Tourunen, A., Kärki, J., Nieminen, M., Aatos, S. Hiilidioksidin talteenotto ja varastointi (CCS). 2009. VTT Tiedotteita -Research Notes 2503. pp. 61.

[6] Kleine, O. Tislauskolonnien täytekerroksen laskenta ja mitoitus, Diplomityö. Lappeenrannan teknillinen korkeakoulu, Lappeenranta 1998. 119 p.

[7] Stichlmair, J.G., Fair, J.R. Distillation: principle and practice. USA: Wiley-VCH, 1998.

[8] Chalmers, H., Gibbins, J. Initial evaluation of the impact of post-combustion capture of carbon dioxide on supercritical pulverised coal power plant part load performance. Fuel, 2007. Vol. 86(14). P. 2109-2123.

[9] Stikkelman, R.M. Gas and Liquid Maldistribution in Packed Columns. Dissertation. Delft University of Technology. Delft, 1989. 87 p.

[10] http://www.aspentech.com/products/aspen-plus.aspx, cited 2.10.2013

[11] Controlled liquid maldistribution studies on structured packing. Chemical Engineering Research & Design (Trans IChemE), 1999. Vol. 77 Part A September.

[12] Kolev, N. Packed Bed Columns for Absorption, Desorption, Rectification and Direct Heat Transfer, 1st ed. Amsterdam, Netherlands: Elsevier B.V., 2006. ISBN-13: 978-0-444-52829-2, ISBN-10: 0-444-52829-6.

[13] Machowiak, J. Fluiddynamik von Füllkörpern und Packungen, Grundlagen der Kolonnenauslegung. 2.Aufl. Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2003. ISBN 3-540-00493-9.

[14] Machowiak, J. Fluid Dynamics of Packed Columns, Principles of the Fluid Dynamic Design of Columns for Gas/Liquid and Liquid/Liquid Systems. Springer-Verlag Berlin Heidelberg, 2010. ISBN 978-3-540-88780-5.

[15] Billet, R. Packed Towers in Processing and Environmental Technology, 1st ed. Weinheim, Germany: VCH Verlagsgesellschaft mbH, 1995. ISBN 3-527-28616-0.

[16] Kister, H.Z. Distillation Design, 1st ed. USA: McGraw-Hill, 1992. ISBN-0-07-034909-6.

[17] http://www.sulzer.com/en/Products-and-Services/Separation-Technology/Structured-Packings/Mellapak-MellapakPlus-Mellapak-Plastic, cited 1.6.2013

[18] Darakchiev, R., Dodev, Ch. Chem. Eng. Process, 2002. Vol. 41. pp. 385.

[19] Yuan, X., Li, W. Institutions of Chemical Engineers Symposium Series No 142, 1997. Part 2. pp. 931.

[20] Olujić, Z., Haaring, J.P., van Baak, R. Effect of a severe form of initial gas maldistribution on pressure drop of a structured packing bed. Chemical Engineering and Processing, 2006. Vol. 45. P. 1059-1064.

[21] Cai, T.J., Chen, G.X., Fitz, C.W., Kunesh, J.G. Vapor Maldistribution Studies on Structured and Random Packings. Chemical Engineering Research and Design, 2003. Vol. 81 Part A.

[22] Mahr, B., Mewes, D. CFD Modelling And Calculation of Dynamic Two-Phase Flow in Columns Equipped With Structured Packing. Chemical Engineering Research and Design, 2007. Vol. 85 (A8). P.1112-1122.

[23] Raynal, L., Royon-Lebeaud, A. A Multi-Scale Approach For CFD Calculations Of Gas-Liquid Flow Within Large Size Column Equipped With Structured Packing. Chemical Engineering Science, 2007. Vol. 62. P. 7196-7204.

[24] Owens, S.A. Advanced Analysis of Structured Packing Using Computational Fluid Dynamics Simulations. Dissertation. The University of Texas at Austin, 2010. 144 p.

[25] Owens, S.A., Kossmann, A., Farone, J., Eldridge, R.B. Flow Field Visualization in Structured Packing Using Real Time X-ray Radiography. Industrial and Engineering Chemistry Research, 2009. Vol. 48. P. 3606-3618.

[26] Olujić, Z. Comparison of Gas Distribution Properties of Conventional and High Capacity Structured Packings. Chinese Journal of Chemical Engineering, 2011. Vol. 19(5) P. 726-732.

[27] Olujić, Z., Mohamed Ali A., Jansens, P.J. Effect of the initial gas maldistribution on the pressure drop of structured packings. Chemical Engineering and Processing, 2004. Vol. 43. P. 465-476.

[28] Siikonen, T. Virtaussimulointi (in Finnish), 2010. Aalto University, Finland.

[29] Siikonen, T. Laskennallisen virtausmekaniikan ja lämmönsiirron peruskurssi (in Finnish), 2012. Aalto University, Finland.

[30] Young, D.F., Munson, B.R., Okiishi, T.H., Huebsch, W.W. A Brief Introduction to Fluid Mechanics, 4th ed. USA: John Wiley & Sons, Inc., 2007. ISBN 978-0-470-03962-5.

[31] White, F.M. Viscous Fluid Flow, 3rd ed. New York, USA: McGraw-Hill, 2006. ISBN 007-124493-X.

[32] Ferziger, J.H., Peric M. Computational Methods for Fluid Dynamics, 2nd ed. Germany: Springer-Verlag Berlin Heidelberg Newyork, 1996. ISBN 3-540-59434-5.

[33] Juretic, F. Error Analysis in Finite Volume CFD. Dissertation. Imperial College London, 2004. 217 p.

[34] Cebeci, T., Shao, J.P., Kafyeke, F., Laurendeau, E. Computational Fluid Dynamics for Engineers. Germany: Springer-Verlag Berlin Heidelberg Newyork, 2005. ISBN 3-540-24451-4.

[35] Anderson, D.A., Tannehill, J.C., Pletcher, R.H. Computational Fluid Mechanics and Heat Transfer, 2nd ed. Washington, USA: Taylor & Francis, 1984. ISBN 1-56032-046-X.

[36] Pope, S.B. Turbulent Flows. Cambridge, United Kingdom: Cambridge University Press, 2000. ISBN 0-521-59886-9.

[37] Menter, F.R., Kuntz, M., Langtry, R. Ten Years of Industrial Experience with the SST Turbulence Model. Turbulence, Heat and Mass Transfer 4, 2003.

[38] Moradnia, P. CFD of Air Flow in Hydro Power Generators. Thesis for Licenciate of Engineering. Chalmers University of Technology. Göteborg, Sweden 2010. ISSN 1652-8565.

[39] Pietarinen, K. Fluid Flow and Dispersion Predictions in Porous Media by Utilizing Advanced Numerical Methods. Masters Thesis. Lappeenranta University of Technology. Lappeenranta, Finland. 2005. 137 p.

[40] Donald A. Nield and Adrian Bejan. Convection in Porous Media. New York, USA: Springer, 2006. ISBN 978-0387-29096-6.

[41] Jambhekar, V.A. Forchheimer Porous-media Flow Models - Numerical Investigation and Comparison with Experimental Data. Masters Thesis. Stuttgart University of Technology. Stuttgart, Germany, 2011.

[42] Lemos, M.J.S. Turbulence in Porous Media, 1st ed. Oxford, Great Britain: Elsevier Ltd., 2006. ISBN-13: 978-0-080-44491-8.

[43] Teruel, F.E. Macroscopic Turbulence Modeling and Simulation for Flow Through Porous Media. Dissertation. University of Illinois. Urbana, Illinois, 2007.

[44] Antohe, B.V., Lage,J.L. A general two-equation macroscopic turbulence model for incompressible flow in porous media. International Journal of Heat and Mass Transfer, 1997. Vol 40, No. 13. P. 3013-3024.

[45] Chan, H.C., Huang, W.C., Leu, J.M., Lai, C.J. Macroscopic modeling of turbulent flow over a porous medium. International Journal of Heat and Fluid Flow, 2007. Vol. 28. P. 1157-1166.

[46] Nield, D.A. Alternative Models of Turbulence in a Porous Medium, and Related Matters. Journal of Fluids Engineering, 2001. Vol. 123. pp. 928-931.

[47] Weller H.G., Tabor G., Jasak H., Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. American Institute of Physics, 1998. S0894-1866(98)01906-3

[48] http://www.openfoam.org/download/, cited 2.10.2013

[49] OpenFOAM 2.1.1 User Guide

[50] OpenFOAM 2.1.1 Programmers Guide

[51] http://www.paraview.org/, cited 10.9.2013

[52] http:/aerojet.engr.ucdavis.edu/gambithelp/index.htm, cited 5.9.2013

[53] Vaittinen, J., Björklöf, T., Nyman, T. CFD and kinetic modelling of postcombustion CO_2 absorber. Neste Jacobs Oy report, 28.2.2013.

[54] http://openfoamwiki.net/index.php/Contrib/swak4Foam, cited 3.9.2013

[55] http://www.openfoam.org/version2.1.0/ami.php, cited 3.9.2013

[56] http://www.openfoam.com/features/mesh-manipulation.php, cited 17.9.2013

[57] http://openfoamwiki.net/index.php/SetSet, cited 17.9.2013

[58] https://www.sulzer.com/en/Resources/Online-Tools/Sulcol, cited 25.9.2013

[59] Jørgensen, F.E. How to measure turbulence with hot-wire anemometers - a practical guide. Dantec Dynamics, 2002.

A Appendix: Numerical schemes

Numerical schemes of OpenFOAM defined in the file `/system/fvSchemes`:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
ddtSchemes
{
    default         steadyState;
}
gradSchemes
{
    grad(U)        Gauss linear;
    grad(p)   Gauss linear;
    grad(k)   Gauss linear;
    grad(omega)    Gauss linear;
}
divSchemes
{
    div(phi,U)         Gauss linearUpwindV grad(U);
    div((nuEff*dev(T(grad(U)))))    Gauss linear;
    div(phi,omega)    Gauss linear; //upwind;
    div(phi,k)          Gauss linear; //upwind;
}
laplacianSchemes
{
    laplacian(nuEff,U)  Gauss linear corrected;
    laplacian(rAU,p)  Gauss linear corrected;
    laplacian(DomegaEff,omega)  Gauss linear corrected;
    laplacian(DkEff,k)  Gauss linear corrected;
}
interpolationSchemes
{
    interpolate(U)     linear;
    interpolate(grad(p))     linear;
}
snGradSchemes
{
}
fluxRequired
{
    p               ;
}
// ************************************************************************* //
```

B Appendix: Solution methods

Solution methods for linear equations of OpenFOAM defined in the file `/system/fvSolution`:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
solvers
{
    p
    {
        solver           GAMG;
        tolerance        1e-08;
        relTol           0.05;
        smoother         GaussSeidel;
        cacheAgglomeration  off;
        nCellsInCoarsestLevel  20;
        agglomerator     faceAreaPair;
        mergeLevels      1;
    }
    "(k|omega)"
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }
}
SIMPLE
{
    nUCorrectors    2;
    nNonOrthogonalCorrectors 0;
}
relaxationFactors
{
    fields
    {
        p               0.1;
    }
    equations
    {
        U               0.1;
        k               0.3;
        omega           0.3;
    }
}
// ************************************************************************* //
```

C Appendix: Parabolic inlet velocity profile

The parabolic inlet velocity profile is defined in file `system/funkySetFieldsDict`:

```
/*-------------------------------*- C++ -*---------------------------------*/
//
// funkySetFields example for cavity tutorial
// run: blockMesh
// run: funkySetFields -time 0
//
// See instructions in:
// http://openfoamwiki.net/index.php/Contrib/swak4Foam
// http://openfoamwiki.net/index.php/Contrib_funkySetFields
//
FoamFile
{
    version         2.0;
    format          ascii;

    root            "";
    case            "";
    instance        "";
    local           "";

    class           dictionary;
    object          funkySetFieldsDict;
}
expressions
(
  // 1. initialize whole U field to rotation and copy
  // values to valuePatches

  initMovingWall
  {
    field U;

    variables
    (
      "x=pos().x;"
      "z=pos().z;"
      "r=sqrt(x*x+z*z);"     //origo in the middle
    );
    keepPatches true;
    valuePatches ("inlet_d1");
    expression "vector (0, -750.6*(r*r)+4.0, 0)";
  }
  // 2. Set internal field to a fixed value
  initInternalField
  {
    field U;
    keepPatches true;
    expression "vector (0, 0, 0)";
  }
);
```

D Appendix: Piecewise continuous inlet velocity profile

The piecewise continuous inlet velocity profile is defined in file `system/funkySetFieldsDict`:

```
/*--------------------------------*- C++ -*----------------------------------*/
//
// funkySetFields example for cavity tutorial
// run: blockMesh
// run: funkySetFields -time 0
//
// See instructions in:
// http://openfoamwiki.net/index.php/Contrib/swak4Foam
// http://openfoamwiki.net/index.php/Contrib_funkySetFields
//
FoamFile
{
    version         2.0;
    format          ascii;

    root            "";
    case            "";
    instance        "";
    local           "";

    class           dictionary;
    object          funkySetFieldsDict;
}


expressions
(
  // 1. initialize whole U field to rotation and copy values to valuePatches
  initMovingWall1
  {
    field U;

    variables
    (
      "x=pos().x;"
      "y=pos().y;"
      "z=pos().z;"
    );

    //keepPatches true;
    valuePatches ("inlet");

    expression "vector (0, 0, 0)";
    condition "pos().x < 4.0";
  }
  initMovingWall2
  {
    field U;

    variables
    (
      "x=pos().x;"
      "y=pos().y;"
      "z=pos().z;"
    );

    //keepPatches true;
    valuePatches ("inlet");
    expression "vector (0, 4.0, 0)";
    condition "pos().x > 6.0";
```

```
  }
  initMovingWall3
  {
    field U;

    variables
    (
      "x=pos().x;"
      "y=pos().y;"
      "z=pos().z;"
    );

    //keepPatches true;
    valuePatches ("inlet");
    expression "vector (0, 2*x-8.0, 0)";
    condition "pos().x >= 4.0 && pos().x <= 6.0";
  }

  // 2. Set internal field to a fixed value
  initInternalField
  {
    field U;
    keepPatches true;
    expression "vector (0, 0, 0)";
  }

);
```

E Appendix: Implementing a new turbulence model in OpenFOAM

Implementing an own turbulence model in OpenFOAM using `kOmegaSST` as a base is presented in this appendix. The modifications to `kOmegaSST` are made to set $\nu_t = 0$ in the porous zones. The new turbulence model is thus called `porouskOmegaSST`

- Go to the folder containing turbulence models, like:
  `/opt/openfoam211/src/turbulenceModels/incompressible/RAS`

- Copy `kOmegaSST`-folder as a new folder in your user account :
  `cp -r -parents src/turbulenceModels/incompressible/RAS/kOmegaSST $WM_PROJECT_USER_DIR`

- Go to the copied folder:
  `cd $WM_PROJECT_USER_DIR/src/turbulenceModels/incompressible/RAS` and give a command `sed -i ’s/kOmegaSST/porouskOmegaSST/g’ *`

- Copy `Make`-folder from `$FOAM_SRC/turbulenceModels/incompressible/RAS/Make` to the copied turbulence model folder. Do the modifications to obtain

  `Make/files`:

```
EXE_INC = \
    -I$(LIB_SRC)/turbulenceModels \
    -I$(LIB_SRC)/transportModels \
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/meshTools/lnInclude
    -I$(LIB_SRC)/turbulenceModels/incompressible/RAS/lnInclude

LIB_LIBS = \
    -lincompressibleTurbulenceModel \
    -lincompressibleRASModels \
    -lfiniteVolume \
    -lmeshTools
```

  `Make/files`:

```
porouskOmegaSST/porouskOmegaSST.C
LIB = $(FOAM_USER_LIBBIN)/libuserRAS
```

- In folder `user/src/turbulenceModels/incompressible/RAS` give command `wmake libso`

- Add a line in the `controlDict`-file:
  `libs ("libuserRAS.so");`

- The new turbulence model should work as the new turbulence model is given in a file `constant/RASProperties`.

The modified `kOmegaSST.C` -file for porous media: `porouskOmegaSST.C`

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

\*---------------------------------------------------------------------------*/

#include "porouskOmegaSST.H"
#include "addToRunTimeSelectionTable.H"

#include "backwardsCompatibilityWallFunctions.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{
namespace incompressible
{
namespace RASModels
{

// * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //

defineTypeNameAndDebug(porouskOmegaSST, 0);
addToRunTimeSelectionTable(RASModel, porouskOmegaSST, dictionary);

// * * * * * * * * * * * * Private Member Functions  * * * * * * * * * * * * //

tmp<volScalarField> porouskOmegaSST::F1(const volScalarField& CDkOmega) const
{
    tmp<volScalarField> CDkOmegaPlus = max
    (
        CDkOmega,
        dimensionedScalar("1.0e-10", dimless/sqr(dimTime), 1.0e-10)
    );

    tmp<volScalarField> arg1 = min
    (
        min
        (
            max
            (
                (scalar(1)/betaStar_)*sqrt(k_)/(omega_*y_),
                scalar(500)*nu()/(sqr(y_)*omega_)
            ),
            (4*alphaOmega2_)*k_/(CDkOmegaPlus*sqr(y_))
        ),
```

```
        scalar(10)
    );

    return tanh(pow4(arg1));
}

tmp<volScalarField> porouskOmegaSST::F2() const
{
    tmp<volScalarField> arg2 = min
    (
        max
        (
            (scalar(2)/betaStar_)*sqrt(k_)/(omega_*y_),
            scalar(500)*nu()/(sqr(y_)*omega_)
        ),
        scalar(100)
    );

    return tanh(sqr(arg2));
}


// * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * //

porouskOmegaSST::porouskOmegaSST
(
    const volVectorField& U,
    const surfaceScalarField& phi,
    transportModel& transport,
    const word& turbulenceModelName,
    const word& modelName
)
:
    RASModel(modelName, U, phi, transport, turbulenceModelName),

    alphaK1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaK1",
            coeffDict_,
            0.85034
        )
    ),
    alphaK2_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaK2",
            coeffDict_,
            1.0
        )
    ),
    alphaOmega1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaOmega1",
            coeffDict_,
            0.5
        )
    ),
    alphaOmega2_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaOmega2",
            coeffDict_,
```

```
        0.85616
    )
),
gamma1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "gamma1",
        coeffDict_,
        0.5532
    )
),
gamma2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "gamma2",
        coeffDict_,
        0.4403
    )
),
beta1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "beta1",
        coeffDict_,
        0.075
    )
),
beta2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "beta2",
        coeffDict_,
        0.0828
    )
),
betaStar_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "betaStar",
        coeffDict_,
        0.09
    )
),
a1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "a1",
        coeffDict_,
        0.31
    )
),
c1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "c1",
        coeffDict_,
        10.0
    )
),

y_(mesh_),
```

```
    k_
    (
        IOobject
        (
            "k",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        autoCreateK("k", mesh_)
    ),
    omega_
    (
        IOobject
        (
            "omega",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        autoCreateOmega("omega", mesh_)
    ),
    nut_
    (
        IOobject
        (
            "nut",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        autoCreateNut("nut", mesh_)
    )
{
    bound(k_, kMin_);
    bound(omega_, omegaMin_);

    nut_ =
    (
        a1_*k_
      / max
        (
            a1_*omega_,
            F2()*sqrt(2.0)*mag(symm(fvc::grad(U_)))
        )
    );
    nut_.correctBoundaryConditions();

    printCoeffs();
}


// * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * //

tmp<volSymmTensorField> porouskOmegaSST::R() const
{
    return tmp<volSymmTensorField>
    (
        new volSymmTensorField
        (
            IOobject
            (
                "R",
                runTime_.timeName(),
```

```
                mesh_,
                IOobject::NO_READ,
                IOobject::NO_WRITE
            ),
            ((2.0/3.0)*I)*k_ - nut_*twoSymm(fvc::grad(U_)),
            k_.boundaryField().types()
        )
    );
}


tmp<volSymmTensorField> porouskOmegaSST::devReff() const
{
    return tmp<volSymmTensorField>
    (
        new volSymmTensorField
        (
            IOobject
            (
                "devRhoReff",
                runTime_.timeName(),
                mesh_,
                IOobject::NO_READ,
                IOobject::NO_WRITE
            ),
           -nuEff()*dev(twoSymm(fvc::grad(U_)))
        )
    );
}


tmp<fvVectorMatrix> porouskOmegaSST::divDevReff(volVectorField& U) const
{
    return
    (
      - fvm::laplacian(nuEff(), U)
      - fvc::div(nuEff()*dev(T(fvc::grad(U))))
    );
}


bool porouskOmegaSST::read()
{
    if (RASModel::read())
    {
        alphaK1_.readIfPresent(coeffDict());
        alphaK2_.readIfPresent(coeffDict());
        alphaOmega1_.readIfPresent(coeffDict());
        alphaOmega2_.readIfPresent(coeffDict());
        gamma1_.readIfPresent(coeffDict());
        gamma2_.readIfPresent(coeffDict());
        beta1_.readIfPresent(coeffDict());
        beta2_.readIfPresent(coeffDict());
        betaStar_.readIfPresent(coeffDict());
        a1_.readIfPresent(coeffDict());
        c1_.readIfPresent(coeffDict());

        return true;
    }
    else
    {
        return false;
    }
}


void porouskOmegaSST::correct()
{
```

```
    RASModel::correct();

    if (!turbulence_)
    {
        return;
    }

    if (mesh_.changing())
    {
        y_.correct();
    }

    const volScalarField S2(2*magSqr(symm(fvc::grad(U_))));
    volScalarField G("RASModel::G", nut_*S2);

    // Update omega and G at the wall
    omega_.boundaryField().updateCoeffs();

    const volScalarField CDkOmega
    (
        (2*alphaOmega2_)*(fvc::grad(k_) & fvc::grad(omega_))/omega_
    );

    const volScalarField F1(this->F1(CDkOmega));

    // Turbulent frequency equation
    tmp<fvScalarMatrix> omegaEqn
    (
        fvm::ddt(omega_)
      + fvm::div(phi_, omega_)
      - fvm::Sp(fvc::div(phi_), omega_)
      - fvm::laplacian(DomegaEff(F1), omega_)
     ==
        gamma(F1)*S2
      - fvm::Sp(beta(F1)*omega_, omega_)
      - fvm::SuSp
        (
            (F1 - scalar(1))*CDkOmega/omega_,
            omega_
        )
    );

    omegaEqn().relax();
//mesh_.cellZones().findIndices("porosity1")

    omegaEqn().boundaryManipulate(omega_.boundaryField());

    solve(omegaEqn);

    bound(omega_, omegaMin_);

    // Turbulent kinetic energy equation
    tmp<fvScalarMatrix> kEqn
    (
    //forAll(mesh_.cellZones()["porosity4"], i)
    //{
// omega_[i] = 10.*omega_[i];
    //}
        fvm::ddt(k_)
      + fvm::div(phi_, k_)
      - fvm::Sp(fvc::div(phi_), k_)
      - fvm::laplacian(DkEff(F1), k_)
     ==
        min(G, c1_*betaStar_*k_*omega_)
      - fvm::Sp(betaStar_*omega_, k_)
    );

    kEqn().relax();
```

```
    solve(kEqn);
    bound(k_, kMin_);


    // Re-calculate viscosity

    nut_ = a1_*k_/max(a1_*omega_, F2()*sqrt(S2));
    nut_.correctBoundaryConditions();

    //---Modified to set nut = 0 in porousZones.
    //---Check porousZone.C
    Info<< "Setting nut=0 in porosity1" << endl;
    const labelList& cells1 = mesh_.cellZones()["porosity1"];
    forAll(cells1, i)
    //forAll(mesh_.cellZones()["porosity1"], i)
    {
     nut_[cells1[i]] = 0.;
    }
    Info<< "Setting nut=0 in porosity2" << endl;
    const labelList& cells2 = mesh_.cellZones()["porosity2"];
    forAll(cells2, i)
    //forAll(mesh_.cellZones()["porosity2"], i)
    {
     nut_[cells2[i]] = 0.;
    }
    Info<< "Setting nut=0 in porosity3" << endl;
    const labelList& cells3 = mesh_.cellZones()["porosity3"];
    forAll(cells3, i)
    //forAll(mesh_.cellZones()["porosity3"], i)
    {
     nut_[cells3[i]] = 0.;
    }
    Info<< "Setting nut=0 in porosity4" << endl;
    const labelList& cells4 = mesh_.cellZones()["porosity4"];
    forAll(cells4, i)
    //forAll(mesh_.cellZones()["porosity4"], i)
    {
     nut_[cells4[i]] = 0.;
    }
    //---
}


// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

} // End namespace RASModels
} // End namespace incompressible
} // End namespace Foam

// ************************************************************************* //
```

F Appendix: Grid generation in OpenFOAM

A simple grid is generated with `blockMesh`-utility. The grid is defined in file `constant/polyMesh/blockMesh`:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
convertToMeters 1.0;
vertices
(
    (0 0 0)
    (1.4 0 0)
    (1.4 1.5 0)
    (0 1.5 0)
    (0 0 1.4)
    (1.4 0 1.4)
    (1.4 1.5 1.4)
    (0 1.5 1.4)
);
blocks
(
    hex (0 1 2 3 4 5 6 7) (140 150 140) simpleGrading (1 1 1)
);
edges
(
);
boundary
(
    inlet
    {
        type patch;
        faces
        (
            (1 5 4 0)

        );
    }
    outlet
    {
        type patch;
        faces
        (
            (3 7 6 2)
        );
    }
    wall
    {
        type wall;
        faces
        (
            (2 6 5 1)
            (0 4 7 3)
    (4 5 6 7)
            (0 3 2 1)
```

```
        );
    }
);
mergePatchPairs
(
);

// ************************************************************************* //
```

With `snappyHexMesh`-utility of OpenFOAM the arbitrary geometry from the existing `blockMesh`-grid can be taken by defining the geometry for example with STL-surfaces. The `snappyHexMesh`-utility is defined by user in the file `snappyHexMeshDict`:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Which of the steps to run
castellatedMesh true;
snap            true;
addLayers       false;

// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting it
// - to specify refinement for any mesh cell inside/outside/near
// - to 'snap' the mesh boundary to the surface
geometry
{
    inlet.stl
    {
        type triSurfaceMesh;
        name inlet;
    }

    outlet.stl
    {
        type triSurfaceMesh;
        name outlet;
    }

    wall.stl
    {
        type triSurfaceMesh;
        name wall;
    }

    refinementBox
    {
        type searchableBox;
        min (-1.0 -0.7 0.0);
        max ( 8.0  0.7 2.5);
    }
};

// Settings for the castellatedMesh generation.
castellatedMeshControls
{
    // Refinement parameters
    // ~~~~~~~~~~~~~~~~~~~~~

    // If local number of cells is >= maxLocalCells on any processor
    // switches from from refinement followed by balancing
```

```
// (current method) to (weighted) balancing before refinement.
maxLocalCells 10000000;

// Overall cell limit (approximately). Refinement will stop immediately
// upon reaching this number so a refinement level might not complete.
// Note that this is the number of cells before removing the part which
// is not 'visible' from the keepPoint. The final number of cells might
// actually be a lot less.
maxGlobalCells 30000000;

// The surface refinement loop might spend lots of iterations refining just a
// few cells. This setting will cause refinement to stop if <= minimumRefine
// are selected for refinement. Note: it will at least do one iteration
// (unless the number of cells to refine is 0)
minRefinementCells 0; // 10;

// Allow a certain level of imbalance during refining
// (since balancing is quite expensive)
// Expressed as fraction of perfect balance (= overall number of cells /
// nProcs). 0=balance always.
maxLoadUnbalance 0.10;

// Number of buffer layers between different levels.
// 1 means normal 2:1 refinement restriction, larger means slower
// refinement.
nCellsBetweenLevels 3;

// Explicit feature edge refinement
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// Specifies a level for any cell intersected by its edges.
// This is a featureEdgeMesh, read from constant/triSurface for now.
features
(
    {
        file "wall.eMesh";
        level 0;
    }
);

// Surface based refinement
// ~~~~~~~~~~~~~~~~~~~~~~~~

// Specifies two levels for every surface. The first is the minimum level,
// every cell intersecting a surface gets refined up to the minimum level.
// The second level is the maximum level. Cells that 'see' multiple
// intersections where the intersections make an
// angle > resolveFeatureAngle get refined up to the maximum level.

refinementSurfaces
{
    inlet
    {
        level (0 0);
    }

    outlet
    {
        level (0 0);
    }

    wall
    {
        level (0 0);
    }

}
```

```
    // Resolve sharp angles
    resolveFeatureAngle 30;

    // Region-wise refinement
    // ~~~~~~~~~~~~~~~~~~~~~~

    // Specifies refinement level for cells in relation to a surface. One of
    // three modes
    // - distance. 'levels' specifies per distance to the surface the
    //   wanted refinement level. The distances need to be specified in
    //   descending order.
    // - inside. 'levels' is only one entry and only the level is used. All
    //   cells inside the surface get refined up to the level. The surface
    //   needs to be closed for this to be possible.
    // - outside. Same but cells outside.

    refinementRegions
    {
/*
       refinementBox
        {
            mode inside;
            levels ((1E15 4));
        }
*/
    }

    // Mesh selection
    // ~~~~~~~~~~~~~~

    // After refinement patches get added for all refinementSurfaces and
    // all cells intersecting the surfaces get put into these patches. The
    // section reachable from the locationInMesh is kept.
    // NOTE: This point should never be on a face, always inside a cell, even
    // after refinement.
    locationInMesh (0.7 0.7 0.7);


    // Whether any faceZones (as specified in the refinementSurfaces)
    // are only on the boundary of corresponding cellZones or also allow
    // free-standing zone faces. Not used if there are no faceZones.
    allowFreeStandingZoneFaces false; // true;
}

// Settings for the snapping.
snapControls
{
    //- Number of patch smoothing iterations before finding correspondence
    //  to surface
    nSmoothPatch 3;

    //- Relative distance for points to be attracted by surface feature point
    //  or edge. True distance is this factor times local
    //  maximum edge length.
    tolerance 4.0;

    //- Number of mesh displacement relaxation iterations.
    nSolveIter 500; // 0;

    //- Maximum number of snapping relaxation iterations. Should stop
    //  before upon reaching a correct mesh.
    nRelaxIter 30; // 5;

    //- Highly experimental and wip: number of feature edge snapping
    //  iterations. Leave out altogether to disable.
    //  Do not use here since mesh resolution too low and baffles present
    nFeatureSnapIter 10;
}
```

```
// Settings for the layer addition.
addLayersControls
{
    // Are the thickness parameters below relative to the undistorted
    // size of the refined cell outside layer (true) or absolute sizes (false).
    relativeSizes true;

    // Per final patch (so not geometry!) the layer information
    layers
    {
    }

    // Expansion factor for layer mesh
    expansionRatio 1.0;

    //- Wanted thickness of final added cell layer. If multiple layers
    //  is the
    //  thickness of the layer furthest away from the wall.
    //  Relative to undistorted size of cell outside layer.
    //  is the thickness of the layer furthest away from the wall.
    //  See relativeSizes parameter.
    finalLayerThickness 0.3;

    //- Minimum thickness of cell layer. If for any reason layer
    //  cannot be above minThickness do not add layer.
    //  Relative to undistorted size of cell outside layer.
    minThickness 0.1;

    //- If points get not extruded do nGrow layers of connected faces that are
    //  also not grown. This helps convergence of the layer addition process
    //  close to features.
    // Note: changed(corrected) w.r.t 17x! (didn't do anything in 17x)
    nGrow 0;

    // Advanced settings

    //- When not to extrude surface. 0 is flat surface, 90 is when two faces
    //  make straight angle.
    featureAngle 30;

    //- Maximum number of snapping relaxation iterations. Should stop
    //  before upon reaching a correct mesh.
    nRelaxIter 3;

    // Number of smoothing iterations of surface normals
    nSmoothSurfaceNormals 1;

    // Number of smoothing iterations of interior mesh movement direction
    nSmoothNormals 3;

    // Smooth layer thickness over surface patches
    nSmoothThickness 10;

    // Stop layer growth on highly warped cells
    maxFaceThicknessRatio 0.5;

    // Reduce layer growth where ratio thickness to medial
    // distance is large
    maxThicknessToMedialRatio 0.3;

    // Angle used to pick up medial axis points
    // Note: changed(corrected) w.r.t 17x! 90 degrees corresponds to 130 in 17x.
    minMedianAxisAngle 90;

    // Create buffer region for new layer terminations
    nBufferCellsNoExtrude 0;
```

```
    // Overall max number of layer addition iterations. The mesher will exit
    // if it reaches this number of iterations; possibly with an illegal
    // mesh.
    nLayerIter 50;
}
// Generic mesh quality settings. At any undoable phase these determine
// where to undo.
meshQualityControls
{
    //- Maximum non-orthogonality allowed. Set to 180 to disable.
    maxNonOrtho 65;

    //- Max skewness allowed. Set to <0 to disable.
    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    //- Max concaveness allowed. Is angle (in degrees) below which concavity
    //  is allowed. 0 is straight face, <0 would be convex face.
    //  Set to 180 to disable.
    maxConcave 80;

    //- Minimum pyramid volume. Is absolute volume of cell pyramid.
    //  Set to a sensible fraction of the smallest cell volume expected.
    //  Set to very negative number (e.g. -1E30) to disable.
    minVol 1e-13;

    //- Minimum quality of the tet formed by the face-centre
    //  and variable base point minimum decomposition triangles and
    //  the cell centre. This has to be a positive number for tracking
    //  to work. Set to very negative number (e.g. -1E30) to
    //  disable.
    //     <0 = inside out tet,
    //      0 = flat tet
    //      1 = regular tet
    minTetQuality 1e-30;

    //- Minimum face area. Set to <0 to disable.
    minArea -1;

    //- Minimum face twist. Set to <-1 to disable. dot product of face normal
    //- and face centre triangles normal
    minTwist 0.02;

    //- minimum normalised cell determinant
    //- 1 = hex, <= 0 = folded or flattened illegal cell
    minDeterminant 0.001;

    //- minFaceWeight (0 -> 0.5)
    minFaceWeight 0.02;

    //- minVolRatio (0 -> 1)
    minVolRatio 0.01;

    //must be >0 for Fluent compatibility
    minTriangleTwist -1;


    // Advanced

    //- Number of error distribution iterations
    nSmoothScale 4;
    //- amount to scale back displacement at error points
    errorReduction 0.75;
}

// Advanced

// Flags for optional output
```

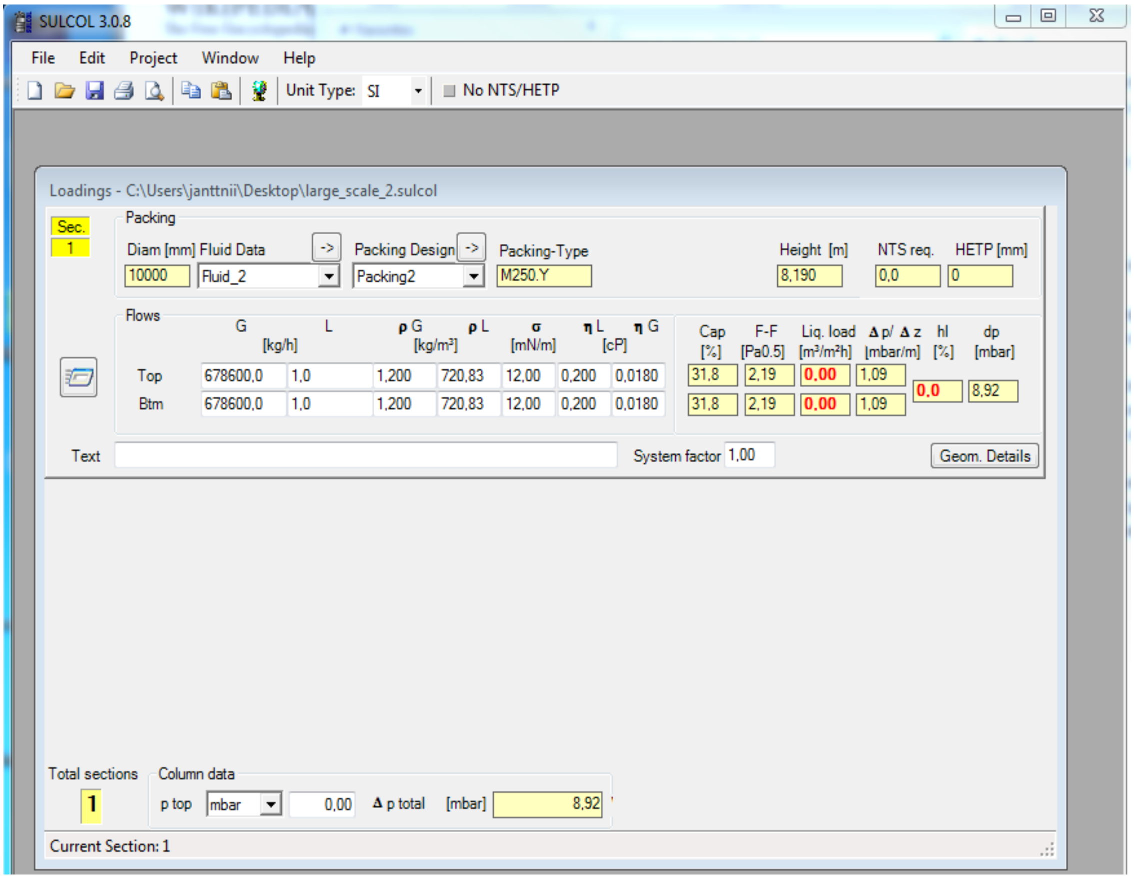
```
// 0 : only write final meshes
// 1 : write intermediate meshes
// 2 : write volScalarField with cellLevel for postprocessing
// 4 : write current intersections as .obj files
debug 2;

// Merge tolerance. Is fraction of overall bounding box of initial mesh.
// Note: the write tolerance needs to be higher than this.
mergeTolerance 1e-6;

// ************************************************************************* //
```

G Appendix: SULCOL

The pressure drop computation with a SULCOL 3.0.8 sizing program for a column with a round cross-section and a diameter of 10 m. The mass flux is calculated with an inlet velocity of 2 m/s.



Unit Type: SI

☐ No NTS/HETP

Loadings - C:\Users\janttnii\Desktop\large_scale_2.sulcol

Sec. 1

Packing

Diam [mm]	Fluid Data	Packing Design	Packing-Type	Height [m]	NTS req.	HETP [mm]
10000	Fluid_2	Packing2	M250.Y	8,190	0,0	0

Flows

	G [kg/h]	L	ρ G [kg/m³]	ρ L	σ [mN/m]	η L [cP]	η G	Cap [%]	F-F [Pa0.5]	Liq. load [m³/m²h]	Δ p/ Δ z [mbar/m]	hl [%]	dp [mbar]
Top	678600,0	1,0	1,200	720,83	12,00	0,200	0,0180	31,8	2,19	0,00	1,09	0,0	8,92
Btm	678600,0	1,0	1,200	720,83	12,00	0,200	0,0180	31,8	2,19	0,00	1,09		

Text

System factor 1,00

Total sections 1

Column data

p top mbar 0,00 Δ p total [mbar] 8,92

Current Section: 1