PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

MARCOS BATISTELLA LOPES

NUMERICAL MODELLING OF PARTICLE DISPERSION AND DEPOSITION IN VENTILATION DUCT BENDS

CURITIBA
Dezembro – 2019

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

MARCOS BATISTELLA LOPES

NUMERICAL MODELLING OF PARTICLE DISPERSION AND DEPOSITION IN VENTILATION DUCT BENDS

Thesis submitted to the "Programa de Pós-Graduação em Engenharia Mecânica", "Escola Politécnica", "Pontifícia Universidade Católica do Paraná", in partial fulfillment of the requirements for the degree of Doctor in Mechanical Engineering

Advisor: Viviana Cocco Mariani
Claudine Béghein
Kátia Cordeiro Mendonça

CURITIBA
Dezembro – 2019

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Luci Eduarda Wielganczuk – CRB 9/1118

Lopes, Marcos Batistella
L864n Numerical modelling of particle dispersion and deposition in ventilation duct
2019 bends / Marcos Batistella Lopes ; advisor: Viviana Cocco Mariani, Claudine Beghein, Kátia Cordeiro Mendonça. – 2019.
217 f. f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2019
Bibliografia: f. 155-164

1. Turbulência. 2. Partículas. 3. Modelos matemáticos. 4. Langrange, Funções de. I. Mariani, Viviana Cocco. II. Beghein, Claudine. III. Mendonça, Kátia Cordeiro. IV. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia Mecânica. V. Título.

CDD 20. ed. 532.0527

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Engenharia Mecânica

PUCPR

TERMO DE APROVAÇÃO

Marcos Batistella Lopes

Numerical Modelling of Particle Deposition and Resuspension in Ventilation Duct Bends

Tese aprovada como requisito parcial para obtenção do grau de doutor no Curso de Doutorado em Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, da Escola Politécnica da Pontifícia Universidade Católica do Paraná, pela seguinte banca examinadora:

Prof. Dr. Silvio Luiz de Mello Junqueira
Universidade Tecnológica Federal do Paraná, UTFPR

Prof. Dr. Luís Mauro Moura
Pontifícia Universidade Católica do Paraná, PUCPR

Prof.ª Drª Kátia Cordeiro Mendonça
Laboratoire d'Innovation Numérique, CESI École d'Ingénieurs, França, Orientadora

Prof.ª Drª Claudine Béghein
Laboratoire des Sciences de l'Ingénieur pour l'Environnement, França, Orientadora

Presidente: **Prof.ª Drª Viviana Cocco Mariani**
Pontifícia Universidade Católica do Paraná, PUCPR, Orientadora

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
PUCPR

Curitiba, 03 de dezembro de 2019

Rua Imaculada Conceição, 1155, Prado Velho, CEP 80215-901, Curitiba, Paraná, Brasil
Tel.: (41) 3271-1585 ou (41) 3271-1562, www.pucpr.br/pgem

To Edna, Marcos and Ane, my eternal love.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Prof. Dra. Viviana Cocco Mariani, Prof. Dra. Kátia Cordeiro Mendonça and Professor Claudine Béghein for their paramount contribution to this work, understanding, patience, secure guidance, and exchange of ideas about several topics, most of them pertaining to this work.

I am most grateful to all professors of the PPGEM and LaSIE. Many thanks should be also given to Mrs. Jane Marques da Rocha, secretary of PPGEM, for her dedication and helpful.

I would like to thank my colleagues of PPGEM and LaSIE, special mention to Emerson Hochsteiner Vasconcelos Segundo, Paulo Paneque Galuzio, Marcelo Borges do Santos, Carlos Freire, Mayara Calado, Claudia Bittencourt, Ana Masur, Mamadou Aliou Diallo, Ali Tfayli, Ahmed Benabed, Ahmad Deeb, and Rama Ayoub.

I would like to express my special thanks to my friends from the begin of this journey, Santiago Castilho Riquelme and Clebe Vitorino Junior; my friends from the neighbor laboratory, Adiren Gros, Claudia Morishita, Ana Rocha, Walter Mazuroski, Lorena Freitas Dutra, Suelen Gasparin; my friends from La Rochelle, Aimn Matr, Lionel Houssou, and Daniel Fiorito; my Brazilian friends who spent a good time in La Rochelle with me, Anastácio da Silva Junior and Marcos Garcia.

I would also like to thank my colleagues, professors, and workers, of PUCPR and Université de La Rochelle for their support and friendship. Since to nominate every one of them would be impossible for my unreliable memory, I beg them to please fill properly and individually acknowledge. Thanks should also be given to my undergraduate students of PIBIC, from whom I continuously learn, for their understanding.

I would like to also express my gratitude to my parents, Edna and Marcos, my friends, Anthony, Gilberto, Thiago, Alexandre, Diego, to my several relatives, and especially including my deceased grandfather Hélio Batistella.

Last, but not least, my uttermost gratitude to my beloved wife Ane Elise Brandalise Gonçalves. Your unconditional love, continuous support, and eternal patience were fundamental to this accomplishment.

Finally, my sincere thanks to those not expressly mentioned, but to whom I will be forever in debt.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Love is wise, hatred is foolish.

Bertrand Russell, 1959.

ABSTRACT

This study aims to investigate the numerical modelling of the dilute particle-laden turbulent airflow within elbows typical of HVAC (Heating, Ventilation, and Air-conditioning) applications. The objective is to evaluate the influence of some parameters of air-conditioning and mechanical ventilation systems such as the shape on the dispersion and deposition of solid particles. The numerical study was conducted with the OpenFOAM® computational code using an Eulerian-Lagrangian approach with modified particle-wall interaction models. In the first study, the turbulent airflow (without the dispersed phase) through a duct bend was investigated with eight Reynolds Averaged Navier-Stokes (RANS) turbulence models to evaluate their capability to describe the behavior of the airflow in elbows in terms of mean velocity, turbulence intensities, and pressure field. Among the turbulence models tested, the more accurate were the following: the k-ε of Launder and Sharma model, the SST k-ω model, the v^2f model, and the RSM SSG model. The second study was the evaluation of four selected RANS turbulence models combined with four Particle-Wall Interactions (PWI) of the Lagrangian Particle Tracking (LPT) algorithm to predict the particle dispersion mostly in terms of the mean velocities. The Brauer's and Brach & Dunn's PWI models produced the lowest errors among the RANS-LPT/PWI combinations tested. Finally, the third study was the investigation of the influence of duct shape on the aerosol dispersion and deposition in HVAC elbows. For Stokes numbers ranging from 0.4 to 3.0, the circular elbow collection efficiency was higher than the square duct bend. Overall, all the studies followed a proposed systematic methodology of verification and validation (V&V), and the findings agreed quite well with the current literature. Besides, this work implemented new features regarding the PWI in the OpenFOAM® Lagrangian library.

Keywords: CFD; turbulence; particle deposition; particle dispersion; Eulerian-Lagrangian approach; HVAC elbow; OpenFOAM®.

RÉSUMÉ

Cette étude propose l'application de la simulation numérique pour prévoir l'écoulement turbulent gaz-solide dilué à l'intérieur d'un conduit courbé d'un système CVC (Chauffage, Ventilation et Climatisation). Les principaux objectifs sont d'obtenir des résultats fiables et d'évaluer l'importance de certains paramètres tel que la section transversale par rapport au dépôt de particules. Les simulations numériques sont été effectuées avec le code OpenFOAM® avec une formulation Eulerienne-Lagrangienne et des modèles d'interaction particule-paroit modifiés. Le premier cas d'étude concerne la prédiction de l'écoulement turbulant de l'air (sans la phase dispersée) à l'intérieur d'un conduit courbé avec des modèles RANS (*Reynolds Averaged Navier-Stokes*) pour le champs de vitesse moyen, les intensités turbulentes et le champs de pression. Parmi les modèles RANS testés, le plus précis ont été les suivants: le modèle k-ε de Launder et Sharma, le modèle SST k-ω, le modèle v^2f et le modèle RSM SSG. Le deuxième cas d'étude concerne à l'évaluation de quatre modèles RANS sélectionnés et combinés avec quatre modèle d'interaction particule-paroi (PWI – *Particle-Wall Interaction*) de la méthode de Lagrange pour suivre les particules (LPT – *Lagrangian Particle Tracking*) et prédire la dispersion of particules par rapport à la vitesse moyenne. Les modèles d'interaction particule-paroi de Brauer et Brach & Dunn ont obtenu les petits erreurs parmi les combinaisons RANS-LPT/PWI testés. Finalement, le troisième cas d'étude concerne à l'étude de l'influence de la section transversale par raport à la dispersion et au dépôt de particules à l'intérieur des conduits courbés des sytèmes CVC. Pour les nombres de Stokes de 0.4 à 3.0, le dépôt de particules a été supérieur dans le conduit courbé circulaire par rapport le conduit courbé carré. En général, tous les études proposés ont suivi à une méthodologie systématique de vérification et validation (V&V), et les résultats sont d'accord avec the la littérature actuelle. Par ailleurs, cette thèse a implementé de nouvelles fonctionnalités dans la bibliothèque Lagrangienne de l'OpenFOAM®.

Mots-clés: mécanique des fluides numérique; turbulence; dépôt de particules; dispersion de particules; méthode Eulérienne-Lagrangienne; conduit courbé d'application en CVC; OpenFOAM®.

RESUMO

Apresenta-se um estudo que utiliza a dinâmica dos fluidos computacional (CFD – *Computational Fluid Dynamics*) para investigar o escoamento turbulento de uma mistura bifásica gás-sólido diluída no interior de cotovelos de ventilação. O principal objetivo desta tese é avaliar a influência de alguns parâmetros de dutos curvos de ventilação como a seção transversal na dispersão e deposição de partículas sólidas no interior destes dutos. Para tanto, este estudo numérico utiliza algumas das bibliotecas do OpenFOAM® para modelos Euleriano-Lagrangiano com modificações nos modelos de interação entre as partículas e as paredes do duto. Num primeiro caso de estudo, o escoamento de ar turbulento (sem a presença da fase dispersa) em um duto curvo foi investigado com oito modelos RANS (*Reynolds Averaged Navier-Stokes*) para avaliar a capacidade deles em predizer o comportamento do escoamento em dutos curvos em termos da velocidade média, intensidade turbulenta e campos de pressão. Entre os modelos de turbulência testados, os mais precisos foram: o modelo k-ε de Launder e Sharma, o modelo SST k-ω, o modelo v^2f e o modelo RSM SSG. O segundo caso de estudo consiste na combinação de quatro modelos RANS do estudo anterior com quatro modelos de interação entre as partículas e as paredes do duto (PWI – *Particle-Wall Interaction*) aplicados em conjunto com o modelo Lagrangiano (LPT – *Lagrangian Particle Tracking*) para predizer o transporte de partículas especialmente em termos da velocidade média em posições definidas na curva. Os modelos de iteração entre a partícula e parede do duto de Brauer e Brach & Dunn apresentaram os menores erros entre as combinações RANS-LPT/PWI testados. Finalmente, o terceiro caso de estudo investiga a influência da forma do duto na dispersão e deposição de aerossóis nos cotovelos de ventilação. Para números de Stokes entre 0.4 e 3.0, as taxas de deposição no cotovelo do duto circular foram superiores ao duto quadrado. No geral, todos os estudos propostos seguiram uma metodologia rigorosa de verificação e validação (V&V), sendo que os resultados apresentados confirmam os dados encontrados na literatura. Além disso, esta pesquisa implementou novos modelos de interação entre a partícula e a parede do duto nas bibliotecas disponíveis do modelo Lagrangiano da instalação padrão do código OpenFOAM®.

Palavras chave: dinâmica dos fluidos computacional; turbulência; deposição de partículas; dispersão de partículas; modelo Euleriano-Lagrangiano; cotovelo de ventilação; OpenFOAM®.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Roman

a	Cross-section vertical length	[m]
A	Duct cross-section area	[m^2]
A_p	Cross-section area of the particle	[m^2]
b	Cross-section horizontal length	[m]
C	Average concentration of particles in the flow	[kgm^{-3}]
C_c	Stokes-Cunningham slip correction factor	[-]
C_d	Drag coefficient	[-]
C_i	Particle concentration at bend inlet	[kgm^{-3}]
C_l	Lift coefficient	[-]
C_o	Particle concentration at bend outlet	[kgm^{-3}]
C_p	Pressure coefficient	[-]
C_∞	Free-stream airborne concentration	[kgm^{-3}]
C^+	Dimensionless particle concentration	[-]
d_{ij}	Diffusion rate	[m^2s^{-3}]
d_p	Particle diameter	[m]
D	Circular duct diameter	[m]
D_B	Brownian diffusivity	[m^2s^{-1}]
D_h	Hydraulic diameter	[m]
D_t	Eddy diffusivity	[m^2s^{-1}]
De	Dean number	[-]
e	Coefficient of restitution	[-]
e_n	“Normal” coefficient of restitution	[-]
e_t	“Tangential” coefficient of restitution	[-]
E	Surface potential energy	[J]
E_r	Relative error	[-]
f	Darcy-Weisbach friction factor	[-]
f_d	Dynamic friction velocity	[-]
F_d	Drag force	[N]
F_g	Gravity force	[N]

F_l	Shear Saffman and Mei lift force	[N]
$F_{l(Saff)}$	Shear Saffman lift force	[N]
F_p	Particle forces	[N]
F_B	Brownian force	[N]
g	Gravity acceleration	[ms^{-2}]
G	Shear rate	[s^{-1}]
$Gö$	Görtler number	[-]
GCI	Grid Convergence Index	[-]
h_l	Head loss	[m]
h_p	Particle distance from the wall	[m]
H	Hamaker constant	[J]
I_p	Particle momentum of inertia	[kgm^2]
J	Net aerosol flux	[$kgm^{-2}s^{-1}$]
$\vec{J}$	Linear impulse	[Ns]
k	Turbulent kinetic energy	[m^2s^{-2}]
k_{pw}	Dimensionless distance of the particle from the wall	[-]
k_B	Boltzmann constant	[JK^{-1}]
K	Constant	[-]
K_b	Bend loss coefficient	[-]
Kn	Knudsen number	[-]
l	Characteristic length	[m]
l_t	Turbulence length scale	[m]
L	Duct length	[m]
L_e	Entrance length	[m]
L_G	Saffman length scale	[m]
L_S	Stokes length scale	[m]
L_1	Upstream straight duct length	[m]
L_2	Downstream straight duct length	[m]
m_p	Particle mass	[kg]
N	Grid size	[-]
N_p	Number of particles	[-]
N_1	Upstream duct multiplier factor	[-]

N_2	Downstream duct multiplier factor	[-]
p	Pressure	[Pa]
p_m	Numerical scheme order of accuracy	[-]
P	Penetration	[-]
P_{ij}	Generation rate	[m^2s^{-3}]
P_k	Production of turbulent kinetic energy	[m^2s^{-3}]
P_w	Wetted perimeter normal to the principal flow direction	[m]
r	Radius position between the inner and outer bend wall	[m]
r_m	Mesh refinement	[-]
r^*	Relative radius position	[-]
R	Radius of the wall curvature	[m]
R_b	Bend radius	[m]
R_i	Inner bend radius	[m]
R_o	Outer bend radius	[m]
Re	Reynolds number of fluid	[-]
Re_p	Reynolds number of particle	[-]
Re_t	Turbulent Reynolds number	[-]
S	Particle stop distance	[m]
S_{ij}	Mean strain rate	[s^{-1}]
Sc	Schmidt number	[-]
St	Stokes number	[-]
S^+	Dimensionless particle stop distance	[-]
S_0	Spectral density	[s^{-1}]
t	Time	[s]
T	Temperature	[K]
T_i	Turbulence intensity	[-]
T_p	Torque of the particle	[Nm]
u	Characteristic velocity	[ms^{-1}]
u_c	Critical particle velocity	[ms^{-1}]
${u_n}^0$	Normal particle approach velocity	[ms^{-1}]
u_n	Normal particle velocity after the collision	[ms^{-1}]
${u_t}^0$	Tangential particle approach velocity	[ms^{-1}]

u_t	Tangential particle velocity after the collision	[ms^{-1}]
u_p	Particle velocity	[ms^{-1}]
$u_{p,r}$	Relative velocity of the fluid with respect to the particle	[ms^{-1}]
$u_{p,s}$	Streamwise local relative particle velocity	[ms^{-1}]
u_τ	Friction velocity	[ms^{-1}]
u^+	Non-dimensional velocity	[-]
$\overline{u_i u_j}$	Reynolds stress tensor	[m^{2}s^{-2}]
U	Mean velocity	[ms^{-1}]
U_{ave}	Average flow velocity	[ms^{-1}]
U_∞	Free stream velocity	[ms^{-1}]
V	Fluid volume	[m^3]
V_d	Deposition velocity	[ms^{-1}]
V_p	Particle volume	[m^3]
${V_d}^+$	Dimensionless deposition velocity	[-]
V_p	Particle volume	[m^3]
x	Spatial coordinate *x*	[m]
y	Spatial coordinate *y*	[m]
y_n	Normal distance from the wall	[m]
y_0	Equilibrium separation distance	[m]
y^+	Dimensionless normal wall distance	[-]
z	Spatial coordinate *z*	[m]

Greek

α	Dimensionless shear rate	[-]
α_b	Bend angle	[rad]
α_p	Volume fraction of the dispersed phase	[-]
α_A	Particle approach angle	[rad]
${\alpha_A}'$	Modified particle impact approach angle	[rad]
δ	Curvature ratio	[-]
δ_{ij}	Kronecker delta	[-]
δ_{vs}	Thickness of viscous sublayer	[m]
Δp	Pressure loss	[Pa]

Δt	Time step	[s]
Δx	Minimum grid size	[m]
ε	Dissipation rate of the turbulent kinetic energy	[m^2s^{-3}]
ε_{ij}	Process modeled in terms of turbulence variables	[m^2s^{-3}]
η	Deposition efficiency	[-]
θ	Momentum thickness	[m]
λ	Mean free path of air molecules	[m]
μ	Dynamic viscosity of fluid	[$kgm^{-1}s^{-1}$]
ν	Kinematic viscosity of fluid	[m^2s^{-1}]
ν_t	Turbulent kinematic viscosity	[m^2s^{-1}]
ζ	Zero-mean, unit-variance-independent Gaussian random number	[-]
ρ	Density of fluid	[kgm^{-3}]
ρ_p	Density of particle	[kgm^{-3}]
τ_e	Eddy lifetime	[s]
τ_{ij}	Reynolds stress tensor	[Jm^{-3}]
τ_k	Kolmogorov time scale	[s]
τ_p	Particle relaxation time	[s]
$\tau_p{}^+$	Dimensionless relaxation time	[-]
τ_r	Time for the particle pass through the eddy	[s]
τ_w	Wall shear stress	[Pa]
τ_F	Fluid characteristic time	[s]
ϕ	Generic property	[ ]
ϕ_b	Cross-section bend angle	[rad]
ϕ_{ij}	Process modeled in terms of turbulence variables	[m^2s^{-3}]
ω	Turbulent frequency	[Hz]
ω_p	Rotational particle velocity	[$rads^{-1}$]
Ω_{ij}	Mean vorticity	[s^{-1}]

Subscript**s**

p	Particle
t	Turbulent
0	Initial value

NOMENCLATURE

BBO	Boussinesq, Basset and Oseen
CFD	Computational Fluid Dynamics
CFPD	Computational Fluid and Particle Dynamics
CFWN	Continuous Filtered White Noise
DEM	Discrete Element Method
DES	Detached Eddy Simulation
DIM	Diffusion-Inertia Model
DNS	Direct Numerical Simulation
DRW	Discrete Random Walk
DSMC	Discrete Simulation Monte Carlo
EVM	Eddy Viscosity Model
EWT	Enhanced Wall Treatment
FDM	Finite Difference Method
FVM	Finite Volume Method
GCI	Grid Convergence Index
GNU GPL	General Public License
HVAC	Heat, Ventilation, and Air-Conditioning
HRN	High Reynolds Number
IAQ	Indoor Air Quality
LDV	Laser Doppler Velocimeter
LES	Large Eddy Simulation
LRN	Low Reynolds Number
LRR	Launder, Reece and Rodi
LPT	Lagrangian Particle Tracking
ODE	Ordinary Differential Equation
OpenFOAM	Open source Field Operation And Manipulation
PDE	Partial Differential Equation
PDF	Probability Density Function
PISO	Pressure Implicit with Splitting of Operators
PWI	Particle-Wall Interactions
RANS	Reynolds Averaged Navier-Stokes
RKE	Realizable k-ε
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNG	Re-Normalization Group
RSM	Reynolds Stress Model
SAS	Scale Adaptative Simulation
SIMPLE	Semi-Implicit Method of Pressure-Linked Equations

SIMPLEC	Semi-Implicit Method of Pressure-Linked Equations Consistent
SIMPLER	Semi-Implicit Method of Pressure-Linked Equations Revised
SATP	Standard Ambient Temperature and Pressure
SSG	Speziale, Sarkar and Gatski
SST	Shear Stress Transport
V&V	Verification and Validation
VOF	Volume of fluid
WHO	World Health Organization
2D	Two-dimensional
3D	Three-dimensional

CONTENTS

1 INTRODUCTION

Many people spend a large amount of their time in building environment and are exposed to a range of air contaminants, such as dust, aerosol particles, fumes and mists (COLBECK and LAZARIDIS, 2014). Aerosols are a suspension of liquid or solid in gas and have an impact on Indoor Air Quality (IAQ). Many sources of particle generation are found in nature (e.g., virus, bacteria, pollen), in the industrial process (e.g., chemical reaction, manufacturing) and even in the human body (e.g., skin flakes, hair, respiratory emissions). The order of magnitude of the size of these particles varies from 10^{-3} to 10^{2} μm (ASHRAE, 2017). Particles larger than 40 μm normally do not remain suspended in the air for a sufficient amount of time (TU *et al.*, 2013) and the particles less than 2.5 μm can get deep into the lungs and cause serious health problems (INTERNATIONAL ENERGY AGENCY, 2016).

Air-conditioning systems are supposed to provide a thermally comfortable and healthy environment for the occupants with low consumption of power electricity. Since the 1980s the World Health Organization (WHO) has set reference values for a good internal air quality (PENNEY *et al.*, 2010) and many countries follow these guidelines to create their own regulations concerning air-conditioning systems, such as Brazil and France (the countries where this study was developed) as described in Table 1.1. Despite that, there is still a lot of research and discussion regarding this subject in both countries to improve these regulations and, consequently, to allow the reduction of health problems associated with the air quality in conditioned spaces.

Table 1.1 – Indoor air quality control in Brazil and France.

Country	Regulation	Year	Purpose
France	Law N° 2008-757	2008	Environment responsibilities.
	Decree N° 2010-349	2010	Inspection of the air-conditioning system higher than 12 kW every 5 years.
Brazil	Statute N° 3523 Ministry of Health	1998	Technical regulation of air-conditioning system cleaning.
	RE/ANVISA N° 9	2003	Air quality standard for the air-conditioning environment.
	ABNT NBR 16401-3	2008	Establish the basic conditions and minimum requirements to obtain acceptable IAQ.

In mechanically ventilated and conditioned buildings, the level of particle concentration within the build environment can be affected by the duct system. Air pollution occurs in both outdoor and indoor spaces. When the air is introduced into the building through a duct system, the suspended contaminants not filtered can deposit on or can resuspend from the duct walls. If deposition can contribute to reducing the degree of pollution entering the indoor space, resuspension of particles may expose building occupants to high levels of particle concentration (MIGUEL *et al.,* 2004). Therefore, it is important to understand and quantify deposition and resuspension in the duct system in order to evaluate accurately human exposure to particles within mechanically ventilated and conditioned buildings.

HVAC distribution systems are composed of straight ducts and fittings such as inlets, elbows, tees, etc. As pointed out by SIPPOLA and NAZAROFF (2005) deposition, and consequently resuspension, is less pronounced on straight ducts with fully developed turbulent flow profiles than on the fittings where the turbulent flow is disturbed. Predicting the already complex two-phase flow field in the fittings is then further complicated because secondary flows appear, and the flow can be detached from the solid surfaces.

Particle deposition and resuspension can be predicted by experiments and numerical simulations. Experiments are essential for engineering applications; however, they are expensive and time-consuming. Advances in computing technology over the past years have allowed codes of Computational Fluid Dynamics (CFD) to make it possible to reach numerically approximate solutions for complex engineering problems. Indeed, numerical simulation is helpful for cases where experiments are impossible to make with the current technology, and it is a useful tool for parametric studies.

The present study was carried out using CFD techniques, as previous works have shown that CFD is an efficient and reliable tool to investigate particle distribution and deposition in curved ducts (SUN *et al.*, 2011; GAO and LI, 2012; SUN *et al.*, 2012). Specifically, the study employs the Eulerian-Lagrangian approach, where the fluid phase is modeled as a continuum and the particles are tracked throughout the flow. In spite of the Lagrangian approach being computationally time-consuming (OESTERLÉ, 2006), it was chosen for particle tracking because it enables to model the particle interaction with surfaces more easily than with the Eulerian approach (CROWE *et al.*, 2012) and for HVAC applications aerosols are dilute.

Special attention has been given to the choice of the appropriate models to describe the turbulent airflow (without the disperse phase) through the duct bend, and the interactions between the particulate phase and the solid surfaces of the duct. In this way, four Particle-Wall

Interactions (PWI) integrated into the Lagrangian Particle Tracking (LPT) model were investigated regarding their capabilities to predict the particle dispersion within elbows, considering Reynolds Averaged Navier-Stokes (RANS) turbulence models previously selected.

The objective of this study is to identify the most accurate turbulence – LPT/PWI model combination to predict diluted flow in 90° bends and the impact of the PWI on the accuracy of the flow description. In order to reach this goal, the capabilities of sixteen couples of RANS-LPT/PWI models to predict particle-laden flow in a 90° bend were evaluated, qualitatively and quantitatively, by comparing their 3D and 2D simulations results of mean streamwise velocities to experimental and numerical data available in the literature. The sixteen RANS-LPT/PWI models tested in this work are issued from the combination of:

- four turbulence models:
  - the low Reynolds number k-ε model (LAUNDER and SHARMA, 1974);
  - the SST k-ω model (MENTER, 1992);
  - the *v2-f* model (DAVIDSON *et al.*, 2003);
  - the RSM SSG (SPEZIALE *et al.*, 1991).
- with four PWI models from:
  - Brauer (BRAUER, 1980);
  - Grant & Tabakoff (GRANT and TABAKOFF, 1975);
  - Matsumoto & Saito (MATSUMOTO and SAITO, 1970);
  - Brach & Dunn (BRACH and DUNN, 1998).

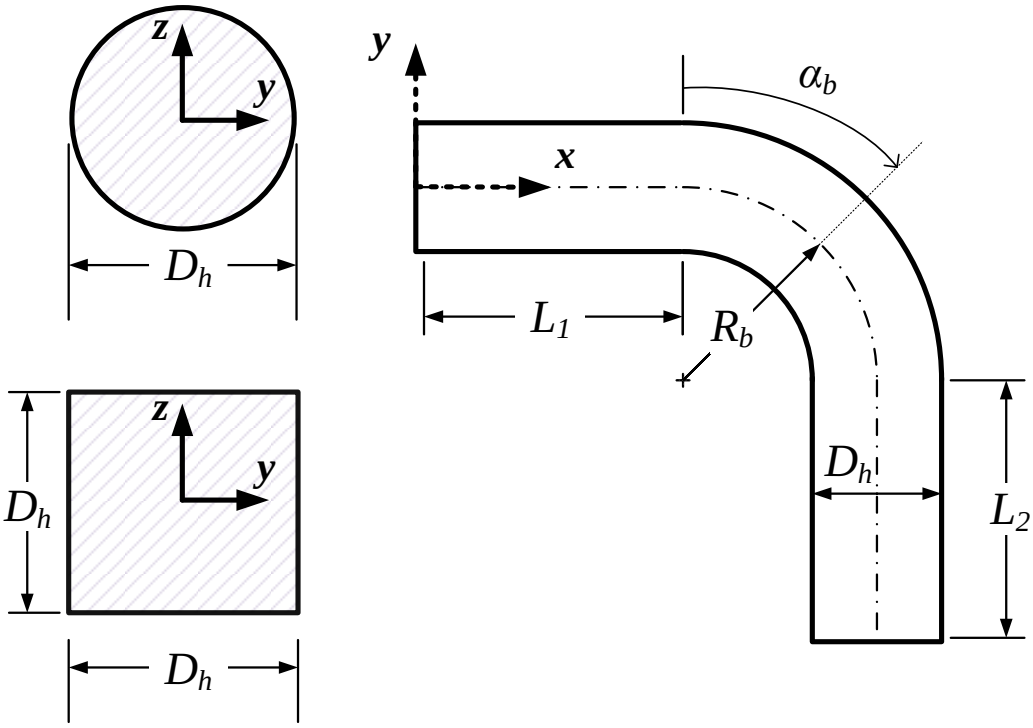
The purpose of this thesis is then to show, via numerical predictions of particle-laden turbulent flows in curved ducts, that the impact of the RANS-LPT/PWI combined model on the airflow and particle dispersion in curved ducts, and the effects of different cross-sections on particle deposition.

1.1 THESIS STRUCTURE

The present thesis is organized as follows. First, following this introduction, Chapter 2 presents the principal fundamentals and definitions of turbulent flows and particle dispersion in curved pipes. In the sequence, Chapter 3 presents an extensive literature review about particle deposition in ducts and duct bends to claim the principal contributions of this study, while

Chapter 4 outlines the physical and mathematical models adopted and reviews some concepts of particle turbulent transport theory. In Chapter 5 the principal findings of this study are revealed into three parts: the first part studied the capability of eight turbulence models to predict the airflow through a circular duct, the second part combined four turbulence models (among the turbulence models of the first part) with four rebound models to predict the mean velocity in a gas-solid flow within a square duct bend, and the third section chose one turbulence model (among the turbulence models of the second part) to study the particle deposition in circular and square duct bends; yielding a total of three geometries of 90° duct bend such as shown in Figure 1.1. Chapter 6 offers a conclusion and recommendations for the remaining work.

Figure 1.1 – Geometries of this study.



2 FUNDAMENTALS AND DEFINITIONS

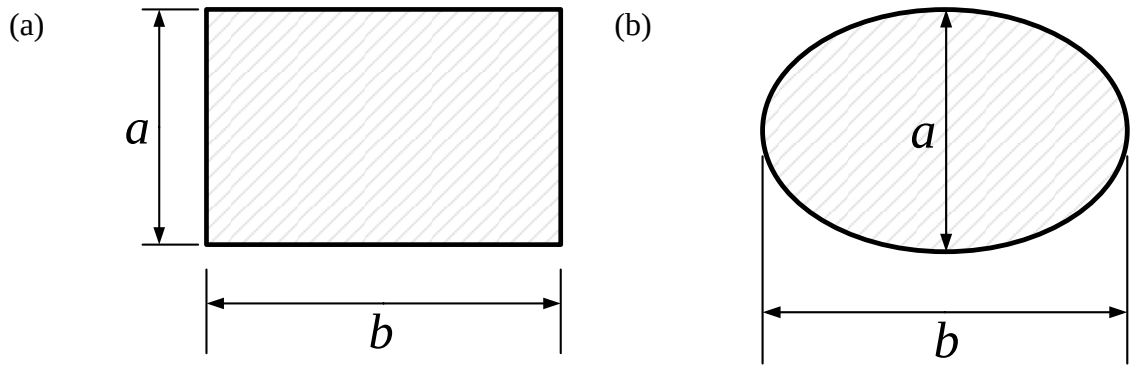
The objective of this chapter is to provide a background in the fundamentals and definitions of the relevant geometrical and physical parameters applied in this dissertation. First, the main properties concerning the curved pipe flow are described. Second, the main properties regarding particle transport and deposition are presented.

2.1 CURVED PIPE FLOW

Since the first observations reported of the wall curvature effects on wall-bounded flows (THOMSON, 1876) and theoretical analysis of the solution of the laminar curved pipe flow (DEAN, 1928) important contributions have been developed such as the bend coefficient loss (ITŌ, 1960). This section focuses on HVAC elbows. A comprehensive review of flow in curved pipes can be found in BERGER and TALBOT (1983) and SPEDDING *et al.* (2004).

The noncircular ducts are general cases of ventilation duct systems and they are given in Figure 2.1. When $a = b$ we have two cases: the square and circular cross-sections.

Figure 2.1 – Duct cross section: (a) rectangular and (b) elliptical.



For the case of noncircular duct, it is necessary to calculate the hydraulic diameter defined as (WHITE, 2010)[1]:

$$D_h = \frac{4A}{P_w}, \tag{2.1}$$

[1] The hydraulic diameter may be used to determine the Reynolds number, friction factor and pressure loss (KUEHN *et al.*, 1998).

where A is the duct cross-section area and P_w is the wetted perimeter normal to the principal flow direction. Equation (2.1) is not applicable to wide, flat ducts, or ducts of triangular or other irregular shapes because the losses caused by secondary flows increase fast in these geometries (FOX *et al.*, 2011). The hydraulic diameter of the possible cross-sections of this study is presented in Table 2.1 where the lengths a and b are indicated in Figure 2.1.

Table 2.1 – Hydraulic diameter for different geometries.

Cross-section	*Hydraulic diameter*
Circular	$D_h = D$
Square	$D_h = a$
Rectangular	$D_h = \frac{2ab}{a+b}$
Elliptical	$D_h = \frac{2ab\left[64 - 16\left(\frac{b-a}{b+a}\right)^2\right]}{(a+b)\left[64 - 3\left(\frac{b-a}{b+a}\right)^4\right]}$

The curvature ratio, δ, is a very important geometric characteristic of a curved duct, and it is calculated as:

$$\delta = \frac{R_b}{D/2}, \tag{2.2}$$

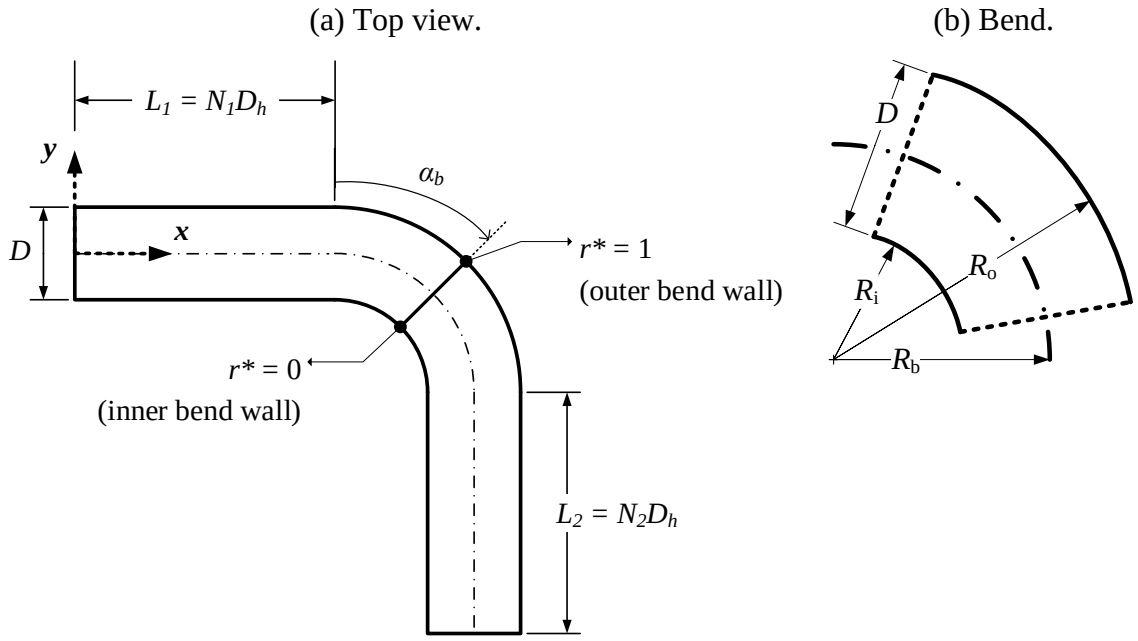
where R_b is the bend radius and D is the distance between the inner and outer bend walls.

A representative top view of a duct bend domain is shown in Figure 2.2. The lengths of the upstream and downstream straight ducts are L_1 and L_2, respectively. These lengths are functions of the hydraulic diameter times the multiplier factors N_1 and N_2 for the upstream and downstream ducts, respectively. The duct bend angle (α_b) is 0° at the bend inlet and 90° at the bend outlet. In the duct bend, the radius can be made non-dimensional as follows:

$$r^* = \frac{r - R_i}{R_o - R_i}, \tag{2.3}$$

where r is the radius position between the inner and the outer bend wall. At the inner bend wall $r^* = 0$, and at the outer bend wall $r^* = 1$.

Figure 2.2 – Representative geometric model of the duct bend: (a) top view and (b) bend.



The Reynolds number is the dimensionless number in fluid mechanics that determines the transition from laminar to turbulent flow, which is defined as:

$$Re = \frac{ul}{\nu}, \tag{2.4}$$

where u is the characteristic velocity, l is the characteristic length and ν is the kinematic viscosity[2]. The Reynolds number represents the ratio between inertial forces and viscous forces. When the Reynolds number in a straight circular pipe is higher than 4,000 (ÇENGEL and CIMBALA, 2006) the flow becomes turbulent as presented in Table 2.2.

Table 2.2 – Transition region from laminar to turbulent flow: circular pipe under practical conditions.

Reynolds number	Type of flow
$Re \lesssim 2{,}300$	Laminar
$2{,}300 \lesssim Re \lesssim 4{,}000$	Transitional
$Re \gtrsim 4{,}000$	Turbulent

In curved ducts, the flow regime can be also to be determined with the Dean number (DEAN, 1928) defined as:

[2] The characteristic velocity in this study is the bulk velocity ($u \equiv U_0$) and the characteristic length is the hydraulic diameter ($l \equiv D_h$).

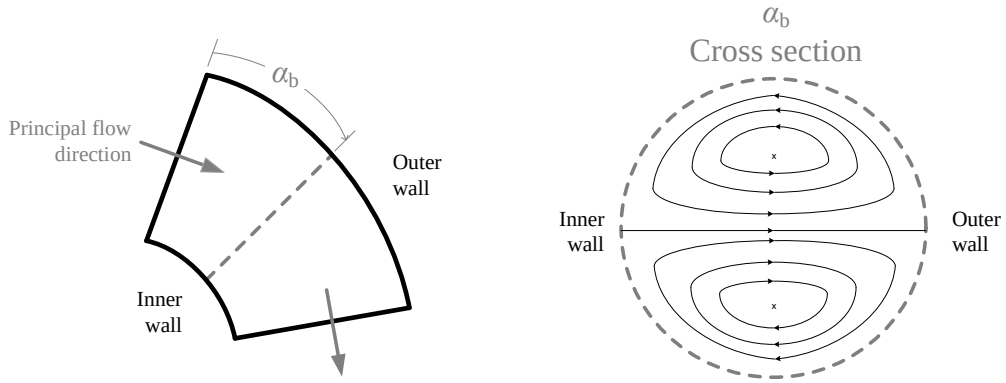
$$De = \frac{Re}{\sqrt{\delta}}, \tag{2.5}$$

where Re is the Reynolds number based on the duct diameter and δ is the curvature ratio. The Dean number is a direct measure of the secondary flow and represents the ratio of the square root of the product between the inertial and centrifugal forces to the viscous forces.

As reported by McFARLAND *et al.* (1997) the laminar-turbulent transition in curved pipe flows occurs for a Dean number approximately equal to 370[3]. For a curvature ratio of 3, the transition to turbulent flow in a duct bend occurs for a Reynolds number of 640, i.e., much less than the usually critical Reynolds number for straight pipe flows.

The Dean vortices are secondary flows induced by the wall curvature as shown in Figure 2.3. They are a pair of counter-rotating vortices that appears due to a change close to the direction of the principal flow. The fluid velocity decreases close to the inner wall and increases in the outer wall of the bend because there is an adverse pressure gradient. This leads to a rise in the secondary motion in which the fluid in the center of the duct is swept towards the outer bend wall and the fluid near the duct wall returns towards the inner bend wall. The Dean vortices also appear in curved pipes laminar flow, however, the axis of the vortex in the turbulent flows is translated towards the bend inner wall and can appear one or more pairs of counter-rotating vortices (ZHANG *et al.*, 2012).

Figure 2.3 – Illustration of Dean vortices in a curved pipe flow.

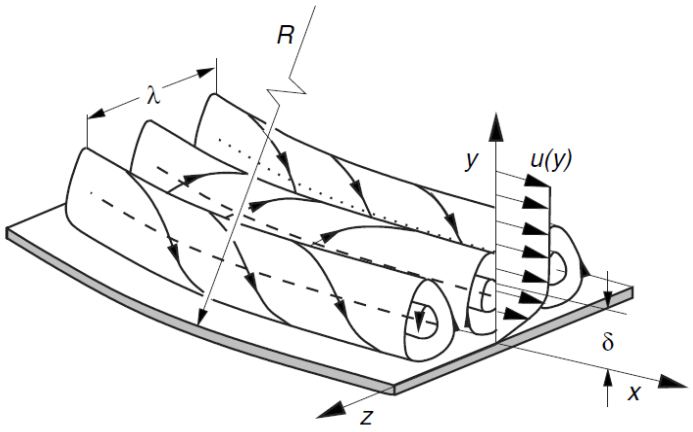


Other kinds of vortices are the boundary-layer instabilities induced by wall curvature. These instabilities are known as the Görtler vortices, being secondary flows that appear in a

[3] According to DENNIS and NG (1982) the critical Dean number is 956. Therefore, there is not an agreement in the literature and we will adopt the critical Dean number of 370.

boundary layer flow along a concave wall (the outer wall of a duct bend). They are counter-rotating vortices aligned in the principal flow direction as presented in Figure 2.4.

Figure 2.4 – Görtler vortices in the boundary layer at a concave wall (the bend outer wall).



Source: SCHLICHTING and GERSTEN (2017).

The Görtler number is defined as (SCHLICHTING and GERSTEN, 2017):

$$G\ddot{o} = \frac{U_\infty \theta}{\nu}\sqrt{\frac{\theta}{R}}, \tag{2.6}$$

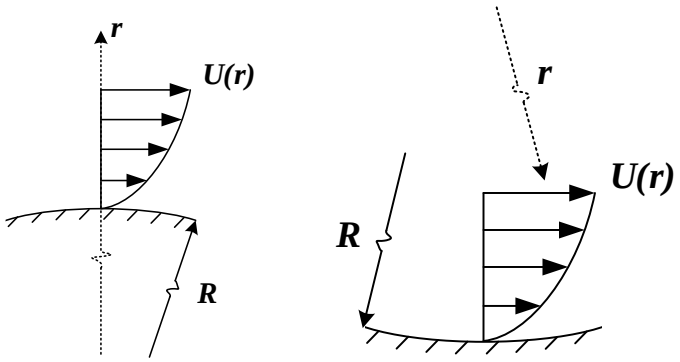
where U_∞ is the free stream velocity, ν is the kinematic viscosity, R is the radius of wall curvature and θ is the momentum thickness given by:

$$\theta = \sqrt{\frac{\nu x}{U_\infty}}, \tag{2.7}$$

where x is a streamwise distance.

If $G\ddot{o} > 0.3$ the streamwise vortices are unstable and grown in the downstream direction (SWEARINGEN and BLACKWELDER, 1987). The Rayleigh circulation criterion (SARIC, 1994) is a necessary condition for centrifugal instabilities, such as the Görtler vortices, to appear along a curved wall. An inner and an outer wall of a duct bend are presented in Figure 2.5. If in the inviscid circular flow, the angular momentum ($|Ur|$) decreases with the increase in r, the flow will be unstable; otherwise, the flow will be stable. In this case, the radial and axial velocity components are assumed to be zero. Only the tangential velocity component varies along the radius of the curved wall.

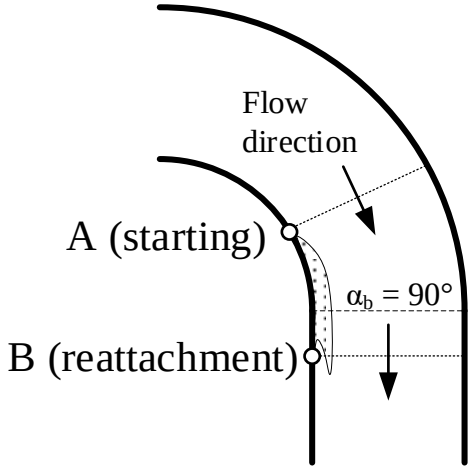
Figure 2.5 – Illustration of velocity distribution in a bend: (a) inner and (b) outer walls.



(a) Stable (b) Unstable

Flow separation occurs near the bend outlet ($\alpha_b = 90°$) along the inner bend wall as presented in Figure 2.6. Point A represents the starting of the flow separation and point B the flow reattachment in the downstream duct. In this region, a flow reversal (backflow) appears near the inner wall causing additional pressure drops.

Figure 2.6 – Flow separation region.



The head loss of a pipe with a constant cross-section and no inclination for the quasi-steady assumption is:

$$h_l = \frac{{U_{ave}}^2}{2g}\left(f\frac{L}{D_h} + K_b\right), \tag{2.8}$$

where h_l is the head loss, U_{ave} is the average flow velocity, g is the gravity acceleration (9.81 m/s^2), f is the Darcy friction coefficient, L is the total length of the straight duct and K_b is the bend loss coefficient given by (WHITE, 2010):

$$K_b = 0.388\left[0.95 + 4.42\left(\frac{\delta}{2}\right)^{-1.96}\right]\left(\frac{\delta}{2}\right)^{0.84} Re^{-0.17}\,. \tag{2.9}$$

The pressure drop (Δp), a consequence of the friction in fluid flow, can be determined from head loss with the following equation:

$$\Delta p = \rho g h_l\,. \tag{2.10}$$

where ρ is the fluid density.

The Darcy friction coefficient can be determined with Prandtl's equation valid for smooth pipes[4]:

$$\frac{1}{\sqrt{f}} = 2log_{10}\left(Re\sqrt{f}\right) - 0.8\,. \tag{2.11}$$

In the near-wall region of turbulent flows (flat plate boundary layer, duct flows, etc.), the viscous phenomena are predominated, and viscous scales can be used in this region. The velocity scale is the friction velocity (u_τ):

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}\,, \tag{2.12}$$

where τ_w is the wall shear stress and the non-dimensional velocity (u^+) is:

$$u^+ = \frac{U}{u_\tau}\,, \tag{2.13}$$

where U is the mean streamwise velocity. The length-scale is the viscous length-scale (δ_{vs}):

$$\delta_{vs} = \frac{\nu}{u_\tau}\,, \tag{2.14}$$

and the non-dimensional distance from the wall (y^+) is:

[4] A numerical method (e.g., the fixed-point method) is necessary to find the Darcy friction coefficient with the Prandtl's equation. A straightforward correlation is the Blasius's equation: $f = (100Re)^{-1/4}$. However, the Blasius's equation is only valid for $Re < 10^5$, while the Prandtl's equation is valid for a larger range of Reynolds numbers. To include the roughness effect, Colebrook's equation may be used instead.

$$y^+ = \frac{y}{\delta_{vs}}\,. \tag{2.15}$$

When calculating particle dispersion in confined flows, attention must be paid to the discretization of the viscous sublayer. The viscous sublayer is located at $y^+ < 5$. Knowing that the wall shear stress is linked to the Darcy friction coefficient according to:

$$\tau_w = \frac{1}{8} f \rho U^2\,, \tag{2.16}$$

one can estimate the grid size of the viscous sublayer. Considering, for instance, a Reynolds number of 10^5, a kinematic viscosity of 1.5×10^{-5} m^2/s (air), and an average velocity of 10 m/s, the thickness of the viscous sublayer will be of the order of 10^{-1} mm and, if a low Reynolds number turbulence model is used, the distance of the first grid to the wall will be of the order of 10^{-2} mm (for such a turbulence model, it is recommended to put the first grid node such that: $y^+ \leq 1$).

The mean stream-wise velocity distribution in a fully turbulent region of a wall-bounded flow is shown in Figure 2.7 where KLEBANOFF (1954) is one of the experimental data available in literature and BIRD *et al.* (2002) gives the curve fitting of LIN *et al.* (1953) for other experimental data summarized in Table 2.3.

Table 2.3 – Dimensionless velocity distribution for fully turbulent flow in circular pipes.

Region	Equation
$0 \leq y^+ \leq 5$ Viscous sublayer	$u^+ = y^+\left[1 - \frac{1}{4}\left(\frac{y^+}{14.5}\right)^3\right]$
$5 < y^+ \leq 30$ Buffer layer	$u^+ = 5ln(y^+ + 0.205) - 3.27$
$y^+ > 30$ Overlap layer	$u^+ = 2.5ln(y^+) + 5.5$

The law of wall given in Table 2.3 with Figure 2.7 provides a sort of universal velocity distribution of the flow in the turbulent boundary layer in the upstream straight pipe. Therefore, whenever possible, we can validate our numerical simulations with this data, observing if the flow condition and duct geometry are like those of the experimental data. Consequently, the numerical simulations will guarantee, at least, a correct velocity distribution profile at the bend inlet.

Figure 2.7 – Stream-wise mean velocity distribution for fully turbulent flows in circular pipes.

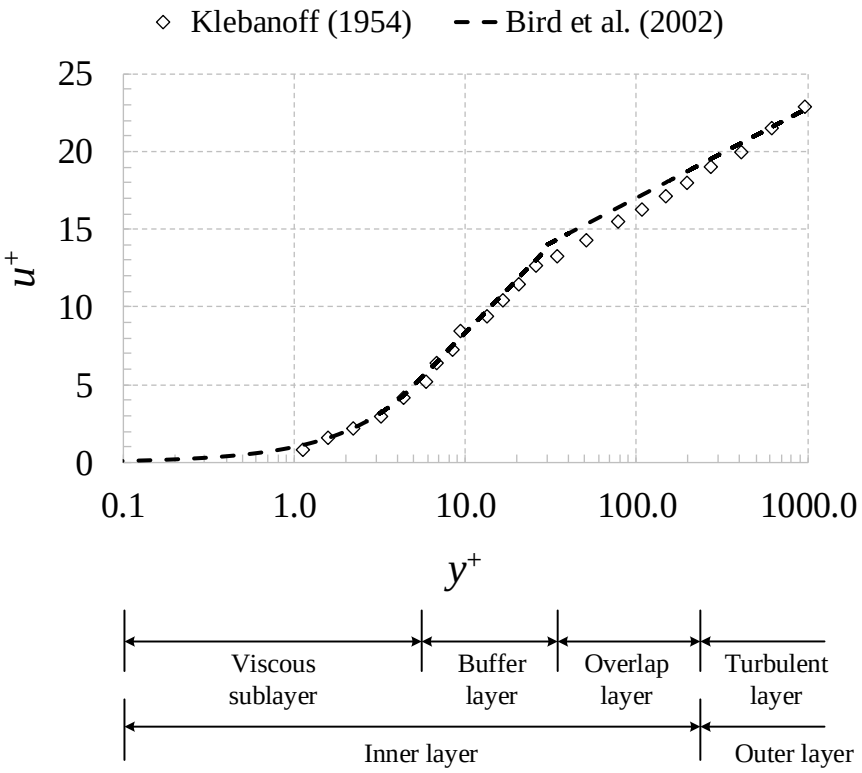


Figure 2.8 depicts the evolution of the pressure coefficient in a 90° bend. This coefficient is calculated as follows:

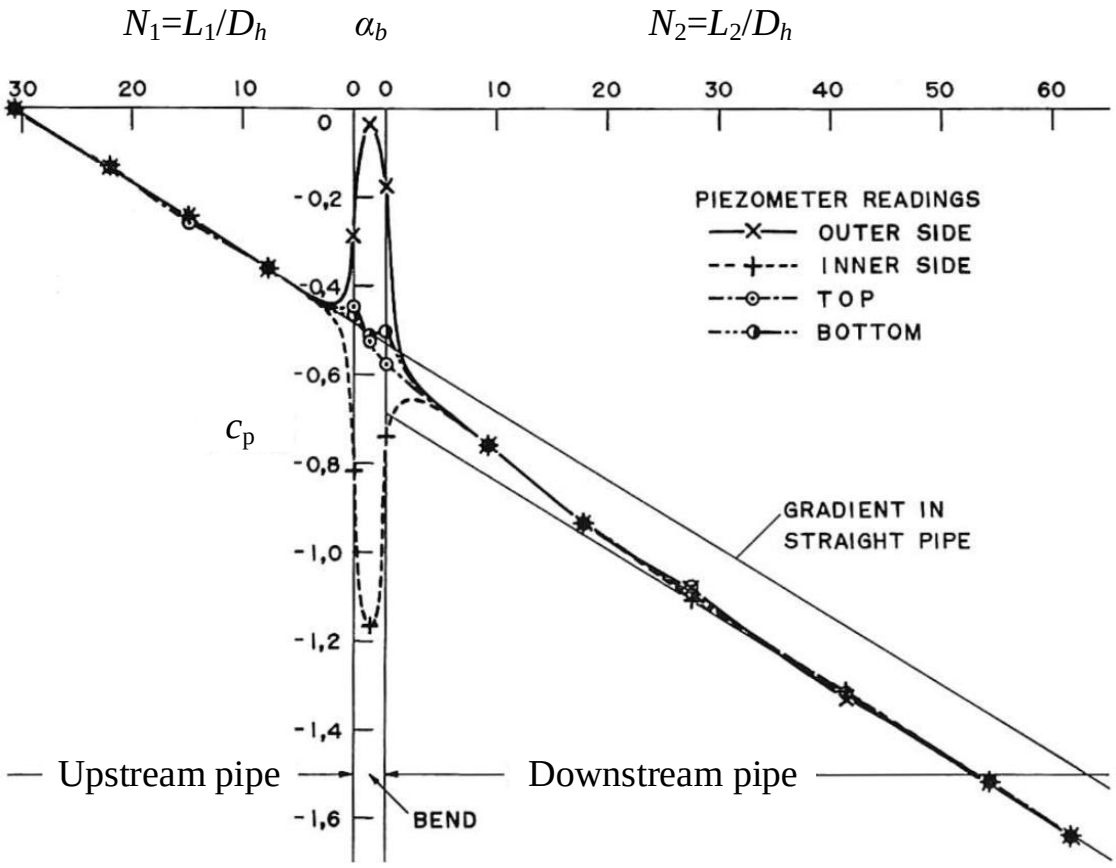
$$C_p = \frac{(p - p_{ref})}{(\rho {U_0}^2/2)_{ref}}, \tag{2.17}$$

where p is the wall pressure at a specific location[5], p_{ref} is the pressure at the same location and at the reference location[6], and the term $\left(\rho {U_0}^2/2\right)_{ref}$ is the dynamic pressure at the reference cross-section.

Along the upstream and the downstream parts of the pipe, the pressure gradient is zero. Near the bend inlet and the bend outlet, an adverse pressure gradient arises, and along the bend, we can see the adverse pressure gradient between the inner bend and outer bend walls. The multiplier factors N_1 and N_2, and the bend angle, α_b, were defined in Figure 2.2. Again, we can validate our numerical simulation against the pressure coefficient curves whenever possible.

[5] The positions are usually: bottom, top, inner and outer walls.
[6] The reference cross section is normally a cross section in the upstream pipe with fully turbulent flow.

Figure 2.8 – Pressure coefficient distribution of a 90° bend with δ = 3.7 and Re = 2×10^5.



Source: adapted from ITŌ (1960).

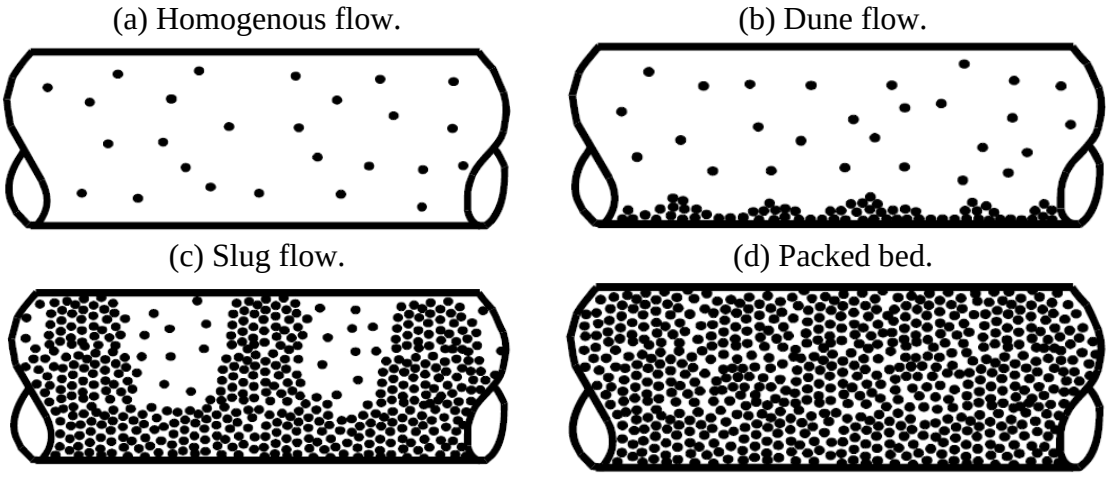
2.2 PARTICLE TRANSPORT AND DEPOSITION

According to CROWE *et al.* (2012) the gas-solid flows, i.e., particles in fluids flows, are a subcategory of multiphase, multicomponent flows. A component is a chemical species (e.g., nitrogen, oxygen, water) and a phase refers to the state of matter (solid, liquid or vapor). The airflow is an example of a single-phase multicomponent flow and the particle-laden flow of fine aerosol particles in the air is an example of a multiphase multicomponent flow.

Multiphase flows can be subdivided into four categories: the gas-liquid flows (e.g., the gas-droplet flow), the gas-solid flows (e.g., the gas-particle flow), the liquid-solid flows (e.g., the sediment transport) and the three-phase flows (e.g., the bubbles in a slurry flow). In this work, we found on the gas-solid flows where the gas (air) is the continuous phase and the particles are the dispersed phase. This category can include for example the pneumatic transport, fluidized beds, etc.

The particle distribution flows encountered in pneumatic transport are presented in Figure 2.9. We can draw some expectations for the case of particle-laden flows in ventilation ducts observing these patterns. In the straight ventilation ducts, we can expect a homogenous flow as illustrated in Figure 2.9a in which the air velocity is sufficiently high that the particles are well mixed and maintained in a nearly homogenous state by turbulence mixing. On the other hand, for other sections of the ventilation duct system, such as the elbows, the particulate-flow patterns will change and we can expect particle concentration such as the dune flow of Figure 2.9b where the particles begin to settle out and collect, for instance, on the outer bend wall because of perturbation of the flow velocity field. The patterns of Figure 2.9c and Figure 2.9d are typical of dense flow and, therefore, they are not ventilation applications.

Figure 2.9 – Horizontal pneumatic transport patterns.



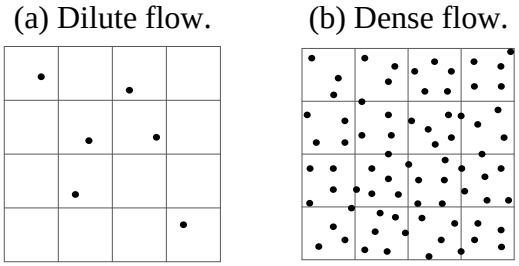
Source: CROWE *et al.* (2012).

As illustrated in Figure 2.10, dispersed flows can be dilute or dense. A dilute dispersed phase flow is one in which the particle motion is controlled by hydrodynamic forces (drag and lift forces). A dense flow is one in which the particle motion is controlled by collision or continuous contact. An indicator for dilute or dense flows is the volume fraction of the dispersed phase, α_p, defined as (MINIER and POZORSKI, 2017):

$$\alpha_p = \frac{N_p V_p}{V}, \tag{2.18}$$

where N_p is the number of particles in the domain, V_p is the volume of one particle and V is the volume of the fluid in the domain. For particle volume fraction of $\alpha_p < 10^{-6}$ the flow can be considered as dilute (TU *et al.*, 2013).

Figure 2.10 – Flow regimes for dilute and dense flow.



An important concept in the analysis of gas-solid flows is the particle-fluid interactions, which is called the phase coupling. The well-known regions proposed by ELGHOBASHI (1994) of the phase-coupling mechanism are illustrated in Figure 2.11. Each of these regions as well as the characteristic times will be explained next.

Figure 2.11 – Phase-coupling mechanism.

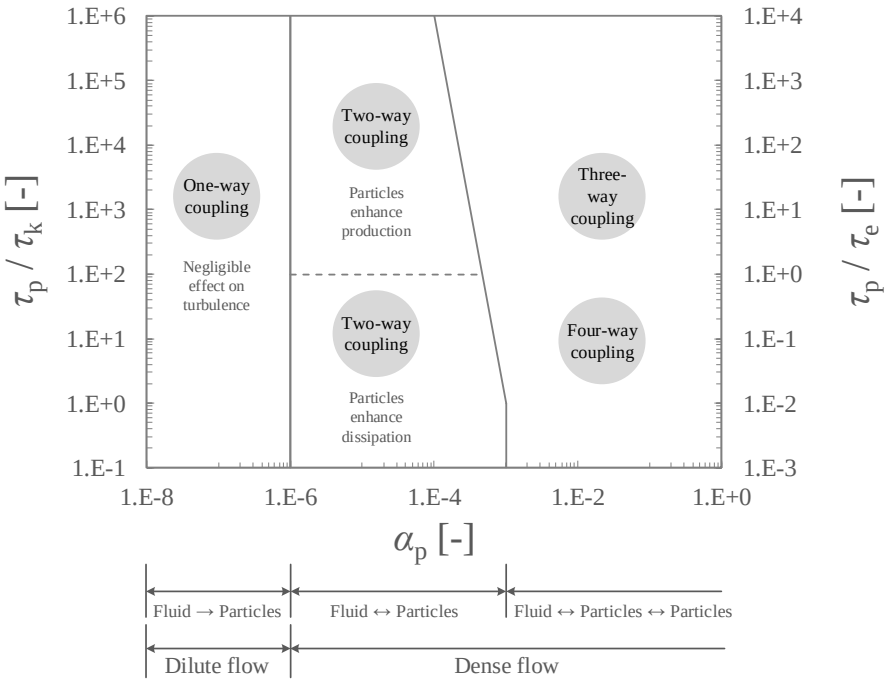


Figure 2.11 presents a map of the phase coupling, according to the particle volume fraction α_p, and the characteristic time of the dispersed and carrier phases. The time scale of the large eddies, τ_e, is given by:

$$\tau_e = \frac{l}{u}, \tag{2.19}$$

where l is the eddy size and u is the characteristic velocity of these eddies (the RMS fluid velocity). The time scale of the smallest eddies, τ_k (the Kolmogorov time scale (POPE, 2000)), can be obtained with a dimensional analysis:

$$\tau_k = \sqrt{\frac{\nu}{\varepsilon}}, \tag{2.20}$$

where ε is the dissipation rate of the turbulent kinetic energy.

The particle relaxation time, τ_p, is a measure of the time the particles need to adapt to a change of the fluid velocity (ARCEN *et al.*, 2006). For the Stokes regime (low Reynolds number flow), the particle relaxation time can be estimated as:

$$\tau_p = \frac{C_c \rho_p {d_p}^2}{18\mu}, \tag{2.21}$$

where d_p is the particle diameter, μ is the dynamic viscosity of the fluid and C_C is the Cunningham slip correction factor given by[7]:

$$C_c = 1 + Kn\left[1.257 + 0.4e^{(-1.1/Kn)}\right], \tag{2.22}$$

where Kn is the Knudsen number:

$$Kn = \frac{\lambda}{d_p/2}, \tag{2.23}$$

where λ is the mean free path of air molecule[8].

A more general relationship to express the particle relaxation time is:

$$\tau_p = \frac{4}{3}\frac{\rho_p}{\rho}\frac{d_p}{C_d\|\vec{u}_{p,r}\|}, \tag{2.24}$$

[7] When the particle size is comparable with the air mean free path ($d_p < 1\mu m$) slips occurs between the air flow and the particle, and the Stokes drag force is modified with the Cunningham correction factor ($C_c > 1$). On the other hand, for higher particles ($d_p \geq 1\mu m$) no correction is needed ($C_c \cong 1$).

[8] The mean free path of air molecules at 25°C and 1 atm is (JENNINGS, 1988): $\lambda = 68.03\ nm$.

where ρ_p is the particle density, ρ is the fluid density, C_d is the drag coefficient and $\vec{u}_{p,r}$ is the particle relative velocity given by:

$$\vec{u}_{p,r} = \vec{U} - \vec{u}_p \,, \tag{2.25}$$

where $\vec{U}$ is the fluid velocity and $\vec{u}_p$ is the particle velocity.

The particle relative velocity can be used to calculate the particle Reynolds number, Re_p, as:

$$Re_p = \frac{\|\vec{u}_{p,r}\| d_p}{\nu} \,. \tag{2.26}$$

The ratio between the particle relaxation time and the eddy lifetime is known as the dimensionless particle relaxation time, ${\tau_p}^+$:

$${\tau_p}^+ = \frac{\tau_p}{\tau_e} \,. \tag{2.27}$$

For one-way coupling, only the fluid affects the particles while the particles do not affect the fluid flow. This occurs when the dispersed phase is dilute and a volume fraction of $\alpha_p \leq 10^{-6}$, usually, this condition is valid particulate laden flows encountered in ventilation ducts.

The two-way coupling phase occurs when the fluid flow influences the particle motion and, in return, the particle motion influences the fluid flow. This approach requires that the particle force source terms (e.g., the drag and lift forces) be included in the momentum equation of the fluid phase. The particle originated source terms are generated for each particle as it is tracked through the flow and these sources are applied in the control volume that the particle is travelling in during a time step.

For very high-volume fraction three-way or four-way coupling can occur. That is, in addition to the two-way coupling between the continuous and discrete phases, the particle-particle interactions become important and need to be accounted for.

Besides the particle dispersion in the flow, the particle can be deposited onto the duct walls, known as particle-wall interactions that will be better explained in section 4.2.3. One of the dimensionless properties of the particle deposition curve (Figure 3.1) is the dimensionless particle deposition velocity that characterizes the flux of particles towards the wall and is defined as (CROWE *et al.*, 2012):

$$V_d{}^+ = \frac{V_d}{u_\tau}, \tag{2.28}$$

where u_τ is the wall friction velocity (Equation (2.12)) and V_d is the deposition velocity that incorporates all complexities of the particle deposition process with the following equation[9]:

$$V_d = \frac{J}{C_\infty}, \tag{2.29}$$

where C_∞ is the free-stream airborne concentration (evaluated away from the surface) and J is the net aerosol flux given by:

$$J = -(D_B + D_t)\frac{dC}{dy}, \tag{2.30}$$

where C is the average concentration of particles in the flow, y is the normal distance from the wall, D_B is the Brownian diffusivity[10] and D_t is the particle turbulent diffusivity.

The Brownian diffusivity is defined as:

$$D_B = \frac{\tau_p k_B T}{m_p}, \tag{2.31}$$

where T is the temperature, m_p is the particle mass and k_B is the Boltzmann constant ($k_B = 1.38\times10^{-23}$ J/K). For particles with $\rho_p = 10^3$ kg/m^3 and $C_c = 1$ in the air at 25°C and 1 atm, the order of magnitude for a range of particle diameter from 0.1 to 100 μm is given in Table 2.4.

Table 2.4 – Order of magnitude of particle relaxation time and Brownian diffusivity

d_p [μm]	τ_p [s]	D_B [m^2/s]
0.1	3×10^{-8}	2×10^{-10}
1	3×10^{-6}	2×10^{-11}
10	3×10^{-4}	2×10^{-12}
100	3×10^{-2}	2×10^{-13}

There are some correlations in the literature to determine the particle turbulent diffusivity (LAI and NAZAROFF, 2000), among them the correlation of LIN *et al.* (1953):

[9] In terms of *M* (mass), *L* (length) and *t* (time) the deposition velocity is (LAI and NAZAROFF, 2005): $V_d = [(M\times L^{-2})\times t^{-1}]/[M\times L^{-3}]$.

[10] Also called the particle mass diffusivity or the particle molecular diffusivity.

$$\frac{D_t}{\nu} = \begin{cases} (y^+/14.5)^3 & , \quad y^+ \leq 5 \\ y^+/5 - 0.959, & y^+ > 5 \end{cases}, \tag{2.32}$$

where y^+ is the non-dimensional distance from the wall, see Eq. (2.15).

We found in the literature some correlations to calculate the deposition velocity for the straight ducts (LIU and AGARWAL,1974; MATIDA *et al.*, 2000) as well as for the curved ducts such as GAO and LI (2012) correlation given by:

$$V_d = -\frac{U_{ave}}{3\pi}\ln(P)\ . \tag{2.33}$$

Another important dimensionless parameter in fluid-particle flow is the Stokes number. The Stokes number, St, is the ratio between a characteristic particle time scale (the particle relaxation time), and a characteristic time scale of the flow defined as (CROWE *et al.*, 2012):

$$St = \frac{\tau_p}{\tau_F}, \tag{2.34}$$

where τ_F is a characteristic time scale of the flow. For a curved duct, the time scale of the flow can be expressed as (SUN *et al.*, 2011):

$$\tau_F = \frac{D_\mathrm{h}/2}{U_{ave}}. \tag{2.35}$$

where U_{ave} is the average velocity of the flow and D_h is the hydraulic diameter of the duct.

If $St \ll 1$ there is an equilibrium condition between the particle and gas, i.e., the particles will follow the flow streamlines. On the other hand, if $St \gg 1$, the particles will follow their own trajectories and there will be a slip between the phases.

Finally, when the number of particles deposited on the duct walls is relevant (typically the case of liquid particles that stick the walls), we can express the result in terms of the collection efficiency[11], η, defined as:

$$\eta = 1 - P\ . \tag{2.36}$$

where P is the penetration:

[11] Also called the deposition efficiency.

$$P = \frac{C_o}{C_i}, \tag{2.37}$$

where C_o is the particle concentration at the bend outlet and C_i is the particle concentration at the bend inlet. There are many correlations in the literature to estimate the penetration in curved ducts as shown in section 3.2.1 (CHENG and WANG, 1981; PUI *et al.*, 1987; MCFARLAND *et al.*, 1997; ZHANG *et al.*, 2012).

2.3 FINAL REMARKS OF FUNDAMENTALS AND DEFINITIONS

This study seeks first the modelling of turbulent airflows through HVAC elbows. The correct physical representation may provide the principal characteristics presented in this chapter such as the Dean vortices along the curve, the separation region along the bend inner wall near the bend outlet and the Görtler vortices along the bend outer wall. Besides, the adverse pressure gradient in the duct bend that increases the pressure drop along the duct system and accurate modelling must be considered.

We can find in the literature two important experimental contribution regarding the 90° duct bend turbulent airflow similar to the operating conditions of ventilation ducts: the study of SUDO *et al.* (1998) for a circular duct with Reynolds number of 6×10^4 and the study of SUDO *et al.* (2001) for a square duct with Reynolds number of 4×10^4. Both studies presented a reliable methodology of the experimental procedure and, consequently, we may trust these results. In addition, their results have good quality and, therefore, they can be used as a validation of our numerical models for the pressure drop, the mean velocity profiles and turbulent parameters.

With an accurate modelling of the turbulent flow, it is then possible to inject a cloud of particles into the flow. The major difficulty is the modelling and quantification of turbulence and its influence on mass, momentum and energy transfer (CROWE *et al.*, 2012).

The carrier phase in this study is described by the fluid density, the fluid pressure and the fluid velocity fields and the dilute phase is described by the particle size, the particle mass loading and the particle velocity fields. The coupling will take place through momentum energy between the phases in which the momentum coupling is the result of hydrodynamic forces on the dispersed phase. The volume fraction of particles in an HVAC is very low (less than 10^{-6}), consequently, a dilute flow is expected and the coupling between the phases can be considered as the one-way coupling as shown by ELGHOBASHI (1994).

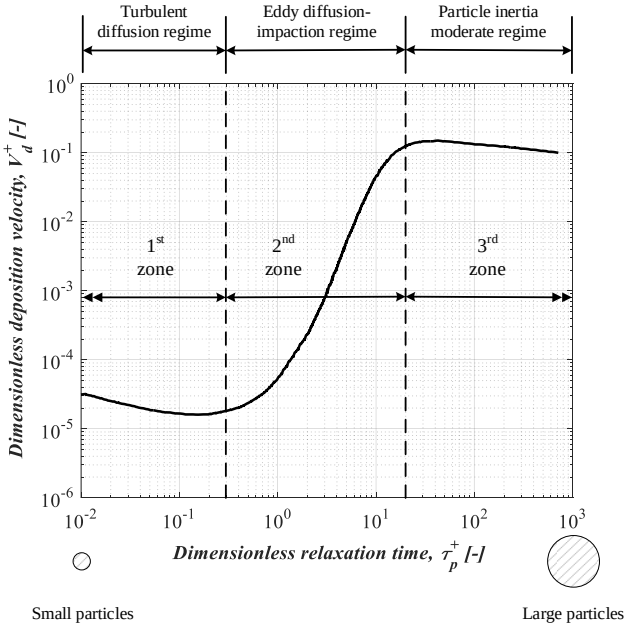
3 LITERATURE REVIEW

The state of the art in this chapter is divided into two parts. First, some relevant numerical and/or experimental works about particle deposition in wall-bounded flows are presented to understand the usual methods applied in this field and the importance of some parameters related to particle dispersion in the flows, such as the Reynolds number, the Stokes number and the particle relaxation time. Second, the works concerning the main subject of this study are discussed, i.e., those related to the numerical study of particle dispersion and deposition on the airflow inside duct bends.

3.1 PARTICLE DEPOSITION IN INTERNAL TURBULENT FLOW

The variation of the deposition velocity with the particle size in a fully developed pipe flow is illustrated in Figure 3.1 through the parameters, dimensionless deposition velocity ($V_d{}^+ = V_d/u_\tau$) and dimensionless relaxation time ($\tau_p{}^+ = \tau_p/\tau_e$). This curve is important because most of the literature presents deposition results in terms of these two dimensionless parameters. Moreover, the shape of the curve changes with the configuration, such as the inclusion of the pipe wall roughness on the deposition velocity as showed in HUSSEIN *et al.* (2012).

Figure 3.1 – A typical variation in a measured deposition rate with particle relaxation time.



Source: Adapted from GUHA (2008).

In the first zone of Figure 3.1, $\tau_p{}^+ < 1$, the deposition is affected by a combination of Brownian and eddy diffusion and depends on a few numbers of properties such as the mean fluid velocity, the duct diameter, and the duct length. The literature presents a lot of straightforward correlations to determine the deposition velocity in this region (LIU and AGARWAL, 1974; LAI and NAZAROFF, 2005; GAO and LI, 2012). For example, Wood's correlation (WOOD, 1981), valid for smooth walls, determines the particle deposition velocity as follow:

$$V_d{}^+ = \frac{3\sqrt{3}}{29\pi} Sc^{-2/3} + K\left(\tau_p{}^+\right)^2 , \tag{3.1}$$

where K is a dimensionless constant ($K = 4.5 \times 10^{-4}$) and Sc is the Schmidt number given by:

$$Sc = \frac{\nu}{D_B + D_t} , \tag{3.2}$$

with ν being the kinematic viscosity of the fluid, D_B the Brownian diffusivity and D_t the turbulent diffusivity.

In the second zone of Figure 3.1, $1 < \tau_p{}^+ < 10$, the velocities acquired by the particles toward the wall is induced by turbulent eddies in the turbulent core and buffer layer as a result of their inertia with a sharp increase in the deposition velocity. Finally, in the third zone, $\tau_p{}^+ > 10$, the particle deposition decreases slightly since large particles are subject to a reduced rate of transport across the inner layer.

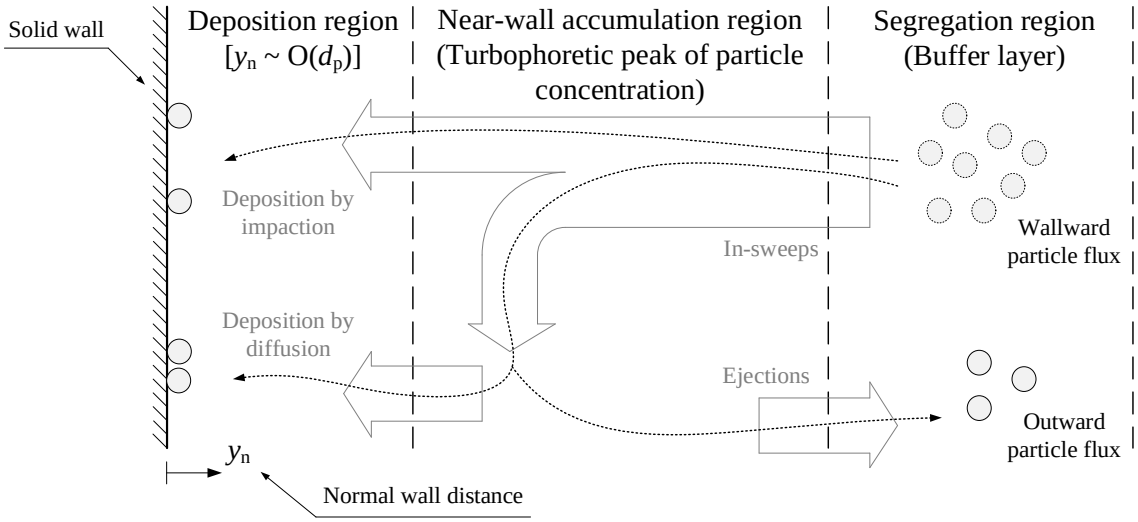
The first step of the particle deposition mechanism is the agglomeration of particles in the region of the buffer layer as shown in Figure 3.2. The particle concentration reaches its maximum in the viscous sublayer and the particles may deposit at the wall or be re-entrained to the outer region by ejection. Finally, the particles reach the wall (the deposition region) for two ways: by impaction where the particles accumulate enough momentum to deposit directly at the wall, and by diffusion where the particles are driven to the wall due to the non-homogeneity of the turbulence and/or electrical charges.

Several works (numerical and/or experimental) related to the study of particle deposition onto surfaces in turbulent diluted internal flows have been developed in the last few decades. They can be classified according to the following categories:

- channel flows: 2D cases of the pipe flows and other 2D channels;

- indoor flows: 3D wall-bounded flows, except pipe flows;
- circular pipe flows: 3D pipe flows within circular pipes;
- rectangular pipe flows: 3D pipe flows within rectangular pipes.

Figure 3.2 – Illustration of the particle near the wall: the deposition mechanisms.



Source: Adapted from SOLDATI and MARCHIOLI (2009).

3.1.1 Channel flows

LI and AHMADI (1992) developed a procedure for simulating the deposition of aerosol particles on a rough surface for a vertical and a horizontal turbulent channel flow with a Reynolds number of 6,700. The turbulence was simulated as a continuous Gaussian random field and the particle trajectories were evaluated with the Finite Difference Method (FDM). The authors considered the particle-wall interactions using the following correlation (FRIEDLANDER, 2000):

$$\frac{u_n}{{u_n}^0} = \sqrt{e^2 - \frac{E_2 - e^2 E_1}{m_p ({u_n}^0)^2/2}}, \tag{3.3}$$

where ${u_n}^0$ is the normal particle approach velocity, u_n is the particle normal velocity after the collision, e is the coefficient of restitution, m_p is the particle mass, E_1 and E_2 are, respectively, the surface potential energy before and after the collision given by (DAHNEKE, 1971, 1972):

$$E = \frac{H d_p}{12 y_0}, \tag{3.4}$$

where H is the Hamaker constant which depends on surface material (ISRAELACHVILI, 2011), d_p is the particle diameter, and y_0 is the equilibrium separation distance between the particle and the surface.

When $u_n = 0$, the particle cannot escape the adhesion forces because all the rebound energy is required to lift it out of the attractive field of the surface. Therefore, the critical approach velocity (u_c), i.e., the velocity below which deposition corresponds to:

$$u_c = \sqrt{\frac{2}{m_p e^2}(E_2 - e^2 E_1)}\,. \tag{3.5}$$

The authors LI and AHMADI (1992) considered particles submitted to Brownian diffusion, shear-lift, and gravity forces, observing that the Brownian forces significantly affect the dispersion of particles with the diameter lower than 0.05 μm and the gravity force is the dominating mechanism for deposition of particles larger than 2 μm. They also studied the influence of the coefficient of restitution for a particle diameter of 30 μm and observed the increase in the coefficient of restitution significantly reduced the particle deposition.

TIAN and AHMADI (2007) conducted numerical simulations to study the transport and deposition of nanoparticles and micro-particles in a turbulent duct flow by testing the accuracy of different turbulence models. The authors modeled a 2D channel airflow with 0.02 m of height and 0.4 m of length, which was discretized with 44,000 quadrilateral elements with the nearest wall node 0.05 mm from the wall after a grid independency study. They concluded that the Reynolds Stress Model (RSM) provides a more accurate description of the flow field near the wall, while the *k-ε* turbulence model leads to a high level of fluctuation perpendicular to the wall and to an over prediction of the particle deposition rates. In addition, they demonstrated that the Continuous Filtered White Noise (CFWN) model and the Discrete Random Walk (DRW) model are both effective stochastic methods for simulating the particle deposition in turbulent flows. They also showed that the use of the standard wall function near the wall region leads to inaccurate deposition velocities.

LU and LU (2015) numerically studied particle deposition in the same channel as previous authors, comparing the particle's behaviors on smooth and ribbed walls. They considered a turbulent flow, and performed 2D simulations, adopting the RSM turbulence and Lagrangian models to respectively model the fluid phase and the particulate phase. They showed that particle deposition is 4,000 times higher onto ribbed walls than onto smooth walls.

This occurs more particularly in the turbulent and eddy diffusion impaction regimes. The authors suggested further investigation dealing with the effects of the ribbed height and spacing.

ALMOHAMMED and BREUER (2016) developed a realistic particle-wall adhesion model validated with a particle-laden turbulent channel flow and compared with other models (WOOD, 1981; PAPAVERGOS and HEDLEY, 1984; KVASNAK *et al.*, 1993; FAN and AHMADI, 1993). They included particle rebound in their model with the following characteristics: the interval of compression and restitution phase depends on the normal restitution coefficient, the effect of the adhesive force (the van der Waals force) is distinguished in the normal and tangential directions, and the treatment of particle motion after impact depends on particle-wall collision type (sticking or sliding). The particle-fluid interactions considered were the drag, lift (due to velocity shear and rotation), gravity and buoyancy forces. The turbulent flow was predicted with the Large Eddy Simulation (LES) modelling. The model predicted well the particle deposition in the channel flow especially for $\tau_p{}^+ > 10$, in which the other models failed because they ignore the rebound effect.

LU *et al.* (2017) investigated particle deposition in ducts typical of building ventilation systems with variable cross-sections using a Eulerian approach based on an RSM turbulence model for describing the flow, and a Lagrangian approach for modelling the particle tracking. They compared particle deposition in expanding and contracting ducts with uniform ducts in 2D configurations. They observed that large particles cannot reach some near-wall regions in expanding ducts while this effect does not occur in contracting ducts.

HONG *et al.* (2018) carried out numerical investigations of particle deposition in a rib-roughed channel using the RSM turbulence model for the carrier phase and the Lagrangian tracking for the dispersed phase. They performed 2D simulation and analyzed the geometry effects of the rib on fluid flow and particle deposition. Their results showed that increasing the roughness height enhanced the entrainment of particles because the particle is entrained by turbulent eddies between the ribs and the particles are deposited mostly in the windward surface of the rough element.

3.1.2 Indoor flow

LAI and NAZAROFF (2000) developed a model based on the modified Fick's law for determining the particle deposition in turbulent flow onto smooth horizontal and vertical indoor surfaces. If the model is relatively limited because of the assumption of a constant flow field,

the particle deposition velocity can be determined almost straightforward from particle size and friction velocity.

LAI and NAZAROFF (2005) conducted an experimental study about particle deposition in turbulent flow onto smooth and rough vertical chamber surfaces for a particle diameter range from 1 to 10 μm, the super-micron particles. The authors showed that the principal factors acting on the particle deposition were the particle size, the airflow condition (e.g., the degree of turbulence) and the surface characteristics (surface roughness).

ZHANG and CHEN (2009) proposed a $\overline{v'^2} - f$ turbulence model with a modified Lagrangian method to predict the particle deposition onto indoor surfaces. An anisotropic particle-eddy interaction model was developed to analyze the particle deposition mechanisms. They validated this model for the case of particle dispersion in an isothermal turbulent channel flow and then focused on the particle dispersion in a cubic cavity where natural convection occurred. The authors founded that the surface orientation (horizontal and vertical) determines the impact of gravitational force on the particle deposition velocity for coarse particles, while the Brownian effect is dominant for particle diameters less than 0.5 μm.

HUSSEIN *et al.* (2009) experimentally investigated aerosol deposition on rough surfaces of different materials (wallpapers, wall plasters, and carpets) in a cubic sealed chamber, observing that the particle deposition on rough surfaces is enhanced by an increase of the roughness surface height.

PISKUNOV (2009) analyzed some classical, semi-empirical and theoretical models of deposition velocity to obtain a model for smooth and rough wall surfaces with vertical and horizontal orientations. The author variated the friction velocity and roughness height and presented the deposition velocity for particle diameter ranging from 0.01 to 100 μm. The results for the smooth surfaces were very accurate, however, there were some differences for the rough surface results regarding particle diameters lower than 1 μm.

3.1.3 Circular pipe flow

LIU and AGARWAL (1974) measured the deposition rate of aerosol in vertical pipes for a range of Reynolds number from 10,000 to 50,000, a range of particle diameters from 1.4 to 21 μm and a range of dimensionless relaxation times from 0.21 to 774. The authors produced reliable results (also reported in table format) that can be used for validation of CFD codes. They concluded that the available theory was not able to explain the results related to a dimensionless relaxation time greater than 30, i.e., in the particle inertia moderate regime.

EL-SHOBOKSHY (1983) determined experimentally the deposition velocity of aerosol in a smooth (glass) and rough (brass) pipe turbulent flow for Reynolds number of 10,000, particle size range from 1 to 6.2 μm and dimensionless relaxation time range from 0.37 to 14.7. The author found that the theory developed by EL-SHOBOKSHY and ISMAIL (1980) predicts reasonably well the deposition velocity in rough pipes. This theory assumes that the average concentration in the bulk of the flow is maintained at the outer part of the boundary layer ($y^+ = 30$), and that the dimensionless deposition velocity of particles on a rough wall is calculated as follows:

$$\frac{1}{{V_d}^+}\int_{{C_S}^+}^{1} dC^+ = \int_{S^+}^{5} \frac{dy^+}{\frac{D_B}{\nu}+\frac{D_t}{\nu}+D_0} + \int_{5}^{30} \frac{dy^+}{\frac{D_t}{\nu}+D_0}\,, \tag{3.6}$$

where ν is the kinematic viscosity, D_B is the Brownian diffusivity, D_t is the turbulent diffusivity and D_0 is given by:

$$D_0 = {\tau_p}^+ \left(\frac{y^+}{y^+ + 10}\right)^2\,, \tag{3.7}$$

where y^+ is the non-dimensional wall distance. The dimensionless particle concentration (C^+) and the dimensionless particle stop distance (S^+) are, respectively, defined as:

$$C^+ = \frac{C}{C_\infty}\,, \tag{3.8}$$

$$S^+ = \frac{S u_\tau}{\nu}\,, \tag{3.9}$$

where C is the particle concentration and S is the particle stop distance. This distance is estimated by $S = v_n \tau_p$, where v_n is the turbulent velocity component normal to the wall and τ_p is the particle relaxation time. The particle stop distance represents the distance that the particle travels towards the wall, while the accompanying eddy is dissipated away.

ZHAO and WU (2006) proposed a Eulerian model based on the LAI and NAZAROFF (2000) model to predict the particle deposition velocity in a turbulent ventilation duct flow. The proposed model includes the turbophoresis phenomenon and needs friction velocity as an input parameter. The authors noted that the turbophoresis where particles move towards the walls by the effect of the turbulence is only significant for particles greater than 1 μm.

HUSSEIN *et al.* (2012) developed a method to describe the lower limit of the boundary layer for calculating the deposition velocity, considering the surface roughness height and the peak-to-peak distance. They validated their results with experimental data from vertical (LIU and AGARWAL, 1974; and EL-SHOBOKSHY, 1983) and horizontal tubes (SEHMEL, 1973). With this model, the authors predicted the deposition velocity with better accuracy than with other models found in the literature (PISKUNOV, 2009).

LOYSEAU and VERDIN (2016) carried out computations to test a Lagrangian model for dispersion and deposition implemented into the OpenFOAM® tool with validation against the results of LIU and AGARWAL (1974). They adopted the one-way coupling Eulerian-Lagrangian approach with the RANS SST k-ω turbulence model to predict particulate transport in a straight pipe turbulent flow. They suggested a new statistical approach for dispersion and deposition analysis.

3.1.4 Rectangular pipe flow

ZHAO and CHEN (2006) numerically analyzed the deposition of particles with size varying from 0.01 to 100 μm in a 3D square straight duct, whose side and length were respectively equal to 0.4 m and 8 m. Two Reynolds numbers were considered: 140,000 and 230,000. The airflow was modeled with the k-ε turbulence model and a drift flux model was used to calculate particle dispersion. The authors found that the particle deposition onto the duct floor was about two times larger than onto other walls. In particles with size up to 0.1 μm, the deposition velocity was roughly the same for the floor and vertical walls.

ZHANG *et al.* (2008) carried out simulations with the Lagrangian tracking model coupled to an RSM turbulence model to compute particle dispersion inside a smooth horizontal square ventilation duct. This study was applied to particle diameters between 10 and 200 μm and Reynolds numbers between 62,500 and 187,500. The authors demonstrated that the dimensionless deposition velocity depends on the friction velocity for the same dimensionless relaxation time. They also demonstrated that the deposition rate on the floor surface decreases when the friction velocity increases, while for the other surfaces the deposition rate increases with the raising of friction velocity.

GAO and LI (2011) predicted the deposition rates for particle diameters between 10 to 200 μm in a 3D vertical smooth ventilation duct with a square cross-section, whose hydraulic diameter varied between 0.1 and 0.3 m. The turbulent flow was predicted with an RSM and particles were tracked with a Lagrangian method, considering inlet velocities between 3 and

7 m/s and Reynolds numbers between 20,000 and 145,000. The results suggested that the particle deposition velocity in vertical ducts was mostly influenced by the lift force and the gravity force.

PHUONG and ITO (2013) conducted both experimental and numerical investigations of the behavior of airflow and particle dispersion in a ventilation duct with a square cross-section and a hydraulic diameter of 0.1 m. The authors considered an isothermal turbulent airflow and the following RANS turbulence models: a low Reynolds number (LRN) k-ε model (ABE *et al.*, 1994), the renormalization group (RNG) k-ε model (YAKHOT *et al.*, 1992) and the Shear Stress Transport (SST) k-ω model (MENTER ,1992). They selected the RNG k-ε model because it yielded a better prediction of the flow patterns. The particle dispersion was computed using a Lagrangian approach. Concerning the study of the influence of baffles on the transport of particles in the duct, it was found that the baffles increased the particle deposition and decreased the particle escape.

3.2 PARTICLE DEPOSITION IN DUCT BENDS

Over the past few decades, several numerical and/or experimental studies investigated particle dispersion and deposition in pipe bends. These studies are presented in this section according to the approach adopted, experimental or numerical, and the shape of the cross-section, round or rectangular. This last subdivision is important because the geometry of the duct influences the flow patterns such as the secondary flow that is different for circular and rectangular ducts, which can, in turn, affect the particle deposition.

3.2.1 Experimental studies

The experimental studies presented here are summarized in Table 3.1. These studies are fundamental cases of particle deposition in duct bends that can be used to validate CFD codes. The details of the experimental apparatus and methodology are not mentioned, the focus is on the flow conditions, particle properties, and principal results. The studies were divided according to the shape of the duct bend cross-section treated: round or rectangular.

Table 3.1 - Summary of experimental studies of particle deposition on duct bends.

Reference	Duct						Particle			
	Cross section	D_h [b] [mm]	δ [c] [-]	Orientation	Re [d] ($\times 10^3$) [-]	De [e] ($\times 10^3$) [-]	ρ_p [f] [gcm^{-3}]	d_p [g] [μm]	τ_p^+ [h] [-]	St [i] [-]
KLIAFAS and HOLT (1987) [a]	Rectangular	100	3.5	Vertical to Horizontal	220-347	117-185	2.99	50-100	-	-
SIPPOLA and NAZAROFF (2005)		152	1.5		21.7-88.8	17.7-72.5	0.89	1-16	0.01-10	0.0001-0.08
YANG and KUAN (2006)		150			102	83.3	2.50	77	-	0.003
SUN and LU (2013)		100	3.4		17.9-35.6	9.7-19.3	2.65	0.7-100	0.337-8	0.01-0.29
SUN *et al.* (2013)										
PUI *et al.* (1987) [a]	Round	1-9	5.6-7	Horizontal to Horizontal	0.1-10	0.03-4	0.89	1-7	-	0.1-1.3
McFARLAND *et al.* (1997) [a]		16	2-10		3-20	1-14		10	0.4-37	0.07-0.7
PETERS and LEITH (2004a)		203	3		361	208.4	2.45	5-150	-	0.025-23
YIN *et al.* (2015) [a]		12	1.4-36		1.2	0.2-1		0.01-0.56	-	-

[a] Not an HVAC application.
[b] Hydraulic diameter.
[c] Curvature ratio.
[d] Reynolds number.
[e] Dean number.
[f] Particle density.
[g] Particle diameter.
[h] Dimensionless particle relaxation time.
[i] Stokes number.

3.2.1.1 Round duct bend

PUI *et al.* (1987) carried out an experimental study about the deposition of droplets on horizontal to horizontal duct bends in laminar (*Re* = 100 and 1,000) and turbulent (*Re* = 6,000 and 10,000) flows. They compared the data of laminar flow with the correlation of CHENG and WANG (1981) in which the penetration, P ($P = C_o/C_i$), is estimated by:

$$P = 1 - \left(\frac{2}{\pi} + \frac{1}{\delta} + \frac{4}{3\pi\delta^2}\right)\alpha_b St \,, \tag{3.10}$$

where δ is the curvature ratio, α_b is the bend angle and St is the Stokes number.

The authors obtained a good agreement for *Re* = 1,000 regarding the deposition efficiency η ($\eta = 1 - P$). However, for *Re* = 100, the experimental data did not fit the model. Therefore, they suggested that the Equation (3.10) should be useful for the following Reynolds number range: $1{,}000 \leq Re \leq 4{,}000$.

PUI *et al.* (1987) also demonstrated that the particle penetration does not depend on the Reynolds number for turbulent flows in pipe bends and proposed the following correlation:

$$P = 10^{-0.963St} \,. \tag{3.11}$$

Furthermore, they showed that approximately 70% of the total internal area of the duct bend is an effective collecting surface because of the secondary motion of the flow, which allows particles to deposit on the outer and inner surfaces of the bends.

A similar study was proposed by MCFARLAND *et al.* (1997) with Reynolds number ranging from 3,200 to 19,900. The authors developed another model to predict particle penetration through the turbulent flow in pipe bends:

$$P = exp\left[\frac{4.61 + C_1\alpha_b St}{1 + C_2\alpha_b St + C_3\alpha_b St^2 + C_4\alpha_b{}^2 St}\right], \tag{3.12}$$

where the coefficients are: $C_1 = -0.9526 - 0.05686\delta$, $C_2 = \frac{-0.297 - 0.0174\delta}{1 - 0.07\delta + 0.0171\delta^2}$, $C_3 = -0.306 + \frac{1.895}{\sqrt{\delta}} - \frac{2}{\delta}$ and $C_4 = \frac{0.131 - 0.0132\delta - 0.000383\delta^2}{1 - 0.129\delta + 0.0136\delta^2}$.

Equation (3.12) is more appropriate than Equation (3.11) since the model adds the influence of the curvature ratio and the bend angle in the particle penetration correlation.

The authors (MCFARLAND *et al.* ,1997) also concluded that the particle deposition does not depend on the Reynolds number for turbulent flows in pipe bends. However, they found that the increase in the curvature ratio improves the aerosol penetration although not significantly for curvature ratio greater than 4[12], and the aerosol penetration decreases with flattening and with the increase in the bend angle.

PETERS and LEITH (2004a) studied the mass distribution of particles collected in a 0.2 m diameter industrial curved duct at Reynolds number of 368,000. The authors noted that the deposition increased from 35% for 15 μm particle size to nearly 100% for 100 μm. In another study (PETERS and LEITH, 2004b), the influence of the bend angle, the bend curvature ratio and the bend orientation on the particle deposition in duct bends was investigated, finding disagreement with the deposition rates measured in small tubes.

Another experimental research was conducted by YIN *et al.* (2015). Although this work does not concern to duct bends encountered in typical HVAC applications, it determined the penetration efficiency as a function of the Dean and Schmidt numbers and the bend angle and showed that the effect of the Dean number on the penetration efficiency depends on the particle size.

3.2.1.2 Rectangular duct bend

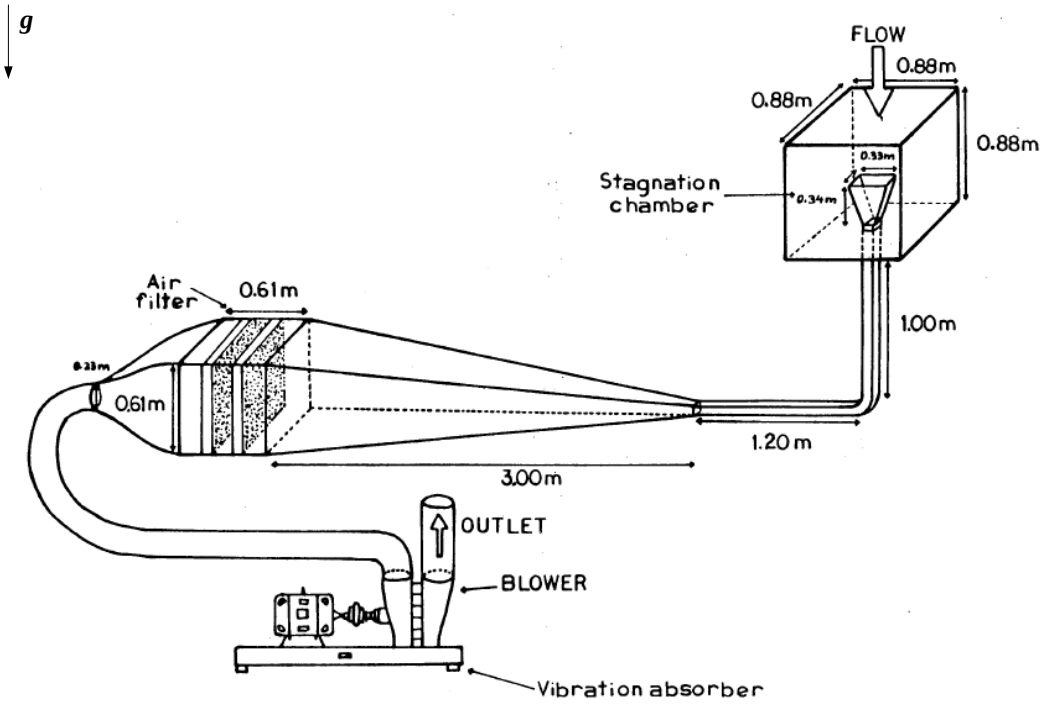
KLIAFAS (1984) measured the stream-wise and radial velocities and associated Reynolds stress components for the gas-solid turbulent flow in a square-sectioned 90° bend with the LDV (Laser Doppler Velocimetry) technique in the experiment presented in Figure 3.3. They considered two Reynolds numbers for the air carrier phase (2.2×10^5 and 3.5×10^5) and two-particle diameters for the glass particle diluted phase (50 and 100 μm). The results were presented in graphical format for many positions inside the bend after statistical analysis, being useful for validation of CFD codes.

KLIAFAS and HOLT (1987) realized experiments in a square duct bend with a hydraulic diameter equals to 0.1 m, measuring the velocities of the solid particles and the fluid at some planes along the elbow and according to two directions. The authors observed that the inner wall of the duct bend was free of erosion while the outer wall was not. Due to the high velocity of the experiment, i.e., an average speed of about 30 and 50 m/s, all walls were free of deposition. They also observed that the level of turbulence of the solid particles was higher than

[12] The curvature ratio of ventilation duct bends is up to 5 according to WU and ZHAO (2007).

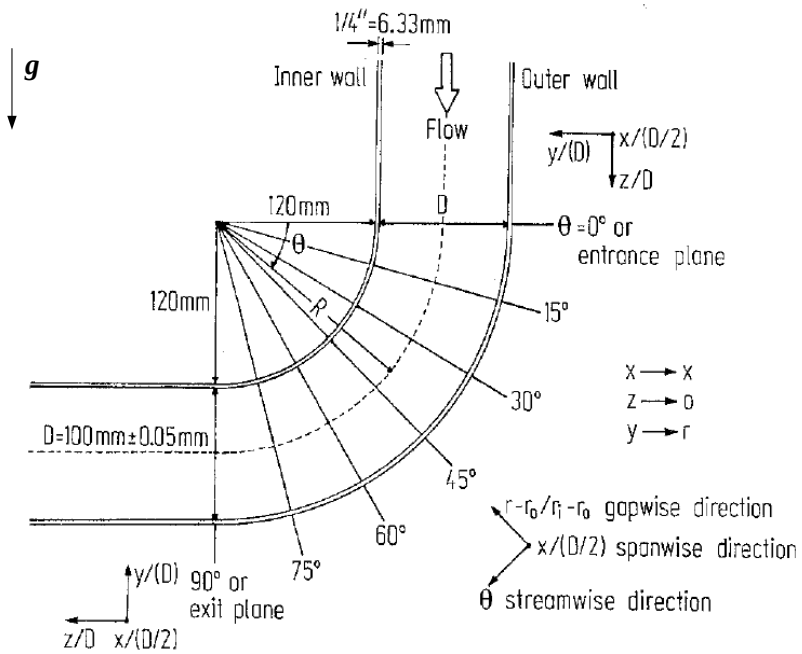
that of the air, except near the outer wall region and among 30° and 60° into the elbow, as indicated in Figure 3.4, where most particle-wall collisions occur.

Figure 3.3 – Representation of the experimental apparatus.



Source: KLIAFAS (1984).

Figure 3.4 – Geometry of the curved duct.



Source: KLIAFAS and HOLT (1987).

SIPPOLA and NAZAROFF (2005) carried out measurements of particle deposition in S-connectors, bends and straight ducts, in a laboratory. They were the first authors to investigate the particle deposition in typical commercial mechanical ventilation systems. Concerning the duct bend, the cross-section was square, the hydraulic diameter was 0.15 m and the flow was turbulent with Reynolds number ranging from 21,700 to 88,800. The authors evaluated the penetration through the bends for different particle diameters, ranging from 1 to 16 μm. They observed that the penetration decreases with Stokes number increasing, i.e., with the particle diameter and/or the air velocity. At Stokes number below 0.01, the penetration through the bends was close to 100%. Also, they found that for turbulent flow the deposition in bends is normally greater than the deposition in straight ducts, for the same particle sizes and airflow velocities.

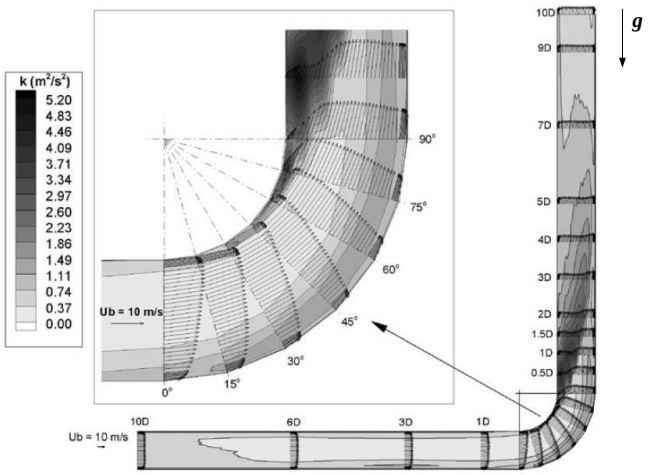
YANG and KUAN (2006) experimentally investigated the behavior of particles inside a curved duct of a pneumatic conveying system with a square cross-section of hydraulic diameter equal to 0.15 m, by measuring the velocities of gas and particle phases. A set of results is shown in Figure 3.5, in terms of gas-phase velocity and turbulent kinetic energy distributions. The authors noted an important level of turbulence near the inner wall after the bend, and that the flow became detached from the inner wall after a position of 45° along the bend. Significant gas-solid separations were detected near the outer wall of the duct while gas and solid velocities were almost the same in the inner wall region.

SUN and LU (2013) used the experimental apparatus shown in Figure 3.6 to study particle deposition in duct bends, by considering three factors: the inlet mass concentration, the Reynolds number, and the duct surface material. Regarding the inlet mass concentration, penetration decreased with the increase in the Stokes number and, for a given particle diameter, the penetration decreased with the increase in the inlet mass concentration. Although Reynolds number does not alter the penetration trend significantly, it affects the particle concentration distribution. For higher Reynolds numbers the stronger turbulent eddies and secondary flows perturbed the particle concentration along the duct radius by decreasing the particle concentration near the bend inner wall and increasing the particle concentration near the bend outer wall. The authors compared two wall materials: acrylic glass and galvanized steel. The penetration in the galvanized steel duct was 5% lower than that of the acrylic glass bend while the materials' mass concentrations along the curve were close to each other.

Similar work was developed by SUN *et al.* (2013). They measured the particle deposition in a duct bend of the rectangular cross-section for two Reynolds numbers (17,900

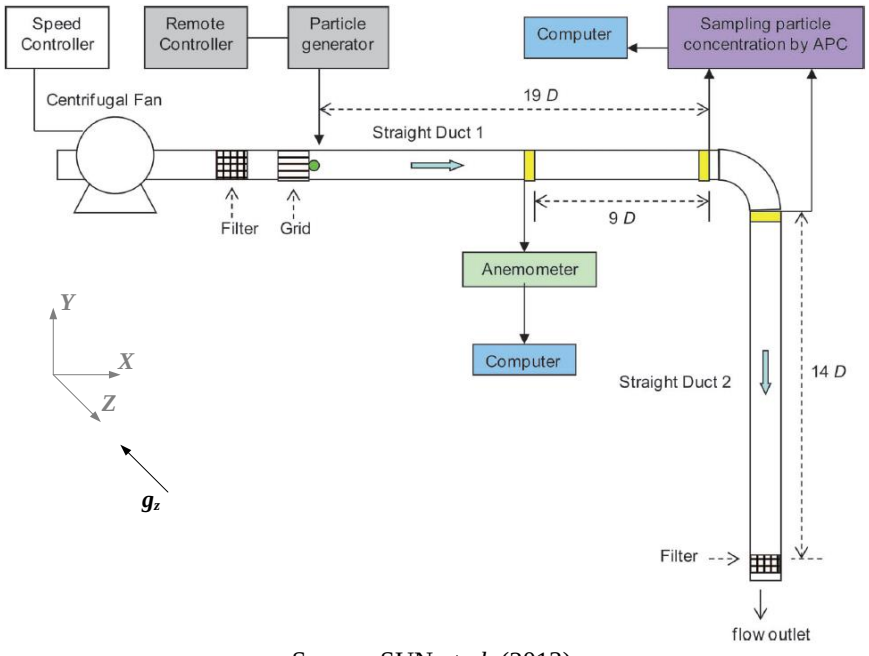
and 35,600) and a range of particle diameters between 0.3 and 25 μm. The authors reported that the particle penetration in the bend reduces from almost 100% for $St = 0.001$ to 64% for $St = 0.55$ while the variation of Reynolds number does not significantly change the deposition velocity and penetration. For $St \geq 0.001$ the particle penetration is higher than that found in previous studies with a circular cross-section (PUI *et al.*, 1987; MCFARLAND *et al.*, 1997; PETERS and LEITH, 2004a) probably caused by the consideration of particle debounce from bend wall in the presence of centrifugal forces.

Figure 3.5 – Mean velocity vectors and turbulent kinetic energy contours for the gas phase at the middle plane.



Source: YANG and KUAN (2006).

Figure 3.6 – Schematic diagram of the experimental apparatus.



Source: SUN *et al.* (2013).

3.2.2 Numerical studies

The principal points observed in the numerical studies discussed in this section are summarized in Table 3.3 and Table 3.4 for bends with round and rectangular cross-section, respectively. Most of these studies selected a one-way coupling Eulerian-Lagrangian approach, with a RANS turbulence model for the continuous phase (air) and Lagrangian particle tracking formulation for the disperse phase (particles) integrating an eddy lifetime model to consider the particle-turbulence interaction. Regarding the particle-fluid interactions, most studies considered the drag and gravity forces, and for some cases, the lift force was also important. The particle-wall interaction depends on the physics of the problem, the solid particles can rebound from the duct wall surface, while liquid particles must stick on the duct wall. The principal advantages and drawbacks of the Eulerian-Lagrangian in comparison to the Eulerian-Eulerian approaches are given in Table 3.2 (CROWE *et al.*, 2012; TU *et al.*, 2013).

Table 3.2 – Particle transport: numerical methods.

Method	Models	Advantages	Drawbacks	Application
Eulerian-Lagrangian	CFD-DEM CFD-DSMC CFPD	ODE (Ordinary Differential Equations). Considers the microscopic transport processes. Detailed information about individual particles. Can handle different particle sizes and characteristics. Non-equilibrium flows. Particle-wall interactions are naturally applied.	Computationally expensive. Low volume fractions. Turbulence dispersion is modeled. The coupling with the fluid is restrictive. The particles are typically points.	Diluted flows. High-inertia and low-inertia particles.
Eulerian-Eulerian	VOF Mixture	Computationally economical. Can handle industrial multiphase applications. Can handle both dilute and dense flow. Equilibrium flows.	PDE (Partial Differential Equations) Requires constitutive modelling of interactions forces. The boundary conditions may be adjusted for different flow regimes. Different particle size and characteristics must be a different phase.	Dense flows. Low-inertia particles.

Table 3.3 - Summary of numerical studies of particle deposition on duct bends with a round cross-section.

Reference	Disperse phase	Continuous phase [b]	Numerical modelling [c]	Stochastic model [d]	Particle-fluid interactions	Particle-wall interactions
LEVY and MASON (1998)	Eulerian	RANS (modified *k*-*ε*)	FVM (3D) -	-	Drag	Rebound
ZAICHIK *et al.* (2010)		RANS (modified *k*-*ε*)	FVM (3D) Two-way coupling	-	Drag	Rebound
TSAI and PUI (1990)	Lagrangian	Laminar	FDM (3D) One-way coupling	-	Drag	Stick
BREUER *et al.* (2006)		LES	FVM (3D) One-way coupling	-	Drag Gravity	Stick
BERROUK and LAURENCE (2008)		LES	FVM (3D) One-way coupling	-	Drag Gravity	Stick
SUN *et al.* (2011) [a]		RANS (RNG *k*-*ε*)	FVM (2D) One-way coupling	DRW	Drag Gravity	Rebound
ZHANG *et al.* (2012)		RANS (RSM)	FVM (3D) One-way coupling	DRW	Drag Gravity	Stick
SUN *et al.* (2012) [a]		RANS (RSM)	FVM (2D) One-way coupling	DRW	Drag Gravity Lift	Rebound

[a] HVAC elbow or similar operational conditions.
[b] Turbulence model of the fluid flow.
[c] FVM (Finite Volume Method) or FDM (Finite Difference Method). The case can be two-dimensional (2D) or three-dimensional (3D). The coupling between the continuous and disperse phases can be: one-way, two-way, three-way or four-way.
[d] DRW (Discrete Random Walk) is also called the eddy lifetime model.

Table 3.4 - Summary of numerical studies of particle deposition on duct bends with a rectangular cross-section.

Reference	Disperse phase	Continuous phase [b]	Numerical modelling [c]	Stochastic model [d]	Particle-fluid interactions	Particle-wall interactions
TU and FLETCHER, (1995)	Eulerian	RANS (RNG *k*-ε)	FVM (3D) One-way coupling	-	Drag Gravity	Rebound
MOHANARANGAM *et al.* (2008)		RANS (RNG *k*-ε)	FVM (3D) Two-way coupling	-	Drag Gravity	Rebound
NAIK and BRYDEN (1999)	Lagrangian	RANS (modified *k*-ε)	FVM (2D) One-way coupling	DRW	Drag Gravity	Rebound
KUAN *et al.* (2007) [a]		RANS (RSM)	FVM (3D) One-way coupling	DRW	Drag Gravity Lift	Rebound
TIAN *et al.* (2008)		RANS (RNG *k*-ε)	FVM (3D) One-way coupling	DRW	Drag Gravity	Rebound
HONGTAO and LI (2011)		RANS (RSM)	FVM (3D) One-way coupling	DRW	Drag Gravity Lift	Stick
GAO and LI (2012) [a]		RANS (RSM)	FVM (3D) One-way coupling	DRW	Drag Gravity Lift	Stick
NJOBUENWU *et al.* (2013) [a]		RANS (RSM)	FVM (3D) One-way coupling	Random Fourier Series	Drag Gravity Lift	Rebound

[a] HVAC elbow or similar operational conditions.
[b] Turbulence model of the fluid flow.
[c] FVM (Finite Volume Method) or FDM (Finite Difference Method). The case can be two-dimensional (2D) or three-dimensional (3D). The coupling between the continuous and disperse phases can be: one-way, two-way, three-way or four-way.
[d] DRW (Discrete Random Walk) model is also called the eddy lifetime model.

3.2.2.1 Round duct bend

TSAI and PUI (1990) studied the particle deposition in a 90° bend with a circular cross-section and laminar flow to propose a correlation for the deposition efficiency[13] as a function of Stokes number. Their correlation agreed with the literature and they found that the bend inlet velocity profile, the curvature ratio and the Dean number have considerable effects on deposition efficiency. The increase in the Dean number and the decrease in the curvature ratio contributed directly to increase the particle deposition efficiency for a given Stokes number. The authors also observed that in the case of a parabolic bend inlet velocity profile, the deposition efficiency is always higher than that of a uniform bend inlet velocity profile.

LEVY and MASON (1998) investigated the particle concentration in a 90° bend with a circular cross-section using a Eulerian-Eulerian model of gas-solid turbulent flow in a pipe system. The authors considered the influence of the pipe diameter, the curvature ratio, the particle size and the gravity direction on the particle concentration in the pipe. They observed that the concentration of particles in the cross-section of a circular pipe was not uniform a few meters after the bend.

BREUER *et al.* (2006) analyzed with an in-house code the turbulent 90° bend laden flow with monodisperse and diluted particles. They employed LES to model the turbulent flow and a Lagrangian approach to predict the motion and deposition of 2.5×10^5 particles. Their geometry model was simplified with one duct diameter length upstream straight pipe and two duct diameters downstream straight pipe, to avoid the inflow and outflow conditions that can disturb the bend flow. The authors showed that the particle deposition in duct bends depends mostly on four dimensionless parameters: Reynolds number of the flow (Re), Reynolds number of the particle (Re_p), curvature ratio (δ) and Stokes number (St).

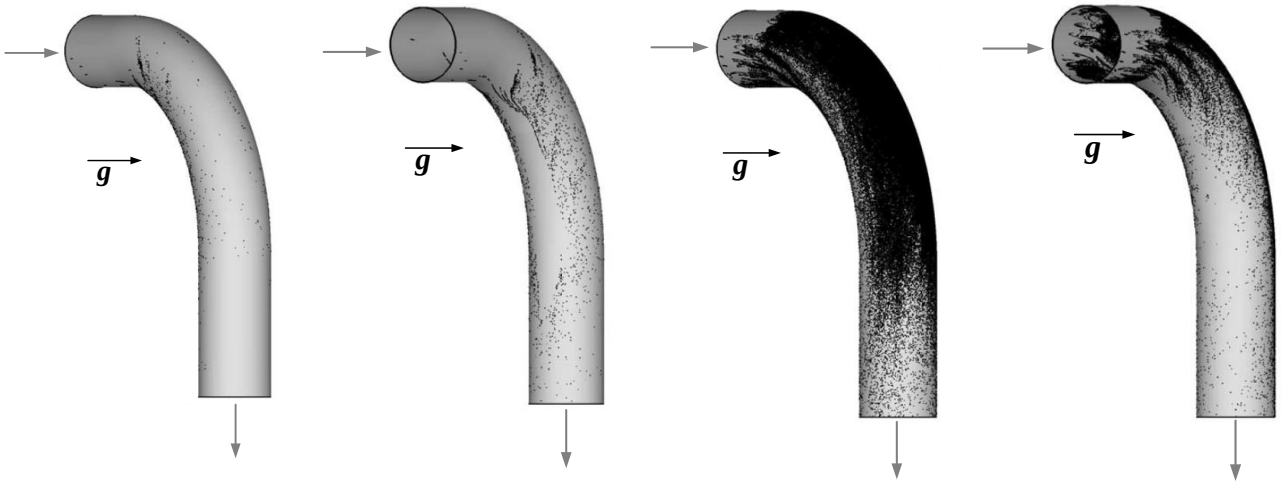
Patterns of particle deposition achieved in the work[14] of BREUER *et al.* (2006) are shown in Figure 3.7a and Figure 3.7b for two Stokes numbers. It can be seen in these figures that the majority of the particles with lower size (lower Stokes number) follow the flow streamlines, while the particles with higher size (higher Stokes number) did not followed the flow streamlines and are deposited in the outer bend wall because of their inertia and the centrifugal forces. Another interesting result of their study is the number of particles deposited

[13] The deposition efficiency is the ratio between the particle mass deposited in the bend and the total particle mass in the domain.
[14] For this case the stick condition is considered. i.e., the particle is deposited when it reaches the wall.

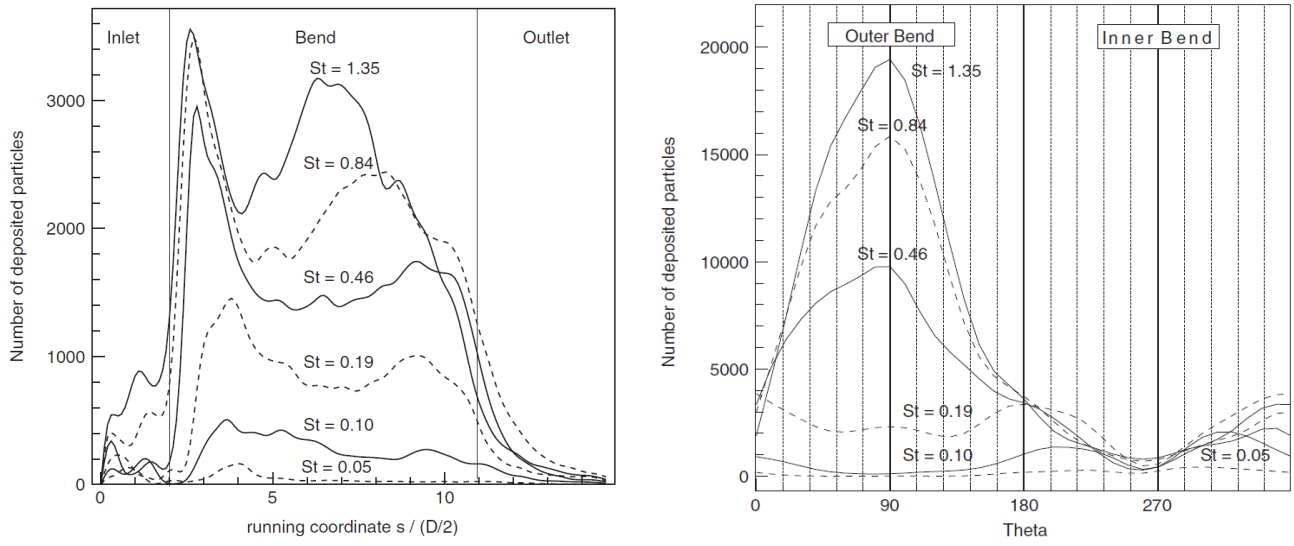
along the middle vertical plane in the outer wall for different Stokes number. As presented in Figure 3.7c, it is possible to note how important is the particle deposition in the bend when we compare it with the deposition in the straight pipes (the inlet and outlet regions), mainly for higher Stokes numbers. Finally, the number of deposited particles at the bend outlet (α_b = 90°) is given in Figure 3.7d, where *Theta* represents the cross-section angle and one can observe that the particle deposition is greater on the outer bend wall than in the inner one.

Figure 3.7 – Particle deposition for *Re* = 10,000: (a) *St* = 0.05, (b) *St* = 1.35, (c) pipe and (d) bend outlet.

(a) $St = 0.05$. (b) $St = 1.35$.



(c) Particle concentration along the pipe. (d) Particle concentration in the bend outlet.



Source: BREUER *et al.* (2006).

BERROUK and LAURENCE (2008) carried out similar computations as BREUER *et al.* (2006) with a shorter computational domain and a coarser grid to save computing time. They attributed some small differences concerning the deposition efficiency to the grid resolution and distribution of cells (especially in the cross-section). They demonstrated the importance of the stochastic model to reproduce the instantaneous particle velocity. They also observed that the injection position of small particles must be close to the wall, otherwise these particles will not be deposited on the outer bend wall.

ZAICHIK *et al.* (2010) extended the Eulerian model DIM (Diffusion-Inertia Model) to include the back-effect on the fluid turbulence of low inertia particles in the frame of two-way coupling. With this model, a concentration equation is used to solve the particulate phase. They evaluated the aerosol deposition in straight pipes and circular bends. The latter case achieved good agreement in terms of deposition efficiency against experimental and numerical (Lagrangian approach coupled with the LES model) studies. Their developed method reduced the computational cost compared to two-fluid Eulerian models, where momentum equations are solved for the particulate phase.

SUN *et al.* (2011) analyzed the particle movement and deposition in a duct bend with a Lagrangian model integrating a particle-wall collision model (BRACH and DUNN, 1998). The airflow was computed with the RNG k-ε turbulent model. They compared the "traditional trap wall model" (where the particles stick when impacting onto a wall) with the particle-wall collision model of BRACH and DUNN (1998) in terms of velocity profiles, distribution number, and deposition velocity. They observed that the 2D simulation underestimates the particle velocity regarding the experimental data of KLIAFAS and HOLT (1987). They also noted that the particle-free zone next to the inner wall of the bend increases for higher particle diameters. The particle wall collision model predicted up to 11% more particle penetration than the traditional trap wall model at the end of the bend.

SUN *et al.* (2012) investigated the same problem reported by SUN *et al.* (2011). They analyzed the particle deposition in and after the bend, with different particle diameters and duct materials (copper, mica, and molybdenum). They also employed an algebraic RSM (WALLIN and JOHANSSON, 2000) rather than the RNG k-ε turbulence model. The deposition particle number increased slowly for the deflection section from 0° to 60° and quickly among 60° and 90°. Particle deposition was higher on the molybdenum surface than on the mica and copper surfaces. The deposition velocities were lower on the copper wall, probably because of the low capture velocity near the surface. The authors remarked that the predicted deposition velocities

were much higher in bends than in straight ducts and that the particle deposition raises after the elbow due to bend-induced turbulence.

ZHANG *et al.* (2012) carried out simulations to evaluate the performance of RANS models with different treatment near the wall to determine pressure drop and deposition efficiency of a suspension flow through curved pipes. They selected RSM to model turbulence due to the presence of strong streamline curvatures. They recommended the Enhanced Wall Treatment (EWT) of WOLFSHTEIN (1969) to model the flow near the wall and proposed the following model of particle deposition efficiency in curved pipes[15]:

$$\eta = 1 - e^{\left[-0.528\alpha_b\left(\delta^{1/2}\right)\left(St^{2^{1/\delta}}\right)\right]}, \tag{3.13}$$

where η is the deposition efficiency, α_b is the bend angle, δ is the curvature ratio and St is the Stokes number.

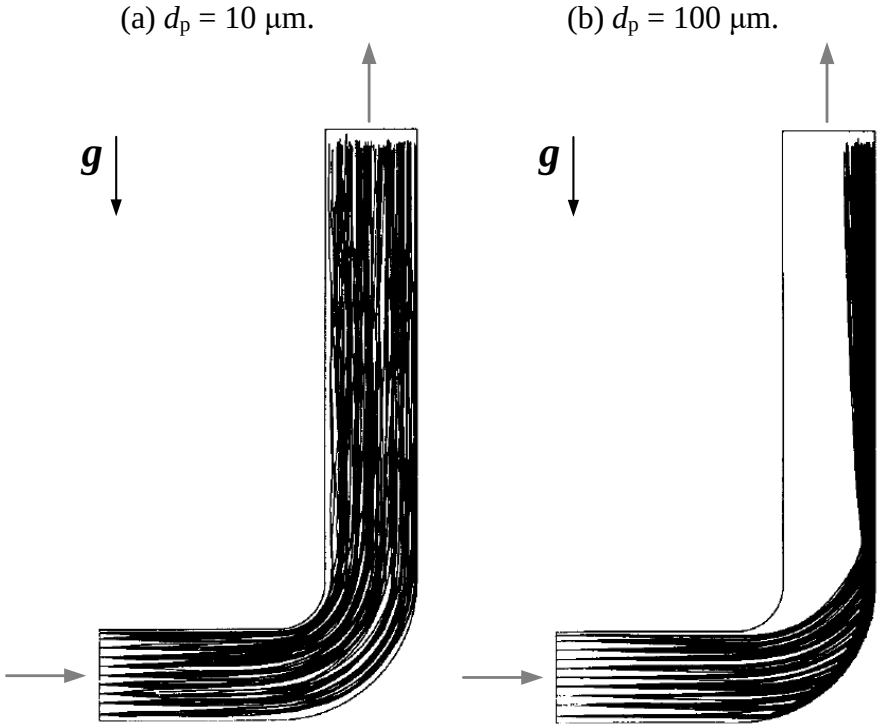
3.2.2.2 Rectangular duct bend

TU and FLETCHER (1995) carried out numerical simulations of a turbulent gas-solid flow in a squared-sectioned 90° bend based on the Eulerian approach for the particles with the RNG *k-ε* turbulence model. They considered the one-way coupling between the particle and gas phase, the gradient hypothesis to model the turbulent effects over the particles and a momentum source term to model the particle-wall interactions. Their main results were presented in terms of particle mean velocity and turbulence intensities. The authors observed that the particle-wall interactions proved to be a controlling factor that most influenced the behavior of solid particles in turbulent two-phase flows.

NAIK and BRYDEN (1999) examined a dilute two-phase flow in a curved pipe with an Eulerian-Lagrangian approach and a low-Reynolds *k-ε* turbulence model. They detailed the numerical and mathematical modelling such as the description of the lose-find Lagrangian particle tracking method. One of their results is illustrated in Figure 3.8 to show that small particles (d_p = 10 μm) followed the flow streamlines while large particles (d_p = 100 μm) do not follow the flow because of their inertia. The authors noted that the particle dispersion due to the gas turbulence affected only the smaller particles, i.e., particles smaller than the smallest turbulent length scales.

[15] Valid for a curvature ratio range from 1.5 to 20. The only model that considers all important parameters of a curved pipe.

Figure 3.8 – Particle trajectories for particle diameter of (a) 10 μm and (b) 100 μm.



Source: NAIK and BRYDEN (1999).

KUAN *et al.* (2007) performed both numerical and experimental study of dilute gas-solid flow through a 90° bend of 0.15 m hydraulic diameter for Reynolds number of 10^5 and particle diameter of 77 μm. Regarding the numerical study, they employed the RSM SSG turbulence model to solve the carrier phase and a Lagrangian method to track almost 10^5 particles of the diluted phase using the one-way coupling and the eddy lifetime model to consider the velocity fluctuations. The authors validated both solid and gas velocity and turbulence intensity profiles against their experimental data. They observed some deficiencies of the RSM SSG to describe the flow near the outer bend wall, which affects the particle tracking since the collision between particles and the outer bend wall occurs more often than the collisions with the inner bend wall.

MOHANARANGAM *et al.* (2008) investigated the gas-particle flow in a square-sectioned duct bend of hydraulic diameter 0.1 m, for a Reynolds number of 347,000. The authors employed a modified Eulerian method to describe the gas-particle flow and modeled the turbulence with the RNG k-ε model. The authors applied the particle-wall collision model of TU and FLETCHER (1995) as a generalized boundary condition for the particle phase, to compensate for the lack of formulation of the Eulerian model to describe the particle-wall

interactions. The results were compared to those obtained experimentally by KLIAFAS and HOLT (1987). Gas and particle mean velocity profiles along some positions along the curved duct showed good agreement between empirical and predicted data. Besides, results of turbulence intensity and particle concentration along with the flow into the elbow were presented for a large range of Stokes numbers, from 0.01 to approximately 13.

A similar numerical study was developed by TIAN *et al.* (2008) who modeled the same problem with a Lagrangian approach. The authors investigated the effects of wall roughness on the particle and flow properties by comparing particle and gas mean velocity profiles with experimental data (relative error of 10%) and analyzing the particle trajectories. They observed that the wall roughness reduces the particle-free zone (i.e., regions in the duct without particle concentration) in the vicinity of the elbow inner wall, decreases the mean particle velocity and increases the particle fluctuation velocity in both stream-wise and span-wise directions.

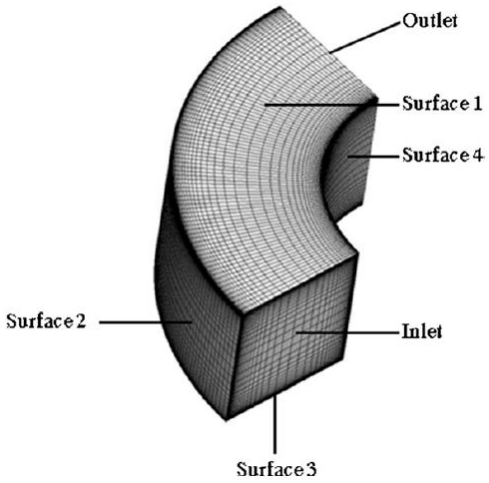
HONGTAO and LI (2011) investigated the particle deposition in a square 90° bend by an Eulerian-Lagrangian approach based on the RSM turbulence model. Their results of collection efficiency were validated against the experimental ones obtained by SIPPOLA (2002). The authors analyzed the influence of the Reynolds number and the curvature ratio regarding the collection efficiency and particle deposition velocity. They showed that the deposition velocity increases with the air mean velocity and the collection efficiency decreases with curvature ratio increasing.

GAO and LI (2012) studied the particle deposition in duct bends with a square cross-section of 0.3 m and horizontal and vertical orientations (Figure 3.9a and Figure 3.9c). They chose an RSM turbulence model and a Lagrangian method to describe the gas and the solid phase, respectively. Regarding the literature of particle deposition on HVAC curved ducts, an enormous range of particle diameters (10 to 200 μm) and Reynolds numbers (34,000 to 365,000) was investigated. The numerical results (the relationship between the dimensionless particle deposition velocity and dimensionless particle relaxation time for a range of 4 to 36) were validated with the empirical results of MCFARLAND *et al.* (1997) and PUI *et al.* (1987), observing that the dimensionless deposition velocity on the outer surface of the bend is always the highest velocity compared to the other surfaces as shown in Figure 3.9b. The interactions between the particles and the eddies near the wall were more important for smaller particles ($\tau_p^+ \leq 100$) than for larger particles ($\tau_p^+ > 100$) which reach the wall through momentum transmitted by large eddies in the turbulent flow. The authors noted that the dimensionless

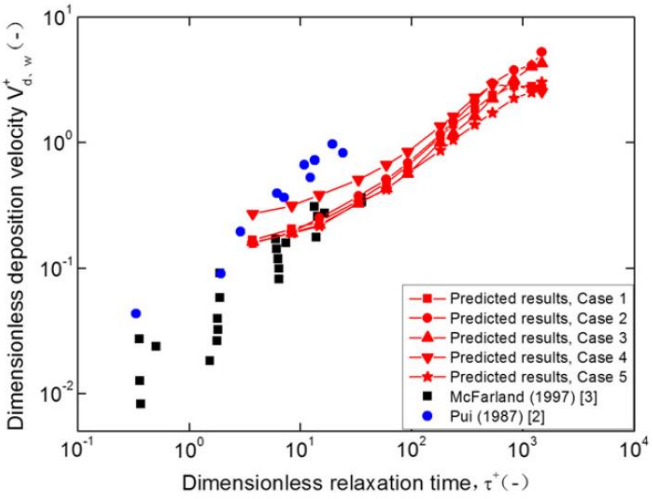
deposition velocity depends on the orientation of the duct, and the effects of gravity and inertia were more significant as the dimensionless relaxation time increases.

Figure 3.9 – Square duct bend: (a) grid, (b) particle deposition velocity and (c) duct orientation.

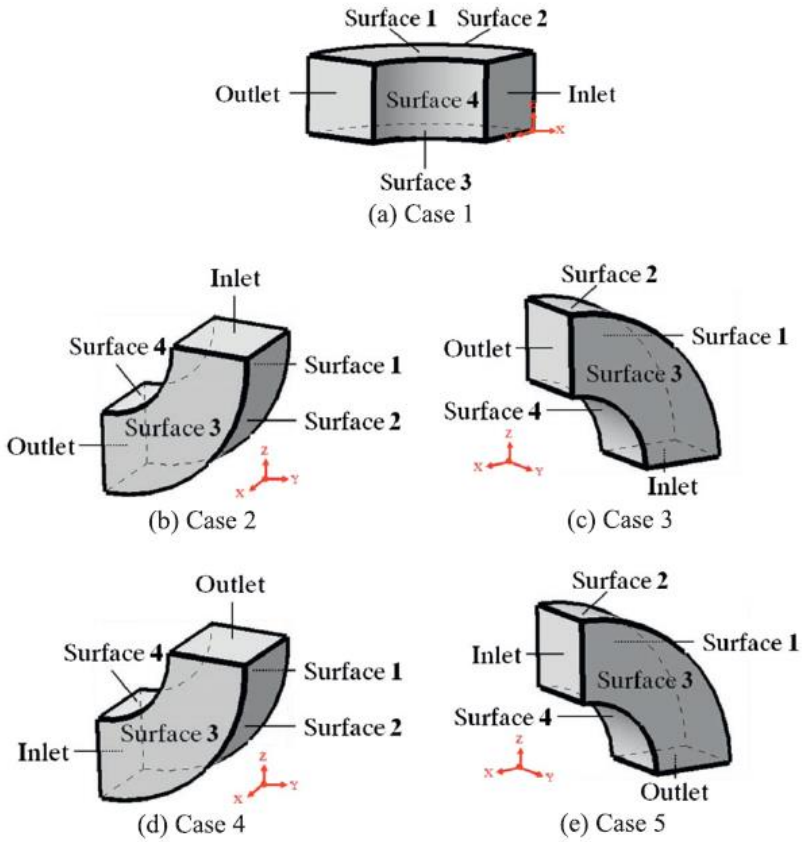
(a) Grid.



(b) Particle deposition velocity.



(c) Duct orientation.



NJOBUENWU *et al.* (2013) performed calculations of a diluted gas-solid flow in a 90° square duct bend, which was modeled using the Eulerian-Lagrangian approach with an RSM and random Fourier series to compute the instantaneous fluid velocity seen by the particles. They validated their results against KLIAFAS and HOLT (1987) experiments regarding the velocity and turbulence intensity for both particle and gas phases. The authors observed that their numerical results from 3D simulations compared favorably relative to previous works, regarding the used experimental data.

3.3 FINAL REMARKS OF LITERATURE REVIEW

The above literature review revealed that the detailed behavior of disperse two-phase flow in bends can be predicted reasonably well by the Eulerian-Lagrangian method, considering multiple combinations of a turbulence model and a Lagrangian particle tracking (LPT) model integrating a particle-wall interaction (PWI) model. However, the literature is not clear about the more accurate turbulence–LPT/PWI model combination to simulate the referred flow, neither about the contribution of each model on the improvement of the accuracy of the flow description. Therefore, one objective of this thesis is to contribute elucidating these issues by evaluating, in a systematic and quantitative way, the capabilities of different couples of turbulence–LPT/PWI models to predict particle-laden flow in a 90° bend.

In order to achieve this goal, the predictions of a particle-laden flow in a square-section 90° bend using different combinations of turbulence-LPT/PWI models are compared to the experimental data from KLIAFAS (1984) study. The literature review also revealed that the reliability of the methodology and the clarity of the results of the experimental studies of KLIAFAS (1984) make it an appropriate benchmark for validating CFD codes.

As indicated, in Chapter 1, this work is focused on HVAC duct systems whose principal characteristics of the airflow and aerosol through them are given in Table 3.5. Although experimental and numerical works regarding particle deposition in HVAC ducts are scarce, the literature review showed that some parameters that could affect this phenomenon have been studied numerically. Concerning the particular case of elbows, one can cite the studies of the duct surface material and the wall roughness (TIAN *et al.*, 2008; SUN *et al.*, 2012), the orientation (horizontal-to-vertical or vertical-to-horizontal) of the duct bend (GAO and LI, 2012), and the bend angle and the curvature ratio (MCFARLAND *et al.*, 1997). According to SUN *et al.* (2013), the studies of particle deposition in bends with rectangular cross-sections

(largely used in HVAC applications) are not fully understanding and most studies found in literature do not assume particle resuspension nor rebounding from duct surfaces (HENRY and MINIER, 2014).

Table 3.5 – HVAC applications.

Property			Range	Reference
Duct	Hydraulic diameter	D_h	0.1 ~ 1 m	Table 2.1
	Curvature ratio	δ	1 ~ 5	Equation (2.2)
	Bend angle	α_b	45° ~ 135°	Figure 2.2
Airflow	Average flow velocity	U_{ave}	5 ~ 10 m/s	-
	Reynolds number of fluid	Re	10^4 ~ 10^5	Equation (2.4)
	Darcy-Weisbach friction	f	0.01 ~ 0.05	Equation (2.11)
	Thickness of viscous sublayer	δ_{vs}	~ 0.01 D_h	Equation (2.14)
Aerosol	Particle diameter	d_p	0.01 ~ 100 μm	-
	Reynolds number of particle	Re_p	< 10^4	Equation (2.26)
	Stokes number	St	< 10	Equation (2.34)
	Particle relaxation time	τ_p	< 10^{-3} s	Equation (2.21)

In this way, another objective of this thesis is to numerically provide reliable information about dispersion and deposition of particles in square-sectioned ducts for HVAC applications, and compare the two-phase flow behavior with that in circular-sectioned ducts.

4 PHYSICAL AND MATHEMATICAL MODELLING

This chapter outlines the physical and mathematical aspects of the methodology employed for the studies presented in this dissertation. First, the governing equations used to solve the turbulent flow will be given, as well as the turbulence models explored to describe the flow field. The particle equation of motion will then be detailed, with emphasis on particle-fluid interaction, particle-wall interaction and particle dispersion in a turbulent flow.

4.1 TURBULENT FLOW

Turbulent flows are random according to time and space. Although many turbulent flows are steady in the time, the presence of fluctuations makes the analysis of turbulent flows a challenge, and if correlations are used, they rely on semi-empirical theories and on experimental data. Therefore, several kinds of turbulence models have been developed to solve as best as possible turbulent flows.

In this study, we focus on an incompressible, Newtonian, isothermal flow, submitted to gravity:

$$\vec{g} = -\vec{\nabla}\Psi \,, \tag{4.1}$$

where Ψ is the potential related to gravity. The Navier-Stokes equations that describe the instantaneous flow are:

$$\frac{\partial u_i}{\partial x_i} = 0 \,, \tag{4.2}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(u_i u_j\right) = -\frac{1}{\rho}\frac{\partial p^*}{\partial x_i} + \nu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) - \frac{\partial \Psi}{\partial x_i} \,, \tag{4.3}$$

where u_i is the instantaneous velocity component according to x_i direction, p^* is the pressure, ρ is the fluid density, ν is the kinematic viscosity of the fluid. If the following variable is introduced to calculate the pressure:

$$p = p^* + \rho\Psi \,, \tag{4.4}$$

the momentum conservation equation can be written as follows:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(u_i u_j\right) = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right). \quad (4.5)$$

Several methods can be used to calculate turbulent flows. The most accurate method is the Direct Numerical Simulation (DNS), solving the Navier-Stokes equations for all sizes of eddies. With this method, the grid size must be less than the smallest eddies (the Kolmogorov length scale), and very small time steps must be used. Such a method cannot be employed for complex configurations, and high Reynolds number flows. With the Large Eddy Simulation (LES) method, the large eddies are solved, and the interactions of the non-resolved small eddies with the large eddies are modeled. This method requires dense grids in the boundary layers.

The Reynolds Averaged Navier-Stokes (RANS) method consists in decomposing the flow variables into an average and a fluctuation and solving the time-averaged Navier-Stokes equations. In addition, further classification of turbulence modelling appeared in the last decades, such as the hybrid RANS-LES models. Among these models, one can notice the Detached Eddy Simulation (DES) model (SPALART *et al.*, 1997), where the RANS equations are solved in the boundary layer, and LES is applied away from walls. A switch function which is based on the grid enables to apply the RANS model close to the wall and the LES model far from the wall. Another hybrid RANS-LES model is the Scale Adaptive Simulation (SAS) model (MENTER and EGOROV, 2005) where the Von Karman length scale is introduced into the RANS model.

Therefore, the principal methods to calculate turbulent flows are RANS, LES, and DNS. Since the computational cost in LES is much higher than with RANS, only RANS models will be applied in this study. RANS models are based on the Reynolds decomposition of the flow variables:

$$\Phi = \bar{\Phi} + \Phi', \quad (4.6)$$

where Φ is the instantaneous variable, $\bar{\Phi}$ is the mean value of this variable and Φ' is the fluctuation of this variable. The mean value can be obtained with an ensemble average:

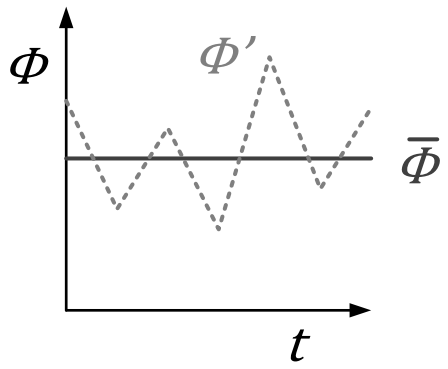
$$\bar{\Phi}(t) = \frac{1}{N}\sum_{n=1}^{N}\phi_n(t), \quad (4.7)$$

if the flow can be repeated N times, or with a time average:

$$\bar{\Phi} = \lim_{t^* \to +\infty} \frac{1}{t^*} \int_{-t^*}^{+t^*} \phi(t) dt. \tag{4.8}$$

The mean and the fluctuation of the instantaneous variable Φ is illustrated in Figure 4.1.

Figure 4.1 – Mean and fluctuation of a property according to time.



The averaged continuity and momentum equations are:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0\,, \tag{4.9}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{u}_i \bar{u}_j\right) = -\frac{\partial}{\partial x_i}\left(\frac{\bar{p}}{\rho}\right) + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial \bar{u}_i}{\partial x_j} + \tau_{ij}\right), \tag{4.10}$$

where $\bar{u}_i$ is the mean velocity in x_i direction, $\bar{p}$ is the mean pressure, and τ_{ij} is the Reynolds stress tensor, the goal of RANS turbulence modelling.

The RANS turbulence models can be sub-classified as:

- Linear EVM (Eddy Viscosity Model), in which the Reynolds tensor is modeled by a linear constitutive relationship with the mean velocity field:
  - Zero equation (algebraic model): no additional transport equation, only recommended for simple flow cases or as initial condition for another turbulence model;
  - One equation: solves one turbulent transport equation;
  - Two equations: widely applied to solve engineering flow problems, includes two extra transport equations to represent the turbulent properties of the flow;

- Non-linear EVM, in which the Reynolds tensor is modeled by a nonlinear relationship with the mean velocity field;
- RSM (Reynolds Stress Models), in which the individual components of the Reynolds tensor are directly computed, i.e., seven additional transport equations for three-dimensional problems (one for the turbulence length scale or an equivalent scale and six for the components of the Reynolds stress tensor).

The RANS models with two additional transport equations for the turbulent variables employ one equation for the turbulent kinetic energy and another for an additional variable that combines the turbulent kinetic energy with a length scale. The turbulent kinetic energy is given as:

$$k = \frac{1}{2}\overline{u_i{}'u_i{}'}\,, \tag{4.11}$$

where $\overline{u_i{}'u_i{}'}$ are the fluctuations of velocity components according to x, y, and z direction.

Many turbulence models are based on the Boussinesq hypothesis that models the turbulent Reynolds stress tensor. This model considers an analogy between the stress-strain relation for a Newtonian fluid and the turbulent stresses in a turbulent flow:

$$-\overline{u_i{}'u_j{}'} = \nu_t S_{ij}\,, \tag{4.12}$$

where ν_t is the turbulent kinematic viscosity and S_{ij} is the mean strain rate:

$$S_{ij} = \frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\,. \tag{4.13}$$

While the molecular kinematic viscosity is a fluid property, the turbulent kinematic viscosity is a flow property that indicates the local turbulence level and is calculated differently for each RANS turbulence model. The Boussinesq hypothesis works well for the flow near the solid surfaces, nevertheless, a more general definition was proposed by Kolmogorov for the linear EVM:

$$-\overline{u_i{}'u_j{}'} = \nu_t S_{ij} - \frac{2}{3}k\delta_{ij}\,, \tag{4.14}$$

where ν_t is the turbulent kinematic viscosity, k is the turbulent kinetic energy and δ_{ij} the Kronecker delta:

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}. \tag{4.15}$$

In non-linear EVM, additional terms are introduced into the Reynolds tensor:

$$-\overline{u_i' u_j'} = \nu_t S_{ij} - f\left(S_{ij}, \Omega_{ij}\right) - \frac{2}{3} k \delta_{ij} \,, \tag{4.16}$$

where $f\left(S_{ij}, \Omega_{ij}\right)$ is a non-linear relationship involving the mean strain rate, S_{ij}, and the mean vorticity, Ω_{ij}:

$$\Omega_{ij} = \frac{\partial \overline{u}_i}{\partial x_j} - \frac{\partial \overline{u}_j}{\partial x_i}. \tag{4.17}$$

The idea of the RSM is to calculate the Reynolds stress-strain from the solution of a set of stress transport equations that can be written in the form:

$$\frac{D\overline{u_i' u_j'}}{Dt} = P_{ij} + \phi_{ij} - \varepsilon_{ij} + d_{ij} \,, \tag{4.18}$$

where P_{ij} is the generation rate of the turbulent stress by mean strain (S_{ij}), d_{ij} is the diffusion rate of the turbulent stress by turbulent and viscous action, ϕ_{ij} and ε_{ij} are processes that are modeled in terms of other turbulent variables such as the Reynolds stress and mean strains[16].

4.1.1 RANS turbulence modelling

The principal objective of the turbulence modelling is to find an approximate solution for the Reynolds stress tensor and, consequently, to find an accurate flow field for a given application. As explained in the bibliographic study (Chapter 3) the behavior of particles in the vicinity of walls is influenced by small turbulence structures located close to the walls. Appropriate turbulence models must, therefore, be chosen.

[16] The term $\frac{D\overline{u_i' u_j'}}{Dt}$ is a material derivative (LAI *et al.*, 2010): $\frac{D\overline{u_i' u_j'}}{Dt} \equiv \frac{\partial \overline{u_i' u_j'}}{\partial t} + \frac{\partial}{\partial x_k}\left(\overline{u}_k \overline{u_i' u_j'}\right)$.

The RANS turbulence models can be also sub-classified as High Reynolds Number (HRN) models and Low Reynolds Number (LRN) models. While the HRN models apply wall functions and the meshes may be coarse because it is not necessary to discretize the inner layer ($30 \leq y^+ \leq 300$), the LRN models demand fine grids near the wall ($y^+ \leq 1$) to discretize the viscous sublayer ($y^+ = 5$), usually with at least 10 nodes in this region and with a low growth rate factor. The prediction of particle dispersion close to the wall demands fine grids near the wall.

The RANS turbulence models tested in this study are the *k-ε* of Launder and Sharma (LAUNDER and SHARMA, 1974), the *k-ε* of Lien and Leschziner (LIEN and LESCHZINER, 1993), the *k-ω* (WILCOX, 1988), the SST *k-ω* SST (MENTER, 1992), the v^2f (LIEN and KALITZIN, 2001), the cubic *k-ε* of Lien (LIEN *et al.*, 1996), the RSM LRR (LAUNDER *et al.*, 1975) and the RSM SSG (SPEZIALE *et al.*, 1991). Each one of these turbulence models will be briefly described as follows.

4.1.1.1 Launder and Sharma LRN *k-ε* model

Proposed by LAUNDER and SHARMA (1974) this model is a two-equation Linear EVM and the additional transport equations are:

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \varepsilon - 2\nu\left(\frac{\partial \sqrt{k}}{\partial x_j}\right)^2, \tag{4.19}$$

$$\frac{\partial \varepsilon}{\partial t} + \bar{u}_j \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + c_1 f_1 \frac{\varepsilon}{k} P_k - c_2 f_2 \frac{\varepsilon^2}{k} + 2\nu\nu_t\left(\frac{\partial^2 \bar{u}_i}{\partial {x_j}^2}\right), \tag{4.20}$$

where P_k is the production of turbulent kinetic energy:

$$P_k = \nu_t S_{ij}\left(\frac{\partial \bar{u}_i}{\partial x_j}\right). \tag{4.21}$$

The second variable, ε, is the dissipation rate of the turbulent kinetic energy, defined as:

$$\varepsilon = \nu\overline{\frac{\partial u_\iota'}{\partial x_J}\frac{\partial u_\iota'}{\partial x_J}} - 2\nu\left(\frac{\partial \sqrt{k}}{\partial y}\right)^2, \tag{4.22}$$

where y is the normal distance to the wall. This method enables to apply at the wall a boundary condition for the dissipation rate of the turbulent kinetic energy. This low Reynolds number k-ε model includes damping functions f_1 and f_μ.

The turbulent kinematic viscosity is defined as:

$$\nu_t = c_\mu f_\mu \frac{k^2}{\varepsilon}\,, \tag{4.23}$$

with:

$$f_\mu = e^{\frac{-3.4}{(1+Re_t/50)^2}}\,, \tag{4.24}$$

where the turbulent Reynolds number Re_t is given by:

$$Re_t = \frac{k^2}{\nu\varepsilon}\,. \tag{4.25}$$

The other damping function of the transport equation for the dissipation rate of the turbulent kinetic energy is:

$$f_2 = 1 - 0.3e^{-{Re_t}^2}\,, \tag{4.26}$$

and the other constants are given in Table 4.1.

Table 4.1 – Constants of the Launder and Sharma LRN k-ε turbulence model.

c_1	c_2	f_1	σ_k	σ_ε	c_μ
1.44	1.92	1.0	1.0	1.3	0.09

4.1.1.2 Lien and Leschziner LRN k-ε model

Proposed by LIEN and LESCHZINER (1993) this a two-equation Linear EVM and the additional transport equations are:

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \varepsilon\,, \tag{4.27}$$

$$\frac{\partial \varepsilon}{\partial t} + \bar{u}_j \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + c_1 f_1 \frac{\varepsilon}{k} P_k - c_2 f_2 \frac{\varepsilon^2}{k}\,. \tag{4.28}$$

where P_k is calculated according to Equation (4.21). For this low Reynolds number *k*-*ε* model, damping functions are introduced into the transport equation for *ε* and the turbulent viscosity is defined as:

$$\nu_t = c_\mu f_\mu \frac{k^2}{\varepsilon}\,, \tag{4.29}$$

where c_1, c_2 and c_μ are given by:

$$c_1 = 1.44\left(1 + \frac{{P_k}'}{P_k}\right), \tag{4.30}$$

$$c_2 = 1.92\left(1 - 0.3e^{-{Re_t}^2}\right), \tag{4.31}$$

$$c_\mu = 0.09\left(\frac{1-e^{-0.016y^*}}{1-e^{-0.263y^*}}\right), \tag{4.32}$$

with:

$${P_k}' = \frac{c_2 k^{3/2} e^{-0.00222{y^*}^2}}{3.53y\left(1-e^{-0.263y^*}\right)}, \tag{4.33}$$

$$y^* = \frac{y\sqrt{k}}{\nu}, \tag{4.34}$$

where y is the normal distance from the wall.

The constants of the above equations are shown in Table 4.2.

Table 4.2 – Constants of the Lien and Leschziner LRN *k*-*ε* turbulence model.

f_1	f_2	σ_k	σ_ε	c_μ
1.0	1.0	1.0	1.3	0.09

4.1.1.3 *k*-*ω* model

This model developed by WILCOX (1988) performs quite well in boundary layer flows. In this model the turbulent viscosity is calculated as follows:

$$\nu_t = \frac{k}{\omega}\,, \tag{4.35}$$

where the second variable ω is a frequency defined as:

$$\omega = \frac{\varepsilon}{k}\,. \tag{4.36}$$

The two additional transport equations for this model are:

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_t)\frac{\partial k}{\partial x_j}\right] + P_k - \beta_1 k\omega\,, \tag{4.37}$$

$$\frac{\partial \omega}{\partial t} + \bar{u}_j \frac{\partial \omega}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t)\frac{\partial \omega}{\partial x_j}\right] + \alpha \frac{\omega}{k} P_k - \beta_2 \omega^2\,. \tag{4.38}$$

The constants of the above equations are presented in Table 4.3.

Table 4.3 – Constants of the *k-ω* turbulence model.

β_1	β_2	σ_k	σ_ω	α
0.09	0.075	0.5	0.5	5/9

4.1.1.4 *k-ω* SST model

Developed by MENTER (1992), the *k-ω* SST turbulence model has good behavior in adverse pressure gradients and separating flows, however, it produces a large turbulence level in the region with the large normal strain (e.g., stagnation regions and regions with strong acceleration). The coefficients of this model were updated by MENTER *et al.* (2003).

The two additional transport equations for this model are:

$$\frac{\partial k}{\partial t} + \bar{u}_i \frac{\partial k}{\partial x_i} = \tilde{P}_k - \beta^* k\omega + \frac{\partial}{\partial x_i}\left[(\nu + \sigma_k \nu_t)\frac{\partial k}{\partial x_i}\right], \tag{4.39}$$

$$\frac{\partial \omega}{\partial t} + \bar{u}_i \frac{\partial \omega}{\partial x_i} = \alpha {S_{ij}}^2 - \beta\omega^2 + \frac{\partial}{\partial x_i}\left[(\nu + \sigma_\omega \nu_t)\frac{\partial \omega}{\partial x_i}\right] + 2(1 - F_1)\sigma_{w2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i}\,. \tag{4.40}$$

All constants (α, β, σ_k and σ_ω) are computed by a blend from the corresponding constants of the *k-ε* turbulence model (applied to the core of the flow) (α_1, β_1, σ_{k1} and $\sigma_{\omega 1}$) and the *k-ω* turbulence model (applied to the vicinity of the walls) (α_2, β_2, σ_{k2} and $\sigma_{\omega 2}$) via the following relationship:

$$\alpha = \alpha_1 F_1 + \alpha_2 (1 - F_1) \, . \tag{4.41}$$

The first blending function F_1 that enables to switch between the *k-ω* model in the vicinity of the wall and the *k-ε* model far from the wall is defined as:

$$F_1 = tanh\left\{\left[min\left(max\left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega}\right) \, , \, \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} y^2}\right)\right]^4\right\}, \tag{4.42}$$

where y is the distance to the nearest wall and $CD_{k\omega}$ is:

$$CD_{k\omega} = max\left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \, , \, 10^{-10}\right) . \tag{4.43}$$

The turbulent viscosity is:

$$\nu_t = \frac{a_1 k}{max(a_1 \omega \, , b_1 F_2 S)} \, , \tag{4.44}$$

where:

$$S = \sqrt{\frac{1}{2} S_{ij} S_{ij}} \, , \tag{4.45}$$

and the second blending function F_2 is defined as:

$$F_2 = tanh\left\{\left[max\left(\frac{2\sqrt{k}}{\beta^* \omega y} \, , \frac{500\nu}{y^2 \omega}\right)\right]^2\right\} . \tag{4.46}$$

To prevent the growth of turbulence in the stagnation region, the production limiter $\tilde{P}_k$ is used:

$$\tilde{P}_k = min(P_k \, , 10\beta^* k\omega) \, . \tag{4.47}$$

The constants above are presented in Table 4.4. These constants follow Equation (4.41), i.e., to calculate σ_k of Equation (4.39).

Table 4.4 – Constants of the *k-ω* SST turbulence model.

β^*	σ_{k1}	σ_{k2}	$\sigma_{\omega1}$	$\sigma_{\omega2}$	β_1	β_2	α_1	α_2	a_1	b_1
0.09	0.85	1.0	0.5	0.856	0.075	0.0828	5/9	0.44	0.31	1.0

4.1.1.5 v^2f model

DURBIN (1991) showed that in the vicinity of the wall, the turbulent transport, which is strongly anisotropic, does not scale with the turbulent kinetic energy, but with $\overline{v'^2}$ (the velocity fluctuation perpendicular to the wall or to the streamlines). He proposed a turbulence model that does not include damping functions for the turbulent viscosity, but which includes $\overline{v'^2}$ (DURBIN, 1995). This model has proven to perform satisfactorily for predicting particle deposition in turbulent flows (ZHANG and CHEN, 2009; MAJLESARA *et al.*, 2013), this is the reason why it was selected for this study. A modified version of the v^2f model, which is more stable than the original model was used in this study (LIEN and KALITZIN, 2001; DAVIDSON *et al.*, 2003). The turbulent viscosity is calculated with:

$$\nu_t = min\left(C_{\mu k\varepsilon}\frac{k^2}{\varepsilon}\ ,\ C_\mu\overline{v'^2}T_k\right), \tag{4.48}$$

where T_k is the turbulence time scale given by:

$$T_k = max\left(\frac{k}{\varepsilon}\ ,\ C_T\sqrt{\frac{\nu}{\varepsilon}}\right). \tag{4.49}$$

This model solves equations for the turbulent kinetic energy and its dissipation rate:

$$\frac{\partial k}{\partial t} + \bar{u}_j\frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \varepsilon\ , \tag{4.50}$$

$$\frac{\partial \varepsilon}{\partial t} + \bar{u}_j\frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{C_{\varepsilon1}P_k - C_{\varepsilon2}\varepsilon}{T}\ , \tag{4.51}$$

and $\overline{v'^2}$ is calculated with the following transport equation:

$$\frac{\partial \overline{v'^2}}{\partial t} + \overline{u}_j \frac{\partial \overline{v'^2}}{\partial x_j} = kf - \overline{v'^2}\frac{\varepsilon}{k} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \overline{v'^2}}{\partial x_j}\right], \tag{4.52}$$

where the source term f is obtained with the following equation:

$$L^2 \frac{\partial^2 f}{\partial x_j \partial x_j} - f = \frac{1}{T}\left[(C_1 - 6)\frac{\overline{v'^2}}{k} - \frac{2}{3}(C_1 - 1)\right] - C_2 \frac{P_k}{k}. \tag{4.53}$$

In the previous equation, L is the turbulent length scale given by:

$$L = C_L max\left[\frac{k^{3/2}}{\varepsilon},\ C_\eta \left(\frac{\nu^3}{\varepsilon}\right)^{1/4}\right]. \tag{4.54}$$

The constants of the v^2f model are presented in Table 4.5.

Table 4.5 – Constants of the v^2f turbulence model.

C_μ	$C_{\mu k\varepsilon}$	$C_{\varepsilon 2}$	σ_k	σ_ε	C_1	C_2	C_L	C_η
0.22	0.09	1.9	1.0	1.3	1.4	0.3	0.23	70

4.1.1.6 Lien cubic *k-ε* model

Linear eddy viscosity models fail to calculate complex flows such as flows in curved ducts. To overcome these difficulties, non-linear eddy viscosity models have been developed. The cubic non-linear eddy viscosity developed by LIEN *et al.* (1996) was tested for a curved channel. It is written as:

$$\begin{aligned}\frac{\overline{u_i' u_j'}}{k} = \frac{2}{3}\delta_{ij} - \frac{\nu_t}{k}S_{ij} + C_1\frac{\nu_t}{\varepsilon}\left(S_{ik}S_{kj} - \frac{1}{3}\delta_{ij}S_{kl}S_{kl}\right) + C_2\frac{\nu_t}{\varepsilon}\left(\Omega_{ik}S_{kj} + \Omega_{jk}S_{ki}\right) + \\ + C_3\frac{\nu_t}{\varepsilon}\left(\Omega_{ik}\Omega_{jk} - \frac{1}{3}\delta_{ij}\Omega_{kl}\Omega_{kl}\right) + C_4\nu_t\frac{k}{\varepsilon^2}\left(S_{ki}\Omega_{lj} + S_{kj}\Omega_{li}\right)S_{kl} + \\ + C_5\nu_t\frac{k}{\varepsilon^2}\left(S_{kl}S_{kl} - \Omega_{kl}\Omega_{kl}\right)S_{ij},\end{aligned} \tag{4.55}$$

where the coefficients are:

$$C_1 = \frac{3/4}{1000 + S^3}, \tag{4.56}$$

$$C_2 = \frac{15/4}{1000+S^3}\,, \tag{4.57}$$

$$C_3 = \frac{19/4}{1000+S^3}\,, \tag{4.58}$$

$$C_4 = -10{C_\mu}^2\,, \tag{4.59}$$

$$C_5 = -2{C_\mu}^2\,, \tag{4.60}$$

with:

$$S = \frac{k}{\varepsilon}\sqrt{\frac{1}{2}S_{ij}S_{ij}}\,, \tag{4.61}$$

$$C_\mu = \frac{0.667}{A_1+S+0.9\Omega}\,, \tag{4.62}$$

and:

$$A_1 = 1.25, \tag{4.63}$$

$$\Omega = \frac{k}{\varepsilon}\sqrt{\frac{1}{2}\Omega_{ij}\Omega_{ij}}\,. \tag{4.64}$$

The mean strain rate, S_{ij}, and the mean vorticity, Ω_{ij}, were defined in Equation (4.13) and Equation (4.17), respectively.

The turbulent kinematic viscosity of the Lien cubic *k*-ε model is defined as:

$$\nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon}\,, \tag{4.65}$$

where the damping function, f_μ, is calculated as:

$$f_\mu = \left(1 - e^{-0.0198y^*}\right)(1 + 5.29/y^*)\,, \tag{4.66}$$

where the dimensionless parameter y^* was defined in Equation (4.34).

In the transport equation of ε, the coefficients C_1 and C_2 are calculated as in the linear LRN *k*-ε model of LIEN and LESCHZINER (1993) (Equations (4.30) and (4.31)).

4.1.1.7 RSM LRR

The Reynolds Stress Models (RSM) have been developed to predict in a better way non-isotropic turbulent flow. It has been shown that RSM performs well for flows in curved ducts (GIBSON *et al.*, 1981; LUO and LAKSHMINAMYANA, 1997). In addition, an advantage of these models is that the terms related to turbulence production are not modeled, they are calculated exactly. The Reynolds Stress equation of the RSM developed by Launder, Reece, and Rodi (LRR) (LAUNDER *et al.*, 1975) is:

$$\frac{D\overline{u_i{}'u_j{}'}}{Dt} = -\left(\overline{u_j{}'u_k{}'}\frac{\partial \bar{u}_i}{\partial x_k} + \overline{u_i{}'u_k{}'}\frac{\partial \bar{u}_j}{\partial x_k}\right) - \frac{2}{3}\varepsilon\delta_{ij} + \Phi_{ij} + \\ + C_s\frac{\partial}{\partial x_k}\left[\frac{k}{\varepsilon}\left(\overline{u_i{}'u_l{}'}\frac{\partial \overline{u_j{}'u_k{}'}}{\partial x_l} + \overline{u_j{}'u_l{}'}\frac{\partial \overline{u_k{}'u_i{}'}}{\partial x_l} + \overline{u_k{}'u_l{}'}\frac{\partial \overline{u_i{}'u_j{}'}}{\partial x_l}\right)\right], \quad (4.67)$$

in this equation, the pressure strain term is modeled as:

$$\Phi_{ij} = C_1 a_{ij}\varepsilon + C_2 k S_{ij} + C_3 k\left(S_{ik}a_{kj} + a_{ik}S_{kj} - \frac{2}{3}S_{kl}a_{lk}\delta_{ij}\right) + \\ + C_4 k\left(\Omega_{ik}a_{kj} - \Omega_{kj}a_{ik}\right), \quad (4.68)$$

where:

$$a_{ij} = \frac{1}{2k}\left(\overline{u_i{}'u_j{}'} - \frac{2}{3}k\delta_{ij}\right), \quad (4.69)$$

is the anisotropic tensor, and:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right), \quad (4.70)$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial \bar{u}_j}{\partial x_i}\right). \quad (4.71)$$

To calculate ε, the following equation is solved:

$$\frac{D\varepsilon}{Dt} = C_\varepsilon\frac{\partial}{\partial x_k}\left(\frac{k}{\varepsilon}\overline{u_k{}'u_l{}'}\frac{\partial \varepsilon}{\partial x_l}\right) - C_{\varepsilon 1}\frac{\varepsilon}{k}\overline{u_i{}'u_k{}'}\frac{\partial \bar{u}_i}{\partial x_k} - C_{\varepsilon 2}\frac{\varepsilon^2}{k}. \quad (4.72)$$

The constants of this RSM are given in Table 4.6.

Table 4.6 – Constants of the RSM LRR turbulence model.

C_S	C_1	C_2	C_3	C_4	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	C_ε
0.11	-3.6	0.8	1.2	1.2	1.44	1.90	0.15

4.1.1.8 RSM SSG

The RSM developed by Speziale, Sarkar, and Gatski (SSG) (SPEZIALE *et al.*, 1991) differs from the RSM LRR for the method to obtain the pressure-strain correlation. In this model, a quadratic non-linear term depending on the anisotropic tensor a_{ij} is added. The pressure-strain correlation model of the RSM SSG is:

$$\Phi_{ij} = C_1 a_{ij}\varepsilon + C_2\varepsilon\left({a_{ij}}^2 - \frac{1}{3}{a_{kk}}^2\delta_{ij}\right) + C_3 k S_{ij} + \\ + C_4 k\left(S_{ik}a_{kj} + a_{ik}S_{kj} - \frac{2}{3}S_{kl}a_{lk}\delta_{ij}\right) + C_5 k\left(\Omega_{ik}a_{kj} - \Omega_{kj}a_{ik}\right), \quad (4.73)$$

where:

$$C_1 = -3.4 - 1.8\frac{P}{\varepsilon}, \quad (4.74)$$

$$C_3 = 0.8 - 1.3\sqrt{A}, \quad (4.75)$$

with:

$$P = -\overline{{u_i}'{u_j}'}\frac{\partial \bar{u}_j}{\partial x_i}, \quad (4.76)$$

and:

$$A = a_{ij}a_{ji}. \quad (4.77)$$

The constants of this RSM are given in Table 4.7.

Table 4.7 – Constants of the RSM SSG turbulence model.

C_2	C_4	C_5
4.2	1.25	0.4

4.2 PARTICLE TRANSPORT

The Lagrangian approach with one-way coupling for the prediction of particle transport and deposition on the dilute flow is the basis of this study and will be explained next.

4.2.1 Particle equation of motion

To predict particle transport flows, two approaches can be used: the Eulerian approach, and the Lagrangian approach. In the Eulerian approach, the particle phase and the gas phase are treated as interpenetrating continua. A continuity equation and a momentum conservation equation are solved for the dispersed phase, as it is the case for the carrier phase. Even if this method is computationally efficient, one of the drawbacks of this method is that it is not easy to apply boundary conditions for the particles at the walls (the particle rebound at the wall).

The Lagrangian approach consists in solving the particle equation of motion for each particle in the flow. For indoor air quality problems, the particles are very small, aerosols are dilute, and particles are considered as points. Many particles are injected and tracked in the flow, and averaged quantities such as the particle velocities, the deposition velocities, etc.… are calculated.

The general mathematical description of the motion of a sphere in a quiescent fluid is governed by the BBO (Boussinesq, Basset, and Oseen) equation (FAN and ZHU, 2005). This equation was then extended to the motion of a sphere in a moving fluid by TCHEN (1947), and MAXEY and RILEY (1983), such as,

$$\rho_p \frac{\pi}{6} {d_p}^3 \frac{d\vec{u}_p}{dt} = \frac{\pi}{6} {d_p}^3 \left(\rho_p - \rho\right)\vec{g} - C_d \frac{1}{2} \rho A_p \left|\vec{u}_p - \vec{u}\right|\left(\vec{u}_p - \vec{u}\right) + \rho \frac{\pi}{6} {d_p}^3 \frac{D\vec{u}}{Dt} +$$

(I) (II) (III)

$$+ C_a \rho \frac{\pi}{6} {d_p}^3 \left(\frac{D\vec{u}}{Dt} - \frac{d\vec{u}_p}{dt}\right) + C_h {d_p}^2 \sqrt{\pi\rho\mu} \int_0^t \frac{\frac{D\vec{u}}{Dt} - \frac{d\vec{u}_p}{dt}}{\sqrt{t-t'}} dt' + \vec{F}_e \,. \quad (4.78)$$

(IV) (V) (VI)

4.2.2 Particle-fluid interactions

Previous studies of wall-bounded particle-laden turbulent flow have shown that the choice of the primary forces acting on the particles is not universal. The forces responsible for the particle transport in the flow are known as hydrodynamic forces (CROWE *et al.*, 2012) such as the drag and the induced shear-lift forces[17]. The hydraulic forces such as the Basset history, virtual mass, Faxén correction, and pressure gradient can be neglected because the density of the particle is much higher than the density of fluid ($\rho_p/\rho \gg 1$), as it is the case for particles in ventilation duct systems. Since the particle-laden flow in ventilation ducts is diluted, three principal particle-fluid interactions were applied in this study: the gravity force, the drag force and the shear-lift force. These forces are detailed in the sequence.

4.2.2.1 Gravity force

The first term of the right-hand side (the term (I)) of the BBO equation (Equation (4.78)) is the gravity force. In this term, $\vec{g}$ is the gravity acceleration, ρ_p is the particle density and ρ is the fluid density.

4.2.2.2 Drag force

The second term (the term (II)) of the Equation (4.78) is the drag force, it is the resistance of the fluid to the particle motion. In this term, C_d is the drag coefficient, A_p is the cross-sectional area of the particle:

$$A_p = \frac{\pi}{4} {d_p}^2 , \tag{4.79}$$

$\vec{u}$ is the instantaneous fluid velocity at the particle location, and $\vec{u}_p$ is the particle velocity. There are many correlations in the literature to determine the drag coefficient of a smooth sphere (CLIFT *et al.*, 1978). Figure 4.2 shows the drag diagram of a smooth sphere in which at $Re_p \approx 3.5 \times 10^5$ the drag coefficient decreases sharply due to the transition from laminar to turbulent boundary layer around the sphere. Stokes' law is as follows:

[17] The lift developed due the rotation of particle is called the Magnus force that can be neglected for aerosols in ventilation systems because $\rho_p/\rho \gg 1$.

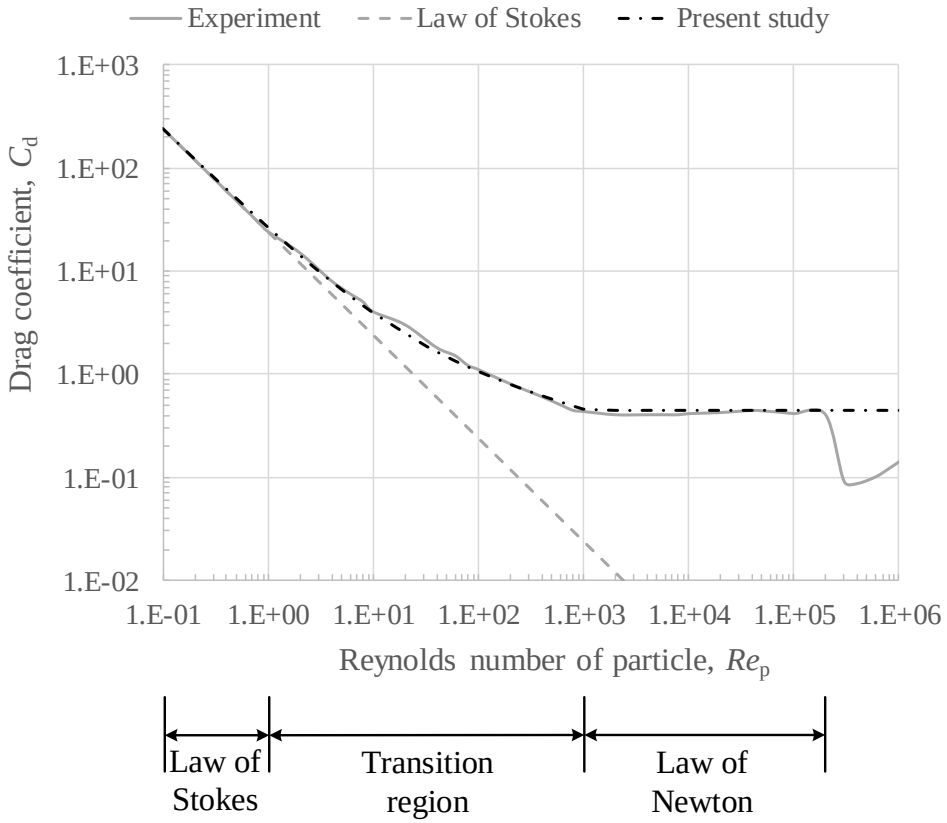
$$C_d = \frac{24}{Re_p}, \tag{4.80}$$

in the present study, the following correlations were used for the drag force:

$$C_d = \begin{cases} \frac{24}{Re_p}\left(1 + \frac{{Re_p}^{2/3}}{6}\right), & \text{for } Re_p \leq 10^3 \\ 0.424, & \text{for } Re_p > 10^3 \end{cases}. \tag{4.81}$$

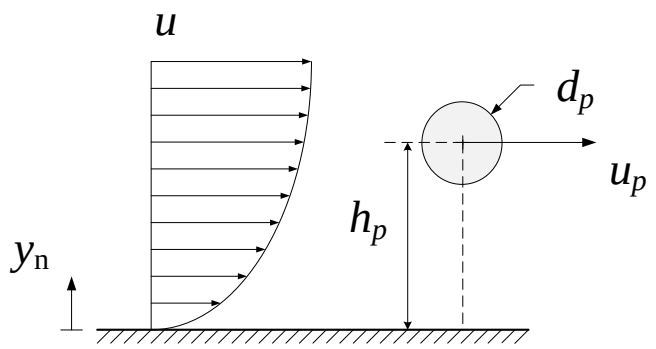
where Re_p is the particle Reynolds number (Equation (2.26)). For a very small Reynolds number, Stokes' law is recovered.

Figure 4.2 – Drag coefficient of a smooth sphere (the standard drag curve).



The drag coefficient can be affected by the presence of a wall (ARCEN *et al.*, 2006; JIN *et al.*, 2016). Several correlations have been proposed to correct the drag coefficient. Figure 4.3 illustrates a particle of diameter d_p and at a normal distance h_p (in the y_n direction) suspended in a fluid with velocity u and moving with a particle velocity u_p tangential to the wall.

Figure 4.3 – Particle moving parallel to the wall.



For a particle moving parallel to a plane wall at a low velocity, FAXÉN (1923) proposed the following correlation:

$$C_d = \left(1 - \frac{9}{16}k_{pw} + \frac{1}{8}{k_{pw}}^3 - \frac{45}{256}{k_{pw}}^4 - \frac{1}{16}{k_{pw}}^5\right)^{-1} \frac{24}{Re_p}, \tag{4.82}$$

with:

$$k_{pw} = \frac{d_p/2}{h_p}. \tag{4.83}$$

ZENG *et al.* (2009) proposed a correlation that can be applied to particles moving parallel to a plane wall at higher Reynolds numbers:

$$C_d = \left[1 + 0.15\left(1 - e^{-\sqrt{\lambda}}\right){Re_p}^{0.687+0.31e^{-2\sqrt{\lambda}}}\right] C_{d0}, \tag{4.84}$$

with:

$$C_{d0} = \left[1.028 - \frac{0.07}{1+4\lambda^2} - \frac{8}{15}ln\left(\frac{270\lambda}{135+256\lambda}\right)\right]\frac{24}{Re_p}, \tag{4.85}$$

and:

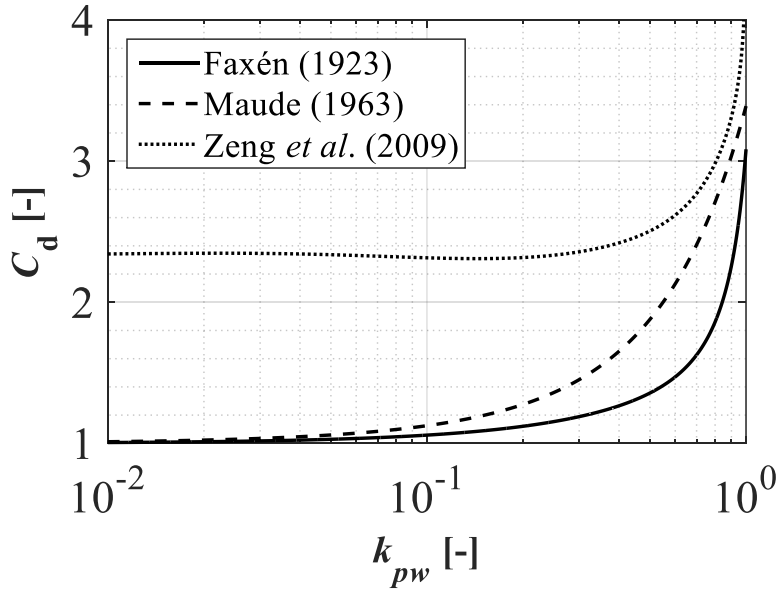
$$\lambda = \frac{1}{2}\left(\frac{1}{k_{pw}} - 1\right). \tag{4.86}$$

For a particle moving normal to the wall, the correction of the drag coefficient was derived by MAUDE (1963):

$$C_d = \left[1 + \frac{9}{8}k_{pw} + \left(\frac{9}{8}k_{pw}\right)^2\right]\frac{24}{Re_p}. \tag{4.87}$$

The correction of the drag coefficient is presented in Figure 4.4 for $Re_p = 24$. We observe that the drag coefficient for the Faxén and Maude correlations is not affected for large distances from the wall ($h_p > 50d_p$ or $k_{pw} < 0.01$), however near the wall ($h_p \approx d_p$ or $k_{pw} \approx 0.5$) the drag force is twice as large as the drag force without any correction. According to RIZK and ELGHOBASHI (1985), the Faxén correlation is excellent up to $k_{pw} = 0.765$. The ZENG *et al.* (2009) correlation affected the drag force for large distances from the wall and near the wall the drag force is higher than the other correlations presented.

Figure 4.4 – Near wall correction models of the drag coefficient.



4.2.2.3 Pressure gradient force

The third term of the right-hand side of the particle equation of motion, i.e., the term (IV) of the Equation (4.78), is the force due to the pressure gradient around the moving particle. In this term, $\frac{D\vec{u}}{Dt}$ is the material derivative:

$$\frac{D\vec{u}}{Dt} = \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \, . \tag{4.88}$$

4.2.2.4 Added mass force

The fourth term of the right-hand side of the particle equation of motion (term (IV) of the Equation (4.78)) is the force due to the added mass of the particle, it accelerates the mass of fluid displaced by the particle. In this expression, C_a is a parameter that is given by correlations.

4.2.2.5 Basset force

The fifth term of the right-hand side (term (V) of the Equation (4.78)) is the Basset force which is due to the temporal development of the particle wake. In this term, C_h is a coefficient that can be calculated with correlations, and μ is the dynamic fluid viscosity.

4.2.2.6 Lift force

The last term of the right-hand side of the particle equation, i.e., the term (VI) of the Equation (4.78), represents the external forces that act on the particle. In this study, the lift force and the Brownian force will be considered.

The shear-lift force is a force that is perpendicular to the flow direction. It is due to inertial effects in the viscous flow around the particle as illustrated in Figure 4.5. In this figure, a flow with a local velocity u surrounding a particle of velocity u_p defects the trajectory of the particle (x_p) because of the shear-lift force ($F_{l(Saff)}$). An expression for the shear lift force on a sphere was given by SAFFMAN (1965, 1968):

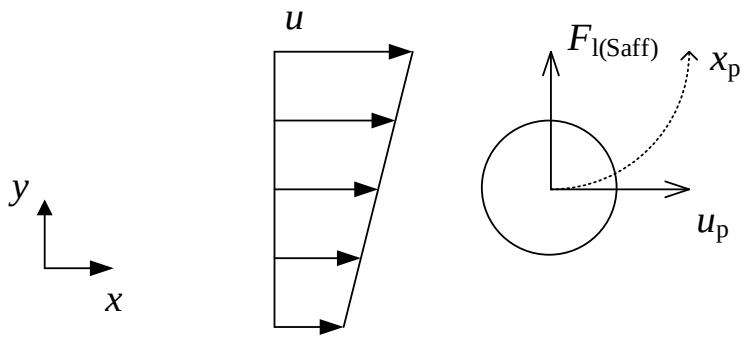
$$F_{l(Saff)} = 1.615 \rho \nu^{1/2} {d_p}^2 u_{p,r} |G|^{1/2} sgn(G) \, , \tag{4.89}$$

where $u_{p,r}$ is the particle relative velocity and G is the shear rate defined as:

$$G = \frac{\partial u}{\partial y} \, , \tag{4.90}$$

where y is the normal coordinate to the flow direction and u is the fluid velocity at the particle location.

Figure 4.5 – Representation of the shear lift force.



Equation (4.89) is only valid for the following constraints:

$$Re_p \ll 1,$$

$$Re_G = \frac{|G| {d_p}^2}{\nu} \ll 1\,,$$

$$Re_{\omega_p} = \frac{\omega_p {d_p}^2}{\nu} \ll 1\,,$$

$$Re_p \ll \sqrt{Re_G}\,,$$

where ω_p is the rotational speed of the particle.

The Saffman lift force was extended by other authors to higher Reynolds numbers (McLAUGHLIN, 1991; MEI, 1992). Among them the empirical correlation of MEI (1992) that was applied in this study, and that is written for a three-dimensional flow as:

$$\vec{F}_l = 1.615 \rho \nu^{1/2} {d_p}^2 \left[\left(\vec{u} - \vec{u}_p\right) \times \vec{\omega}\right] \left(\frac{1}{|\vec{\omega}|}\right)^{1/2} C_l\,, \tag{4.91}$$

where $\vec{\omega}$ is the local vorticity of the fluid and the coefficient C_l is:

$$C_l = \begin{cases} \left(1 - 0.3314\alpha^{1/2}\right) e^{-0.1 Re_p} + 0.3314\alpha^{1/2}\,, & \text{for } Re_p \leq 40 \\ 0.0524\left(\alpha Re_p\right)^{1/2}\,, & \text{for } Re_p > 40 \end{cases}, \tag{4.92}$$

with α being the non-dimensional shear rate:

$$\alpha = \frac{d_p}{2} \frac{|\vec{\omega}|}{|\vec{u} - \vec{u}_p|}\,. \tag{4.93}$$

The near-wall correction of the shear lift force was proposed by WANG *et al.*, (1997). The regions of applicability of this correction are specified by the Saffman length scale, L_G, and the Stokes length scale, L_S, defined, respectively, as:

$$L_G = \sqrt{\frac{\nu}{|G|}}\,, \tag{4.94}$$

$$L_S = \frac{\nu}{|u_{p,s}|}\,. \tag{4.95}$$

and also by two dimensionless parameters:

$${h_p}^+ = \frac{h_p}{L_G}\,, \tag{4.96}$$

$$\varepsilon_p = sgn(G)\frac{\sqrt{|G|\nu}}{u_{p,s}}\,, \tag{4.97}$$

where $u_{p,s}$ is the streamwise local relative particle velocity component.

For $h_p \le min\{L_S, L_G\}$ the shear lift force with near-wall correction, F_l is (this force is normal to the wall):

$$\frac{F_l}{\rho\left(d_p/2\right)^2 {u_{p,s}}^2} = 1.7716 + 0.2160k_{pw} - 0.7292{k_{pw}}^2 + 0.4854{k_{pw}}^3 + \\ -\left(3.2397{k_{pw}}^{-1} + 1.1450 + 2.0840k_{pw} - 0.9059{k_{pw}}^2\right)\alpha + \\ +\left(2.0069 + 1.0575k_{pw} - 2.4007{k_{pw}}^2 + 1.3174{k_{pw}}^3\right)\alpha^2\,. \tag{4.98}$$

For $h_p > min\{L_S, L_G\}$ the shear lift force with near-wall correction is:

$$F_l = -\frac{9}{\pi}{C_l}^*\rho\nu^{1/2}{d_p}^2 u_{p,r}|G|^{1/2}sgn(G)\,, \tag{4.99}$$

with:

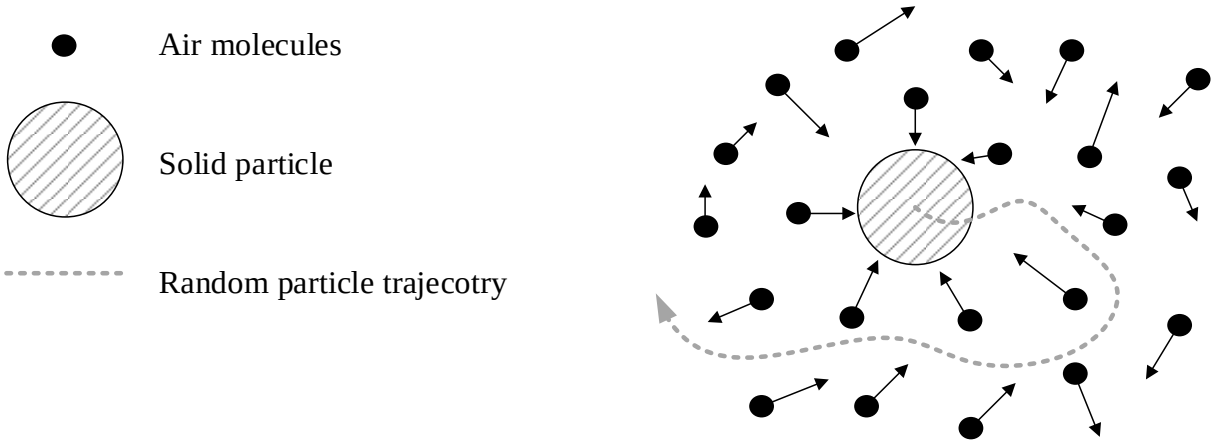
$${C_l}^* = {C_{l,u}}^* + {C_{l,w}}^*\,, \tag{4.100}$$

where $C_{l,u}{}^{*}$ is the lift coefficient correction due to shear-induced lift force and $C_{l,w}{}^{*}$ is the lift coefficient correction due to the wall-induced lift force. The procedure to determinate these coefficients is extensive and the complete procedure can be found in WANG *et al.*, (1997).

4.2.2.7 Brownian diffusion

Small particles ($d_p < 1\mu m$) are affected by the influence of the Brownian motion, i.e., the impact of the air molecules moving randomly changes the aerosol trajectory as illustrated in Figure 4.6.

Figure 4.6 – Brownian motion of a solid particle suspended in the air.



The effects of the Brownian random forces can be included as an additional force term in the particle equation of motion. The Brownian force, F_B, per unit of mass of the particle can be modeled as (LI and AHMADI, 1992):

$$F_B = \rho_p \frac{\pi}{6} {d_p}^3 \zeta \sqrt{\frac{\pi S_0}{\Delta t}}, \tag{4.101}$$

where ζ is a zero-mean, unit-variance-independent Gaussian random number, Δt is the time step size in which the amplitude of Brownian forces components is evaluated and S_0 is the spectral density of a Gaussian white noise random process given as:

$$S_0 = \frac{216 \nu k_B T}{\pi^2 \rho {d_p}^5 \left(\frac{\rho_p}{\rho}\right)^2 C_c}, \tag{4.102}$$

where ν is the kinematic viscosity of the fluid, T is the temperature, ρ is the density of the fluid, ρ_p is the density of the particle, d_p is the diameter of the particle, C_c is the Stokes-Cunningham slip correction factor (Equation (2.22)) and k_B is the Boltzmann constant.

4.2.2.8 Forces considered in this study

By dividing the particle equation of motion (Equation (4.78)) by $\rho_p \frac{\pi}{6} {d_p}^3$, we can select the forces that will be considered in this study. Since the particles in HVAC systems are heavier than the fluid, we have: $\rho/\rho_p \ll 1$. Hence, the gravity force, the drag force, and the Brownian that cannot be neglected for small particles will be retained for this study. Moreover, even if the lift force scales in this modified particle equation with ρ/ρ_p, the lift force will be considered in this work because it has been shown in the literature that it influences the behavior of particles near the wall, and it must be taken into account (UIJTTEWAAL and OLIEMANS, 1996; MARCHIOLI *et al.*, 2007). The following equation will thus be solved:

$$\rho_p \frac{\pi}{6} {d_p}^3 \frac{d\vec{u}_p}{dt} = \vec{F}_g + \vec{F}_d + \vec{F}_l + \vec{F}_B \ , \tag{4.103}$$

where $\vec{F}_g$, $\vec{F}_d$, $\vec{F}_l$ and $\vec{F}_B$ are, respectively, the gravity, drag, lift and Brownian forces. Integrating this equation according to time yields the particle velocity. The particle position $\vec{x}_p$ will then be obtained by integrating according to time the following equation:

$$\frac{d\vec{x}_p}{dt} = \vec{u}_p \ . \tag{4.104}$$

The angular velocity of each particle can be obtained as follows[18]:

$$I_p \frac{d\vec{\omega}_p}{dt} = \vec{T}_p \ , \tag{4.105}$$

where $\vec{\omega}_p$ is the angular velocity of a particle, $\vec{T}_p$ is the torque and I_p is the moment of inertia of the particle given by:

[18] The angular velocity is not relevant in the case of this study, neither the particles forces related to the rotational motion such as the Magnus force because the particle are material points instead of rigid bodies.

$$I_p = \frac{1}{10} m_p {d_p}^2 \ . \tag{4.106}$$

The interactions between the particles and the fluid flow have an important impact on the deposition rates and they can be classified as follows:

- the particle-fluid interaction;
- the particle-particle interaction;
- the particle-wall interaction.

For diluted flows, the particle-particle interactions can be neglected, therefore they are not explained in this study. The particle-fluid interaction of diluted flows is the one-way coupled, i.e., for a given fluid flow the fluid motion acts on the particle motion but the particle motion does not change the fluid flow field.

4.2.3 Particle-wall interactions

In general, interactions with walls are caused by two mechanisms:

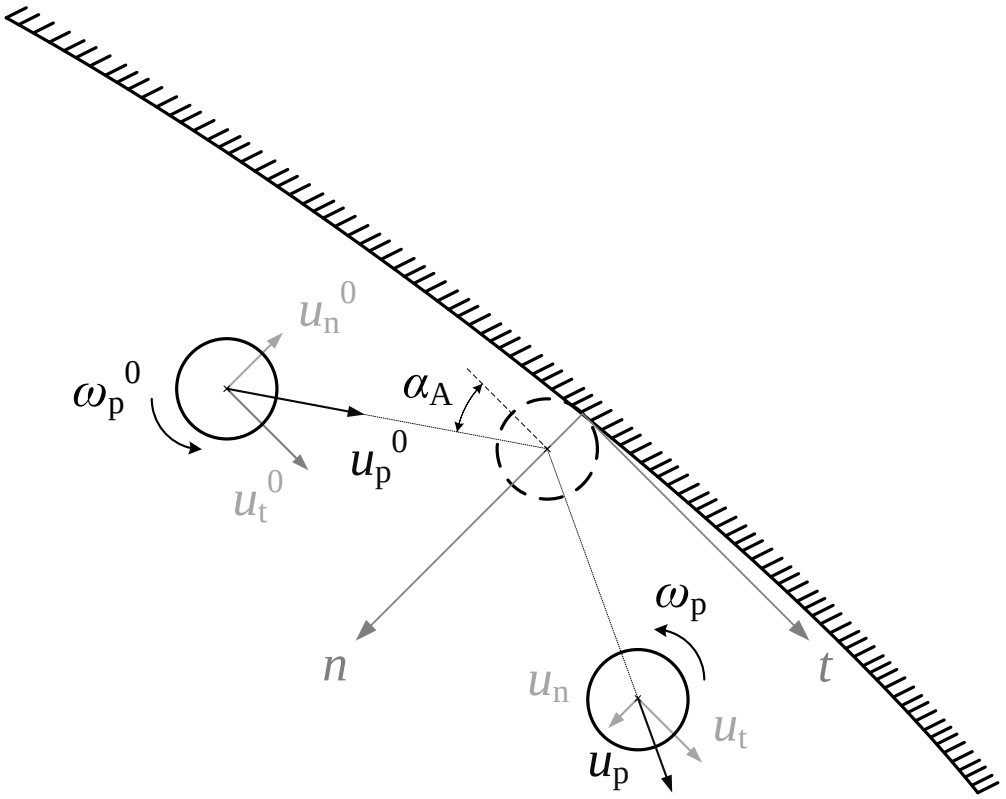
- hydrodynamic forces: the fluid resistance when the particle approaches the wall (the Saffman lift force for instance),
- collision (mechanical) forces, which occur when particles are submitted to collisions.

Collision can be modeled with two methods:

- the soft particle model, which is based on a spring-mass damper system analogy. This method requires small time steps and is computationally time-consuming.
- the hard particle method, which treats binary collisions, and which is usually employed for particle-wall interactions.

Let us consider a particle approaching a wall. To calculate the particle velocity after the impact with the wall, the tangential vector and the normal vector to the wall are needed (the shape of the wall surface does not matter according to OESTERLÉ (2006)). The impact of a sphere with a wall can be viewed as a planar mechanics problem (BRACH and DUNN, 1992), as illustrated in Figure 4.7.

Figure 4.7 – Particle-wall planar collision model.



To find the expression of the post-collisional particle velocity, several equations must be solved:

- the momentum equation between two times, one before the collision, and one after the collision:

$$m_p\left(\vec{u}_p - \vec{u}_p{}^0\right) = \vec{J}_p \,, \tag{4.107}$$

where m_p, $\vec{u}_p$, $\vec{u}_p{}^0$ are, respectively, the particle mass, the particle velocity after impact and the particle velocity before impact, and $\vec{J}_p$ is the impulse of the force acting on the particle:

$$\vec{J}_p = \int_0^\tau \vec{F}_p dt \,, \tag{4.108}$$

where $\vec{F}_p$ is the force acting on the particle, and τ is the duration.

- the angular momentum equation between two times:

$$I_p\left(\vec{\omega}_p - \vec{\omega}_p{}^0\right) = \int_0^\tau \overrightarrow{PM} \times \vec{F}_p dt = \overrightarrow{PM} \times \vec{J}_p \,, \tag{4.109}$$

where I_p, $\vec{\omega}_p$, $\vec{\omega}_p{}^0$ are, respectively, the moment inertia of the particle, the angular velocity after and before impact, and $\left\|\overrightarrow{PM}\right\|$ is the particle radius (the particle is rigid, it is not deformed during the collision).

To solve that problem, additional equations are used:

- Coulomb's friction law which involves the coefficient of friction, f_d, which is the ratio of the tangential to the normal impulse:

$$f_d = \frac{J_t}{J_n}, \tag{4.110}$$

  where J_t and J_n are, respectively, the tangential and normal components of the impulsive force.

- the coefficient of restitution, e_n, which is based on the normal particle velocity component or on the normal component of the impulsive force:

$$e_n = \frac{-u_n}{{u_n}^0} \quad \text{or} \quad e_n = \frac{-J_n}{{J_n}^0} \tag{4.111}$$

  where u_n and ${u_n}^0$ are, respectively, the normal components of the particle velocity after and before collision, and J_n and ${J_n}^0$ are, respectively, the normal components of the impulse force after and before collision (let us notice here that if $e_n = 0$ the particle is stuck at the wall, and if $e_n = 1$ the particle is submitted to a perfect rebound).

- the particle velocity boundary conditions (i.e., the particle velocity at the wall).

With this set of equations, the components of the post-collisional velocities can be obtained. In this study, the following impact models will be employed: Brauer's PWI model (BRAUER, 1980), Grant and Tabakoff's PWI model (GRANT and TABAKOFF, 1975), Brach and Dunn's PWI model (BRACH and DUNN, 1998), Matsumoto and Saito's PWI model (MATSUMOTO and SAITO, 1970).

4.2.3.1 Brauer's PWI model

In this model, constant restitution coefficients deduced from experimental data obtained by BRAUER (1980) are used. According to GORHAM and KHARAZ (2000), these coefficients were obtained from the measurement of the impact of 6 mm steel spheres against a range of pipe wall materials. These data we used by other authors such as NAIK and BRYDEN (1999) to numerically study the impact of micro-particles in the particulate-laden flow in curved ducts. The restitution coefficients according to the tangential and normal directions are:

$$e_t = 0.8, \tag{4.112}$$

$$e_n = 0.9. \tag{4.113}$$

This yields the following equations for the post-collisional normal and tangential components of the particle velocity:

$$u_t = e_t {u_t}^0 , \tag{4.114}$$

$$u_n = -e_n {u_n}^0 . \tag{4.115}$$

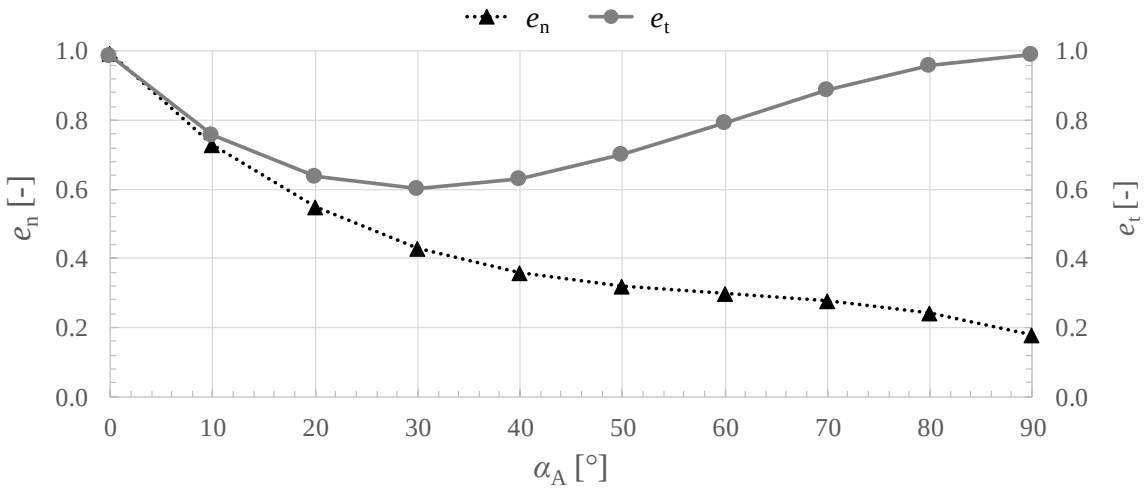
4.2.3.2 Grant & Tabakoff's PWI model

GRANT and TABAKOFF (1975) used experiments to build a model that provides the tangential and normal restitution coefficient, as functions of the impact angle α_A (see Figure 4.7). The experiments were carried out with quartz particles of diameter 200 μm. The particles were injected in a wind tunnel with glass walls. They obtained the following relationships (presented in Figure 4.8) for tangential and normal restitution coefficients:

$$e_t = 0.988 - 1.66\alpha_A + 2.11{\alpha_A}^2 - 0.67{\alpha_A}^3 , \tag{4.116}$$

$$e_n = 0.993 - 1.76\alpha_A + 1.56{\alpha_A}^2 - 0.49{\alpha_A}^3 . \tag{4.117}$$

Figure 4.8 – Grant & Tabakoff's PWI model: coefficient of restitution "normal" and "tangential".



The correlations of the Equations (4.116) and (4.117) were then applied to model tracked particles of diameter varying 20 μm and 200 μm (NJOBUENWU *et al.*, 2013).

4.2.3.3 Brach & Dunn's PWI model

To model micro-particles impact, BRACH and DUNN (1998) developed a rebound model where an additional coefficient is included to the coefficient of restitution, to take into account the loss of energy of the particle due to adhesion forces (such as the van Der Waals force which occurs when there is a potential difference between the particle surface and the wall surface).

With this model, the coefficient of restitution reads:

$$e_n = R_{pw}\left(1 - \rho_{pw}\right), \tag{4.118}$$

where ρ_{pw} is the adhesion coefficient (if $\rho_{pw} = 1$, the particle is stuck at the wall, if $\rho_{pw} = 0$, there is no adhesion of the particle on the wall) and R_{pw} is the coefficient of restitution without adhesion. These coefficients are obtained with correlations deduced from experiments, they are given by:

$$R_{pw} = \frac{{k_1}^a}{{k_1}^a + |{u_n}^0|^a}, \tag{4.119}$$

$$\rho_{pw} = \frac{{k_2}^b}{{k_2}^b + |{u_n}^0 - u_c|^b}, \tag{4.120}$$

where u_c is the critical velocity of the particle. The experimental coefficients used in these equations are listed in Table 4.8.

Table 4.8 – Brach & Dunn's PWI model: experimental coefficients.

Particle	Wall	k_1	k_2	$u_c\ [m/s]$	a	b
Ag-coated glass	stainless steel	272.0	1.74	-0.40	1.0	0.5
Ag-coated glass	copper	38.7	1.61	0.29	1.0	1.0
Ag-coated glass	tedlar	51.7	4.07	-0.44	1.0	1.0
stainless steel	SiO_2	12.5	0.065	0.074	1.0	0.5
polystyrene latex	polished quartz	556.3	0.273	0.967	1.0	1.0
ammonium fluorescein	molybdenum	55.7	0.586	1.47	1.0	1.0
ammonium fluorescein	mica	72.5	1.333	1.10	1.0	1.0

With this model, the post-collision velocity of the particle is obtained as follows:

$$u_n = -e_n {u_n}^0 \,, \tag{4.121}$$

$$u_t = {u_t}^0 - \mu_d (1 + e_n) {u_n}^0 \,, \tag{4.122}$$

$$\omega = \omega^0 + \frac{5\mu_d}{d_p} (1 + e_n) {u_n}^0 \,, \tag{4.123}$$

where ω^0 is the angular velocity before impact, ω is the angular velocity after impact and d_p is the particle diameter. The coefficient μ_d is the ratio between the tangential and the normal impulse. If the particle slides while it is in contact with the wall, μ_d is equal to the coefficient of friction f_d, in another case, if it does not slide, μ_d is given by:

$$\mu_d = \frac{2\eta}{7(1+e_n)} \,, \tag{4.124}$$

with:

$$\eta = \frac{{u_t}^0 - (d_p/2)\omega^0}{{u_n}^0} \,. \tag{4.125}$$

4.2.3.4 Matsumoto & Saito's PWI model

MATSUMOTO and SAITO (1970) obtained the post-collisional normal and tangential velocity components, by solving the impulse equations mentioned previously. They treated the case of collision with sliding along the wall, and the case of collision without sliding along the wall. With this model, the normal coefficient of restitution is:

$$e_n = max(1 - 0.015\alpha_A; 0.73), \tag{4.126}$$

and the tangential coefficient of restitution is:

$$e_t = \begin{cases} 1 - f_d(1 + e_n)|{u_n}^0 / {u_t}^0| & ; \ {u_n}^0 \geq u_c \ (sliding) \\ 5/7 & ; \ {u_n}^0 < u_c \ (non - sliding) \end{cases} . \tag{4.127}$$

This yields the tangential and normal components of the post-collisional velocity:

$$u_t = e_t {u_t}^0 \,, \tag{4.128}$$

$$u_n = -e_n {u_n}^0 \ . \tag{4.129}$$

The critical velocity, u_c, that must be calculated for selecting the sliding or the non-sliding condition is:

$$u_c = 3.5[f_0(1 + e_n)|{u_n}^0|] \ , \tag{4.130}$$

with the dynamic friction coefficient, f_0, given by:

$$f_0 = max(0.4 - 0.00926\alpha_A; 0.15) \ . \tag{4.131}$$

where α_A is the incident angle.

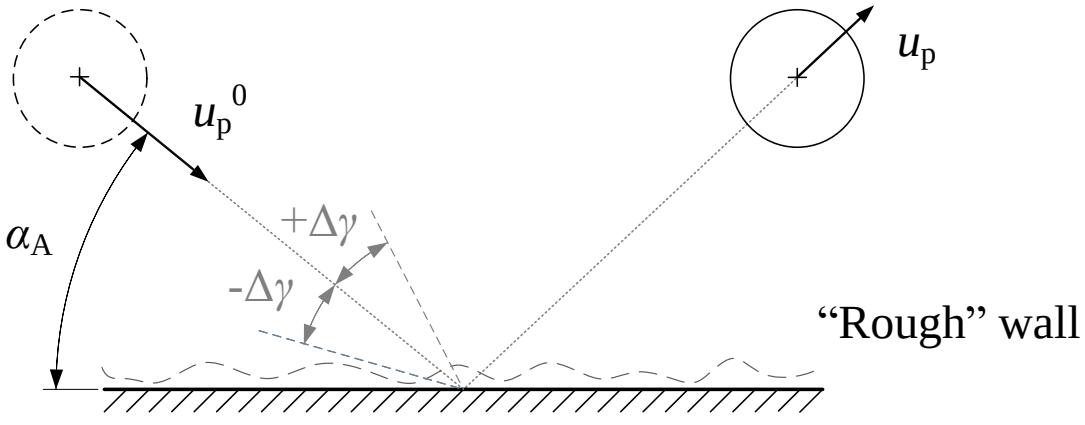
4.2.3.5 Virtual wall model

To consider the influence of the wall roughness effect on the particle collision with the wall, SOMMERFELD and HUBER (1999) introduced in MATSUMOTO and SAITO (1970) model a stochastic model, where the impact angle is modified as illustrated in Figure 4.9. The approach angle ${\alpha_A}'$ is calculated according to:

$${\alpha_A}' = \alpha_A + \Delta\gamma\zeta \ , \tag{4.132}$$

where $\Delta\gamma$ is an angle determined by experiment (e.g., 3.8° for the Plexiglass plate) and ζ is a Gaussian function with zero mean and unit standard deviation.

Figure 4.9 – Virtual wall impact model.



4.2.4 Lagrangian methods

The Lagrangian approach statistically determines the average field of concentration and velocity of a specific number of particle trajectories from the equation of particle dynamics (FAN and ZHU, 2005).

The Lagrangian models consider a smaller number of simplification hypotheses than the Eulerian approaches and they are based on the resolution of the transport particle equation applied for each individual particle in the fluid flow. In the Lagrangian Particle Tracking (LPT) methods, the Equations (4.103)-(4.105) are written and then integrated with respect to time along the particle trajectory. These equations are solved for each individual particle and the Lagrangian approaches can yield the trajectories of all the particles.

The Lagrangian methods are useful to predict particle dispersion and deposition in HVAC elbows because the flow contains a low volume fraction of particles (one-way coupling). Otherwise, this method should not be practical regarding the computational coasts necessary to obtain a converged solution. The flow field must be predetermined for the Lagrangian method, therefore the fluid phase can be simulated with a Eulerian method, i.e., the Eulerian-Lagrangian approach.

When using a RANS model for predicting a turbulent flow, the only time-averaged fluid velocity is calculated. However, to calculate the forces that act on the particle, the instantaneous fluid velocity $\vec{u}$ seen by the particle is needed:

$$\vec{u}(t) = \vec{U} + \overrightarrow{u'}(t) \ . \tag{4.133}$$

where $\vec{U}$ is the time-averaged fluid velocity at the particle location, and $\overrightarrow{u'}$ is the fluctuating velocity. To do that, stochastic models can be used (TU *et al.*, 2013). It has been shown that the model proposed by GOSMAN and IONNIDES (1983) can provide satisfactory results for complex flows (De ANGELIS *et al.*, 1997; LOTH, 2000). This model was used in this study. In this model, a random variable of zero mean and standard deviation proportional to the turbulent kinetic energy is employed. If the turbulence is isotropic, we have:

$$\overline{u'^2} = \overline{v'^2} = \overline{w'^2} = \frac{2}{3}k \ , \tag{4.134}$$

where k is the turbulent kinetic energy and u', v' and w' are the fluctuations of the fluid velocity in the three Cartesian coordinates. The standard deviation is thus $\sqrt{\frac{2}{3}k}$. The local velocity fluctuation of the fluid phase when the particle crosses a turbulent eddy is calculated as:

$$v' = \zeta\sqrt{\frac{2}{3}k}\,, \tag{4.135}$$

where ζ is a Gaussian number, of zero mean and unit variance. The particle equation of motion is integrated until the particle does not interact anymore with the same eddy. To do that, the interacting duration τ_i of the particle with the eddy is calculated. It is the minimum between the eddy lifetime τ_e and the time for the particle to pass through the eddy τ_r:

$$\tau_i = min[\tau_e, \tau_r]\,. \tag{4.136}$$

The eddy lifetime can be estimated as:

$$\tau_e = \frac{k}{\varepsilon}\,. \tag{4.137}$$

The time for the particle to pass through the eddy can be determined as:

$$\tau_r = -\tau_p ln\left(1 - \frac{l_e}{\tau_p|\vec{u}_{p,r}|}\right), \tag{4.138}$$

where τ_p is the particle relaxation time.

4.3 NUMERICAL PROCEDURE

The governing equations are solved using the computational code OpenFOAM® (MARIC *et al.*, 2014). Specifically, two standards OpenFOAM® solvers (version 6.0.0) were employed: *pisoFoam* (to calculate the turbulent flow in the duct bend) and *icoUncoupledKinematicParcelFoam* (to calculate the particle dispersion). The main numerical procedure used by these solvers are explained below and the guidelines of the numerical simulation of this study are found in APPENDIX A, APPENDIX B, and APPENDIX C.

4.3.1 Fluid flow

The governing equations are non-linear PDE and were presented in the Cartesian coordinate system and index notation. These equations are solved with the Finite Volume Method (PATANKAR, 1980) which consists in integrating these equations written in a conservative form over a control volume and solving the resulting algebraic equation system with an appropriate method (e.g., the multigrid method) until the numerical convergence for the target error, a mean property variation range or a number of iterations is obtained. In OpenFOAM®, a collocated variable arrangement is used (FERZIGER and PERIC, 2002).

Details about these numerical methods will not be presented here for sake of brevity and for this purpose the following literature is recommended: HOLZMANN (2018), VERSTEEG and MALALASEKERA (2007) and FERZIGER and PERIC (2002).

The continuity equation of the incompressible flow does not contain the pressure term. To overcome that difficulty, several algorithms have been developed by PATANKAR (1980) and other researchers, e.g.: Semi Implicit Linked Equations (SIMPLE) (PATANKAR and SPALDING, 1972), SIMPLE-Revised (SIMPLER) (PATANKAR, 1980), SIMPLE-Consistent (SIMPLEC) (van DOORMAAL and RAITHBY, 1984), and Pressure Implicit with Splitting of Operators (PISO) (ISSA, 1986). *PisoFoam* is the PISO algorithm in OpenFOAM®. It enables to solve the unsteady Navier-Stokes equations, and it can also be used to calculate steady flows.

The necessary conditions for the numerical convergence of an incompressible turbulent internal flow are:

- the Courant number ($Co = \left|\vec{U}\right|\Delta t/\Delta x$) must be less than 1;
- the appropriate choice of the numeric schemes;
- the appropriate choice of the tolerances;
- the mesh must have an acceptable quality:
    - the y^+ should respect the range of the turbulence model;
    - the non-orthogonality should be as low as possible;
    - the growth rate from the smaller elements from the wall should be as small as possible.

Due to limits of computer power and time, the CFD simulation results depend on the mesh for 3D turbulent airflow through HVAC elbows. Therefore, it is necessary to realize a

grid dependency study[19] in order to evaluate the variation of certain properties of the flow and to determinate how they vary according to the mesh refinement. There are some ways to do this analysis, among them the following three ways were applied in this study:

- the analysis of a profile: to compare each grid with a significant flow property in a representative domain region, such as the analysis of the stream-wise velocity of air in the duct bend outlet[20];
- the evaluation of property: to plot a property with the inverse value of grid size (the finest meshes approach to zero) and to observe a tendency to a coherent value, such as the pressure loss on a duct system that can be determined with well-known correlations;
- the Grid Convergence Index (GCI) parameter: proposed by ROACHE (1994) to report grid refinement studies in CFD, such as the error of the maximum stream-wise velocity at duct bend outlet.

The method of ROACHE (1994) is based on a grid refinement error estimator derived from the theory of generalized Richardson extrapolation. The expression to calculate the GCI is defined as:

$$GCI = \frac{3|E_r|}{{r_m}^{p_m-1}}\,, \tag{4.139}$$

with:

$$E_r = \frac{\varphi_{coarse}}{\varphi_{fine}} - 1\,, \tag{4.140}$$

$$r_m = \left(\frac{N_{fine}}{N_{coarse}}\right)^{1/3}\,, \tag{4.141}$$

where E_r is the relative error between a property of the coarse grid (φ_{coarse}) and the fine grid (φ_{fine}), r_m is the grid refinement and it is the ratio between the fine grid size (N_{fine}) and the coarse grid size (N_{coarse}), and p_m is the numerical order of accuracy. In this work only second-order schemes are employed, therefore $p_m = 2$ and the GCI was performed by fixing the finest mesh and by variation of the coarse mesh results.

[19] Also called sensitive study or convergence study.
[20] Region where the velocity profile is more distorted.

4.3.2 Particle transport

The governing equations are ODE and were presented in symbolic notation. These equations are integrated according to time with the Euler scheme and the particle tracking follows the face-to-face method (MACPHERSON *et al.*, 2009) in which the particles are tracked to the cell faces, updating the cell information and tracking the particle further on. To calculate the forces that act on the particles, the fluid velocity is needed at the particle location. This is done with linear interpolation. The *icoUncoupledKinematicParcelFoam* is a transient solver for the passive transport (one-way coupling) of a single kinematic particle cloud (HOLZINGER, 2019). In order to simplify the modelling of particle dynamics, some hypotheses are assumed:

i. the particles have a spherical shape;
ii. the point particle approximation is used;
iii. the particle density is much higher than the fluid density (i.e., $\rho_p/\rho \gg 1$);
iv. the particle-particle interactions are neglected;
v. the particles are monodispersed;
vi. the fluid-solid flow is dilute.

The aerosol particles found in HVAC ducts have not exactly a spherical shape. For simplification, particles considered in this study are spheres and the models used to calculate the forces that act on the particles are based on this assumption.

The air density at SATP (Standard Ambient Temperature and Pressure) is almost 1.2 kg/m^3 and the particle density in HVAC ducts is of the order of 10^3 kg/m^3, therefore the particle and fluid density ratio must be, at least, of the order of 10^2.

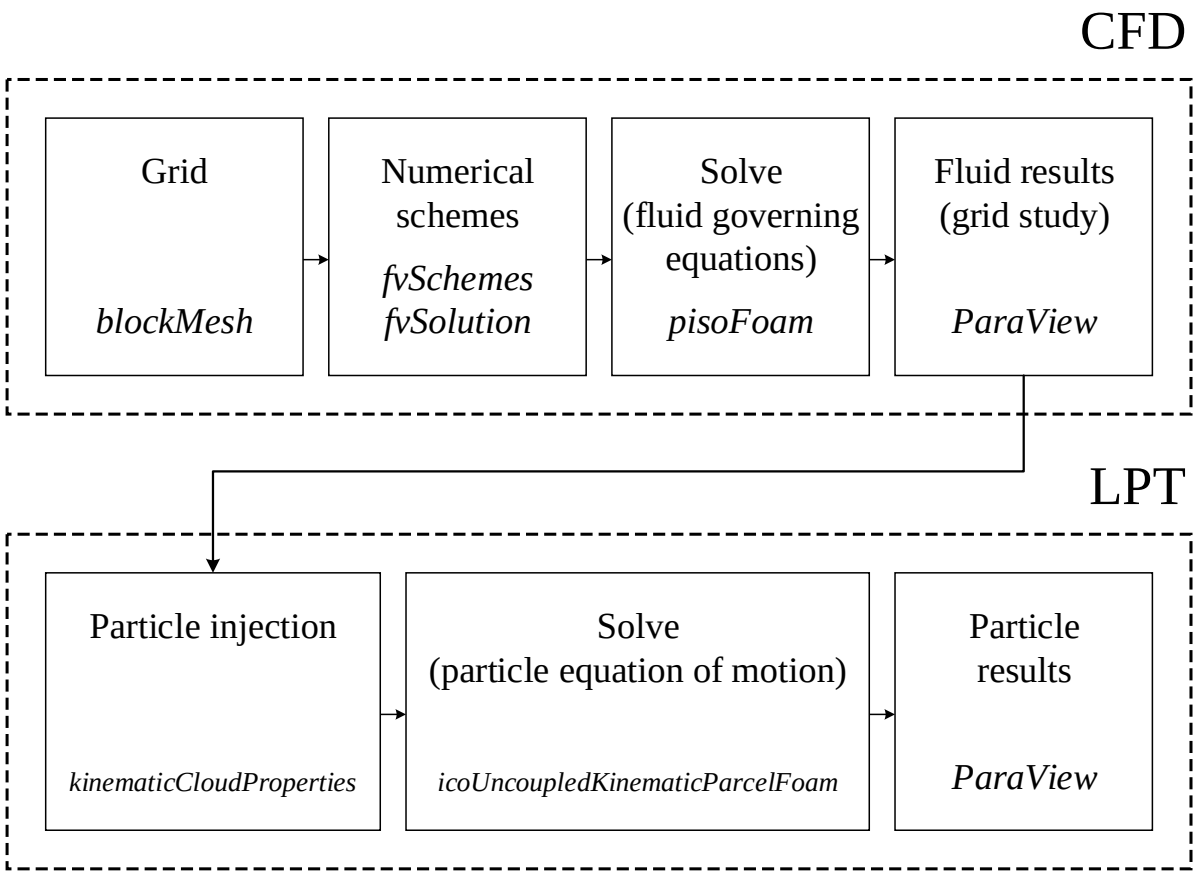
The particle-particle collisions are neglected, and the solid particles have the same shape (spherical), size (diameter) and properties (density), i.e., the particles are monodispersed.

The particle volume fraction in HVAC applications is lower than 10^{-6}, therefore the flow regime is characteristic of diluted flows. Only the flow affects the particle motion, which should not affect significantly the flow field because they are very small, i.e., the one-way coupling can be considered.

4.4 FINAL REMARKS OF PHYSICAL AND MATHEMATICAL MODELLING

In this present chapter, the physical models, and the numerical models, that were chosen to compute particle dispersion in curved ducts have been presented. Since the aerosols in HVAC applications are dilute, the one-way coupling approach was chosen. The Lagrangian approach was selected because dilute aerosols are modeled, and the particle-wall interactions are naturally applied with this method. In the following chapter, several turbulence models will be tested to model flows in curved ducts, several particle-wall interaction models will also be evaluated, and the influence of the shape of the curved duct on particle dispersion will be studied. The OpenFOAM® toolbox was selected because of its recognized capability to handle numerical tools for the solution of continuum mechanics (LAI *et al.*, 2010) in the CFPD (Computational Fluid and Particle Dynamics) field (TU *et al.*, 2013). Figure 4.10 presents a diagram of the numerical procedure applied in this study.

Figure 4.10 – Diagram of the Eulerian-Lagrangian method.



5 RESULTS

The main objective of this chapter is to present the verification and validation (V&V) of cases selected in this study to perform the previously proposed analysis. In order to develop a systematic approach, the geometry with boundary and initial conditions for each case are first summarized. Afterwards, a grid convergence study is made mostly evaluating the pressure drop, the friction factor, and the velocity profiles. Finally, the validation study is carried out to verify the accuracy of the numerical modelling of this research and to perform discussions about the respective principal findings.

Previously, this work attempted to explain that the turbulent flow field patterns influence the particle transport and deposition, and the associated physics. Therefore, the first case study will focus on the selection of an appropriate turbulence model to describe the flow within the curved duct. The second and third case studies are validation cases of particle transport and deposition in elbows.

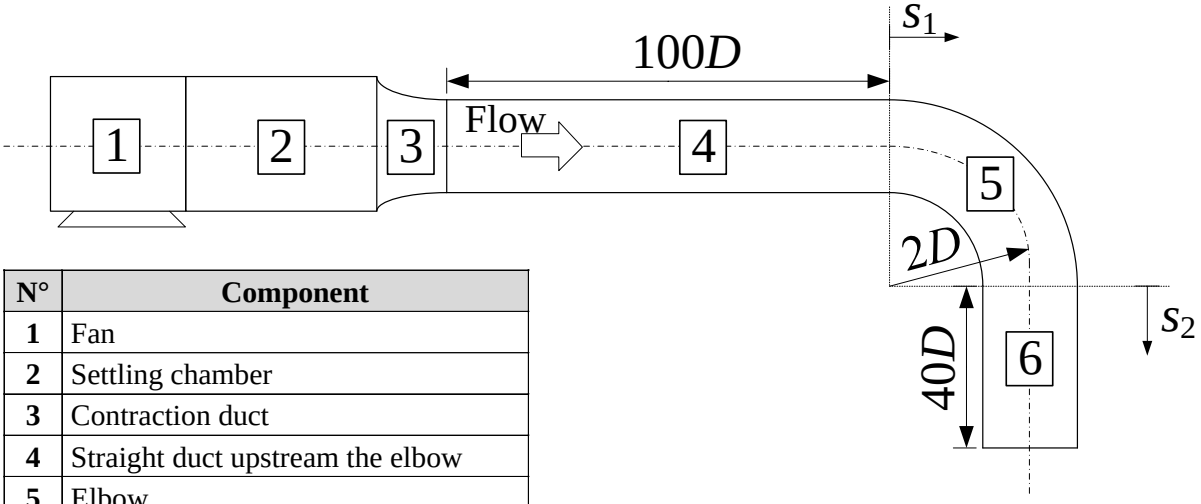
Essentially, three proposed problems were studied:

- The first case study deals with turbulent airflow (without aerosol) in a horizontal to horizontal HVAC circular duct bend (SUDO *et al.*, 1998). This case enables to select a proper turbulence model for flows in curved ducts. The available literature concerning particle deposition in HVAC duct bends usually applied a turbulence model without evaluating the adequacy of the turbulence model for predicting the flow in curved geometries. A selection of an adequate turbulence model to predict the flow in HVAC duct bends is nevertheless essential for the sequence of this study.
- The second case study concerns the numerical modelling of the experimental work of KLIAFAS (1984) that concerned a turbulent airflow in a vertical to horizontal circular curved pipe with micro glass spheres. From the first case study, four RANS turbulence models were chosen and compared against the current literature. Moreover, the influence of four PWI models (associated with each RANS model) was tested regarding the mean velocity. In addition, this study was performed with 2D and 3D models.
- The third study aims at evaluating the particle deposition rates in horizontal to horizontal HVAC elbows with square (SUDO *et al.*, 2001) and circular cross-sections. Results for the collection efficiency and deposition velocity are shown.

5.1 TURBULENT AIRFLOW IN A DUCT BEND

SUDO *et al.* (1998) carried out experiments, as shown in Figure 5.1, to measure mainly the mean and fluctuating velocities in the turbulent airflow through a horizontal to horizontal circular duct with a diameter of 0.104 m, an elbow with a curvature ratio of 4 and a Reynolds number of the flow equal to 6×10^4. This work was chosen as a validation case to study the airflow behavior (without aerosol) in a curved duct because the dimensions and flow conditions were similar to those of HVAC applications.

Figure 5.1 – Schematic diagram of Sudo *et al.* (1998) experimental system.



N°	Component
1	Fan
2	Settling chamber
3	Contraction duct
4	Straight duct upstream the elbow
5	Elbow
6	Straight duct downstream the elbow

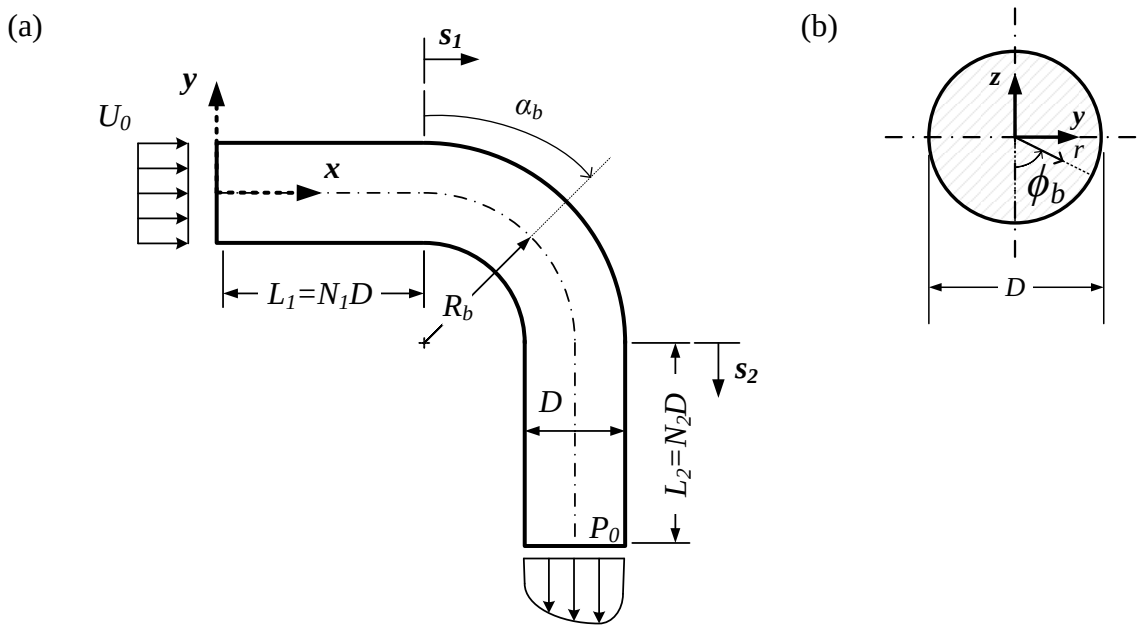
Source: adapted from SUDO *et al.* (1998).

5.1.1 Geometry and boundary conditions

The model of the duct bend is presented in Figure 5.2. In this figure: D is the duct diameter, R_b is the bend radius, α_b and ϕ_b are, respectively, the angular coordinates of the duct bend and of the duct cross-section, L_1 and L_2 are, respectively, the lengths of the straight ducts upstream and downstream the duct bend.

Performed numerical studies showed that it is necessary to include straight ducts upstream and downstream the duct bend for the computational simulation, to prevent reverse flows at the outlet and not influence the airflow through the elbow. In addition, the curvature influences the flow in the bend inlet and outlet, making difficult the task of implementing accurate boundary conditions at these locations.

Figure 5.2 – Geometry and boundary conditions of the duct bend model: (a) top view and (b) cross-section.



Concerning the boundary conditions, the inlet velocity (U_0) is uniform and the atmospheric pressure (P_0) is applied at the outlet[21]. The walls have no-slip boundary conditions for the velocity and zero gradient for the pressure. The entrance length of this case is about $25D$[22].

The boundary conditions regarding the turbulent variables must be specified at the boundaries for turbulent flows[23]. The boundary conditions at the wall for the turbulent variables were specified according to the selected turbulence model.

The turbulent kinetic energy at inlet (k_0) can be estimated as (VERSTEEG and MALALASEKERA, 2007):

$$k_0 = \frac{3}{2}(T_i U_0)^2 \,, \tag{5.1}$$

where the turbulence intensity T_i can be calculated as (BASSE, 2017):

$$T_i = 0.317 Re^{-0.11} \,. \tag{5.2}$$

[21] The outlet velocity and the inlet pressure were given by a zero gradient.

[22] According to ANSELMET *et al.* (2009) the entrance length (L_e) is defined as: $\frac{L_e}{D_h} = 1.6 Re^{1/4}$.

[23] At the outlet, a zero gradient was applied for the turbulent variables and for the wall appropriate wall functions were selected for each case.

The dissipation rate of the turbulent kinetic energy at inlet (ε_0) can be defined as:

$$\varepsilon_0 = {C_\mu}^{3/4} \frac{{k_0}^{3/2}}{l_t}\,, \tag{5.3}$$

where C_μ is a constant ($C_\mu = 0.09$) and l_t is the mixing length ($l_t = 0.07 D_h$).

The turbulent frequency at inlet (ω_0) can be approximated as follow:

$$\omega_0 = \frac{\varepsilon_0}{k_0}\,. \tag{5.4}$$

The turbulent velocity scale at inlet ($\overline{v'^2}_0$) and the relaxation function at inlet (f_0) can be determined, respectively, as (SVENINGSSON, 2003):

$$\overline{v'^2}_0 = \frac{2}{3} k_0\,, \tag{5.5}$$

$$\frac{\partial f_0}{\partial n} = 0\,, \tag{5.6}$$

where n is the normal unit vector at the inlet surface.

The components of the Reynolds stress tensor at the inlet, $\left(\tau_{ij}\right)_0$, can be calculated according to NJOBUENWU *et al.* (2013)[24]:

$$\left(\overline{u'u'}\right)_0 = \frac{1}{2} k_0\,, \tag{5.7}$$

$$\left(\overline{v'v'}\right)_0 = \frac{2}{5} k_0\,, \tag{5.8}$$

$$\left(\overline{w'w'}\right)_0 = \frac{2}{3} k_0\,, \tag{5.9}$$

$$\left(\overline{u'v'}\right)_0 = \left(\overline{u'w'}\right)_0 = \left(\overline{v'w'}\right)_0 = 0\,. \tag{5.10}$$

The principal objective of this section is to apply the RANS turbulence models described in Section 4.1.1 and evaluate the accuracy of each model to predict the turbulent airflow in a typical HVAC duct bend. APPENDIX B presents a typical run in OpenFOAM® with previous boundary and initial conditions. Table 5.1 outlines the geometry, the thermophysical properties

[24] Usually the RSM models have convergence issues, therefore a two-equation liner EVM can be applied as initial condition for the solution with an approximation of the Reynolds stress tensor.

of the fluid and the flow conditions of this study. The lengths of the straight ducts upstream and downstream the bend is 5.20 m and 2.08 m, respectively. The curvature ratio of the bend, which is equal to 4, leads to a Dean number of 3×10^4 and a bend radius of 0.208m.

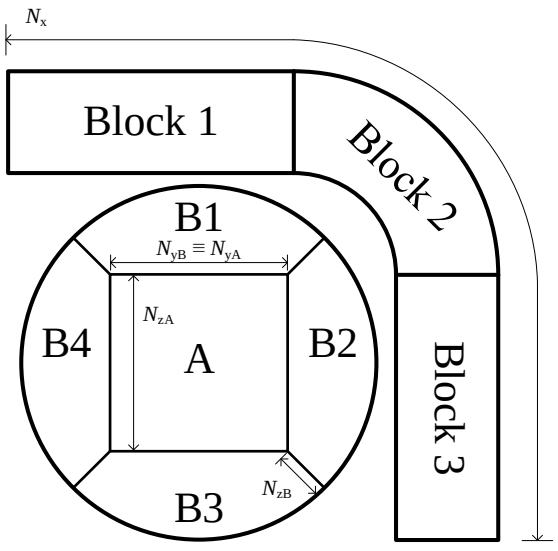
Table 5.1 – Elbow model: geometry, fluid properties, and flow characteristics.

Geometry	Air	Airflow
D = 104 mm	ρ = 1.2 kg/m^3	U_0 = 8.7 m/s
N_1 = 50	μ = 1.81×10^{-5} kg/ms	u_τ = 0.43 m/s
N_2 = 20	ν = 1.51×10^{-5} m^2/s	f = 0.0201
R_b = 208 mm		Δp = 73 Pa
δ = 4		Re = 6×10^4
α_b = 90°		De = 3×10^4

5.1.2 Mesh

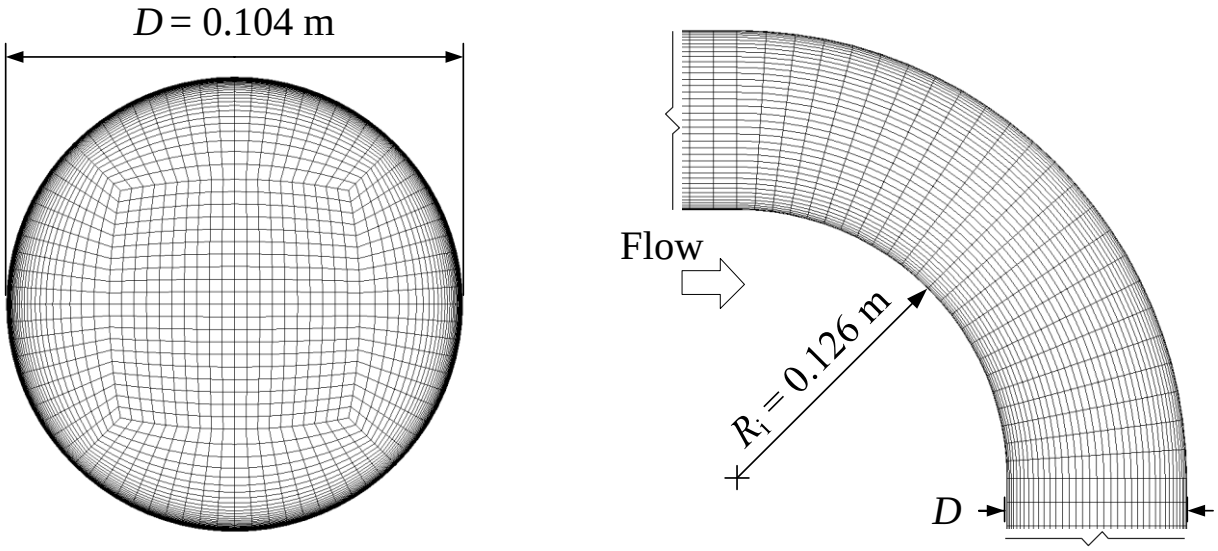
The mesh was generated with the *blockMesh* tool of the OpenFOAM® code. APPENDIX A gives the *blockMeshDict* file which is necessary to create the mesh. The geometry was divided into three parts as presented in Figure 5.3: Block 1 (the upstream duct), Block 2 (the elbow) and Block 3 (the downstream duct). The cross-section of the circular duct is composed of five additional parts to follow the on-grid method for the mesh generation (ASHRAE, 2017). Therefore, we have a total of 15 blocks to generate this mesh with N_x divisions for the stream-wise direction and N_{yA}, N_{yB}, N_{zA} and N_{zB} divisions for the wall-normal directions.

Figure 5.3 – Block meshing strategy.



The mesh generated is shown in Figure 5.4 (the upstream and downstream duct meshes are not presented). The nodes of each block B increase in the wall-normal direction from the inner wall with a cell-to-cell expansion ratio of 1.2 until the block A (and then, the cell dimensions are uniform, i.e., $N_{yA} = N_{zA}$).

Figure 5.4 – Mesh: Grid 5.



5.1.3 Verification and Validation (V&V)

The verification involves:

- the performance of a grid convergence study (Table 5.2);
- the consistency of the pressure drop for each grid (Figure 5.5);
- the analysis of the stream-wise mean velocity profiles for each grid (Figure 5.6).

The grid convergence study of the turbulent airflow was conducted with the SST k-ω turbulence model[25], and the selected grids are listed in Table 5.2, where N is the grid size (number of nodes). The refinement procedure follows the GCI method of ROACHE (1994) with a refinement ratio (r_m) near 1.2.

[25] We assumed that the other turbulence models follow a similar grid convergence study.

Table 5.2 – Grid convergence study for the SST *k-ω* turbulence model.

Mesh	$N_x \times [N_{yA} \times N_{zA} + 4(N_{yB} \times N_{zB})]$	N ($\times 10^6$)	y^+		Δp [Pa]	f
			ave.	max.		
Grid 1	66×[10×10+4(10×26)]	0.075	0.039	0.090	74.24	0.0150
Grid 2	66×[15×15+4(15×32)]	0.141	0.039	0.089	75.50	0.0151
Grid 3	129×[15×15+4(15×32)]	0.277	0.039	0.093	75.56	0.0152
Grid 4	129×[20×20+4(20×40)]	0.464	0.040	0.096	74.77	0.0153
Grid 5	221×[20×20+4(20×40)]	0.796	0.040	0.093	74.96	0.0153
Grid 6	221×[30×30+4(30×50)]	1.525	0.040	0.099	75.06	0.0153

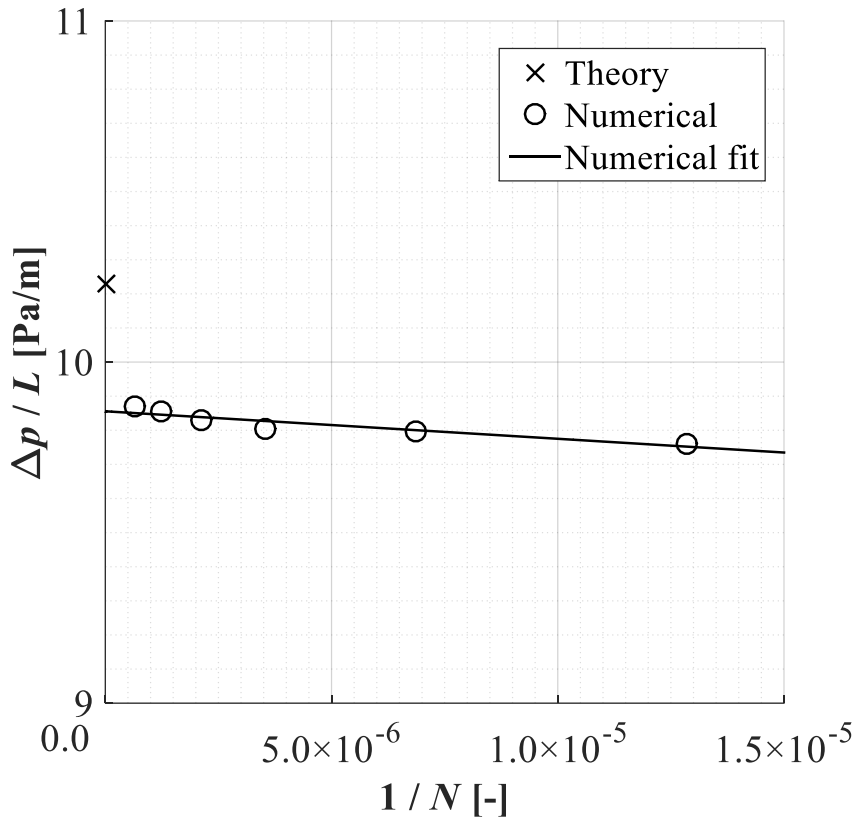
The maximum and average values of y^+ along the walls in the whole domain were calculated from the converged numerical solutions for each grid, and the values were almost the same for all grids and adequate for the low Reynolds number formulation ($y^+ < 1$). The average pressure drop (Δp) and the Darcy-Weisbach friction factor (f) were also calculated. These flow properties do not significantly change among the grids and the converged values were Δp = 75.02 Pa (E_r = +2.5%) and f = 0.0152 (E_r = -24.4%). The relative error (E_r) is calculated as:

$$E_r = \frac{\Phi_{numerical} - \Phi_{theory}}{\Phi_{theory}}, \tag{5.11}$$

where $\Phi_{numerical}$ is the flow property of the converged numerical solution and Φ_{theory} is the flow property calculated according to the theory (FOX *et al.*, 2011).

Another way to verify the grid convergence is the extrapolation point analysis as presented in Figure 5.5. The theory gives a pressure drop per unit length ($\Delta p/L$) of 10.2 Pa/m. The inverse value of grid size ($1/N$) was chosen because when this value is close to 0 the grid is more refined that leads, theoretically, to the best possible numerical solution. The numerical fit extrapolated to 0 gives a pressure drop per unit length of 9.9 Pa/m with a relative error of -3%. Therefore, the extrapolation point method applied to the pressure loss presented a good convergence of the grids in which the refinement leads to the accurate value.

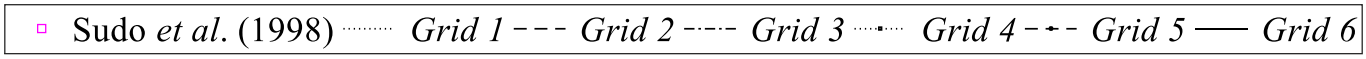
Figure 5.5 – Grid sensitive study with the SST *k-ω* turbulence model for the extrapolation point of the pressure loss per unit length.

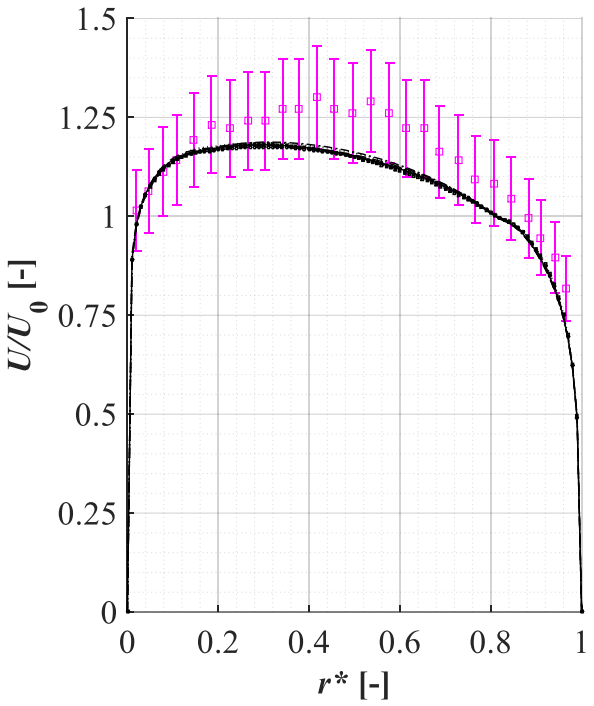


The mean velocity profiles through the elbow can also be a good indicator to observe the convergence of the numerical solution. The stream-wise mean velocity profiles for each grid at the bend angles of 0° (elbow inlet), 30°, 60° and 90° (elbow outlet), normalized with the inlet velocity (U_0 = 8.7 m/s), are presented in Figure 5.6. The dimensionless radius, r^*, is determined according to Equation (2.3), ranging from the inner bend wall (r^* = 0) to the outer bend wall (r^* = 1).

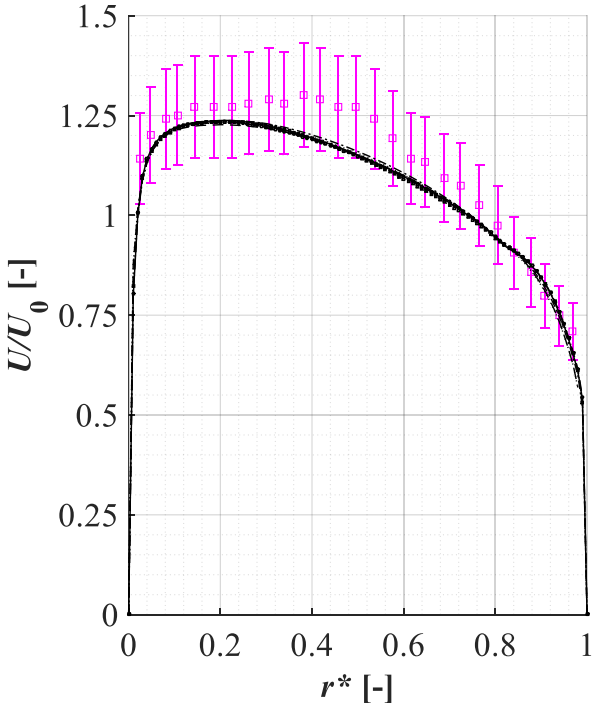
The velocity profiles were almost the same at bend angles of 0° (Figure 5.6a) and 30° (Figure 5.6b), and they agreed with the experimental ones. However, the velocity profiles at bend angles of 60° (Figure 5.6c) and 90° (Figure 5.6d) did not fit well the experimental data near the inner bend wall. Nevertheless, the grid study converged regarding the velocity profile (the velocity profile tends to the finest mesh velocity profile) and *Grid 5* of Table 5.2 was selected for further analysis involving other turbulence models.

Figure 5.6 – Grid independence study with the SST *k-ω* turbulence model for the dimensionless streamwise mean velocity profiles at duct bend angles of (a) 0°, (b) 30°, (c) 60° and (d) 90°.

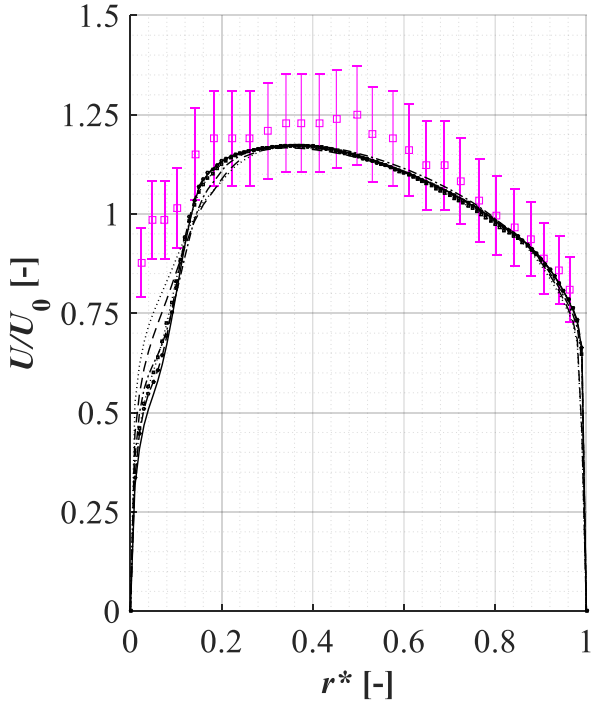




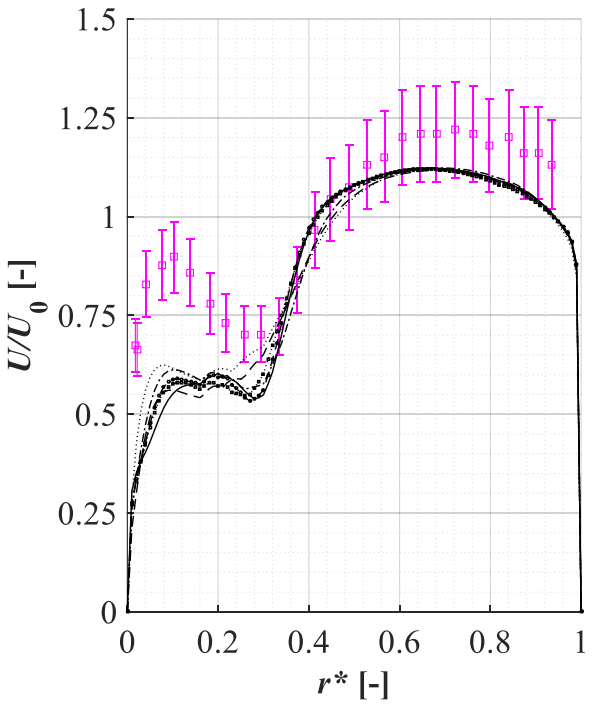
(a) $\alpha_b = 0°$.



(b) $\alpha_b = 30°$.



(c) $\alpha_b = 60°$.



(d) $\alpha_b = 90°$.

The validation of the numerical predictions of this study for each turbulence model against the experimental results of SUDO *et al.* (1998) was divided into the following parts:

- the analysis of pressure drop, friction factor and computing effort (Table 5.3)
- the fully developed turbulent flow profile (Figure 5.7);
- the stream-wise mean velocity (Figure 5.8-Figure 5.11);
- the turbulence intensities (Figure 5.12 and Figure 5.13).

With *Grid 5* of Table 5.2 fixed, other turbulence models were tested and some properties of the converged numerical solution are presented in Table 5.3.

Table 5.3 – Turbulence models tested.

Turbulence model	Δp [a] [Pa] (E_r [b])	f [c] (E_r [b])	N° of iterations [d]	t [e] [h]
k-ε of Lien and Leschziner	76.76 (+4.9%)	0.0159 (-20.9%)	5×10^3	23
k-ε of Launder and Sharma	78.68 (+7.6%)	0.0162 (-19.4%)	1×10^4	21
SST *k-ω*	74.96 (+2.5%)	0.0153 (-23.9%)	8×10^3	10
k-ω	76.26 (+4.2%)	0.0158 (-21.4%)	5×10^3	15
v²f	82.64 (+13.0%)	0.0170 (-15.4%)	4×10^3	19
Lien cubic *k-ε*	76.06 (+4.0%)	0.0159 (-20.9%)	7×10^3	28
RSM LRR	66.71 (-8.8%)	0.0375 (+86.6%)	1×10^4	307
RSM SSG	66.55 (-9.0%)	0.0309 (+53.7%)	1×10^4	282

[a] The calculated pressure drop with Equation (2.10) is 73.16 Pa.
[b] Relative error calculated with Equation (5.11).
[c] The calculated Darcy friction factor with Equation (2.11) is 0.0201.
[d] Number of interactions to achieve convergence.
[e] Total computing time with parallel computation (2 Deca Core Intel Xeon CPU E5-2660 v3s).

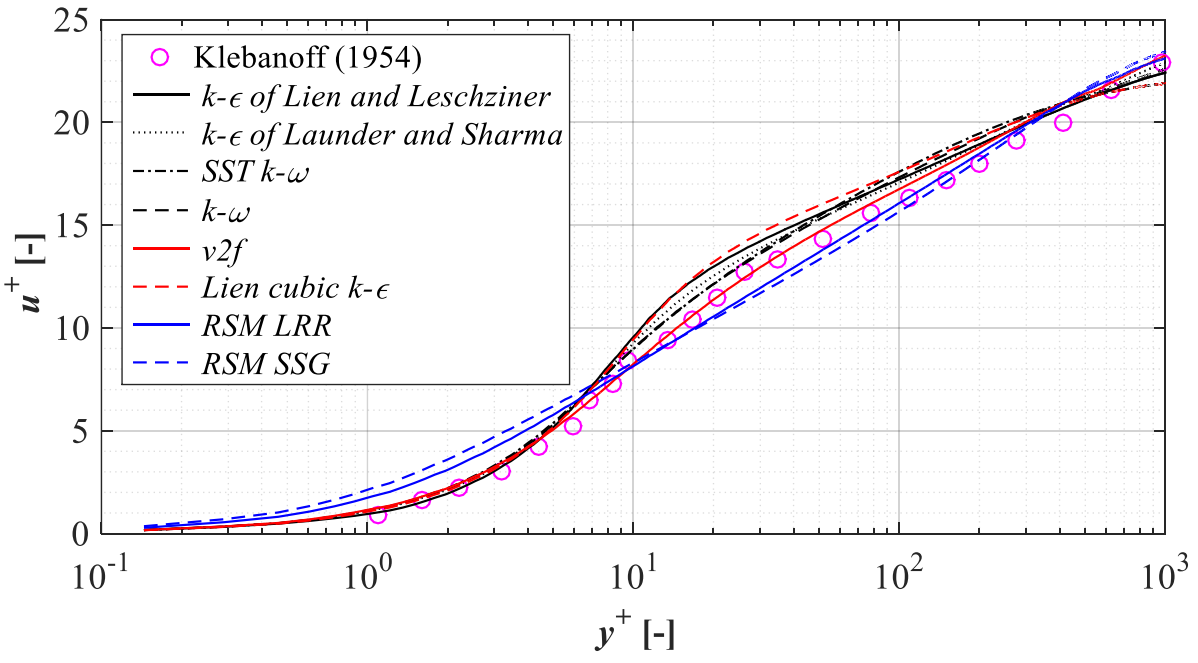
The EVM over-predicts the pressure drop, while the RSM under-predicts it. Nevertheless, all turbulence models were accurate regarding the pressure drop because the absolute maximum relative error was 13% (v^2f model). On the other hand, the EVM under-

predicts the Darcy friction factor, while the RSM over-predicts it. The relative error was ranging from 15% (v^2f model) to 87% (RSM LRR model), therefore all turbulence models were not accurate regarding the Darcy friction factor (despite the k-ε model of Launder and Sharma and the v^2f model yielded a relative error below 20%).

The computational effort was verified according to the number of iterations to achieve the convergence and the total time with parallel computation. The maximum number of iterations was of the order of 10^4 for most calculations. The average total time to reach a converged numerical solution with the EVM was 20h, while the average total time with the RSM was 15 times higher because this model solves additional turbulent variables and is highly non-linear.

Figure 5.7 compares the dimensionless velocity u^+ at s_1/D = -1 from the bottom wall to the duct centerline, represented with the non-dimensional normal distance y^+, with the velocity profile of a fully developed turbulent flow. The predictions of all turbulence models were fairly close to the experimental data of KLEBANOFF (1954) in the viscous sublayer, the overlap layer, and the turbulent layer. However, in the buffer layer, the transition layer between the viscous sublayer and the overlap layer, we observe a difference among the turbulence models and the experimental data. The v^2f model fitted the KLEBANOFF (1954) data with the highest accuracy (*RMSE*~43%), while the Lien cubic k-ε model and the RSM models were the less accurate turbulence models (*RMSE*~100%).

Figure 5.7 – Dimensionless mean velocity profile along the duct radius at s_1/D = -1.



The gap between the numerical and experimental results was evaluated quantitatively by using the Root Mean Square Error ($RMSE$) defined according to the following equation:

$$RMSE = \left[\frac{1}{n}\sum_{i=1}^{n}\left(\Phi_{mes}^{j} - \Phi_{cal}^{j}\right)^{2}\right]^{1/2}, \tag{5.12}$$

where Φ_{cal}^{j} is the property numerically calculated at position j, Φ_{mes}^{j} is the corresponding measured value and n is the number of measured points.

Figure 5.9 presents the normalized mean stream-wise velocity profiles at bend angles of 0°, 30°, 60°, and 90°. All turbulence models accurately predicted the turbulent velocity profile in the bend inlet and at a bend angle of 30° (with an *RMSE* below 11% in this region). At a bend angle of 60°, the RSM LRR predicts the flow separation at the inner bend, but the velocity is too small in comparison with the velocities predicted by the other turbulence models. At a bend angle of 90°, the v^2f, the RSM LRR, and the SST *k-ω* models show similar features. Even though the SST *k-ω* model is linear, it predicts the velocity profile in a similar way as the RSM LRR and *v2f* models. Moreover, the Lien cubic *k-ε* model, which is non-linear, fails to predict the flow in the bend. As indicated in Figure 5.8, the *RMSE* relative to the mean stream-wise velocity for all bend angles and all turbulence models were below 18% (RSM LRR model), while the *RMSE* of most turbulence models tested was below 15% such as the SST *k-ω* and the v^2f turbulence models.

Figure 5.8 – RMSE for the mean stream-wise velocity for each turbulence model.

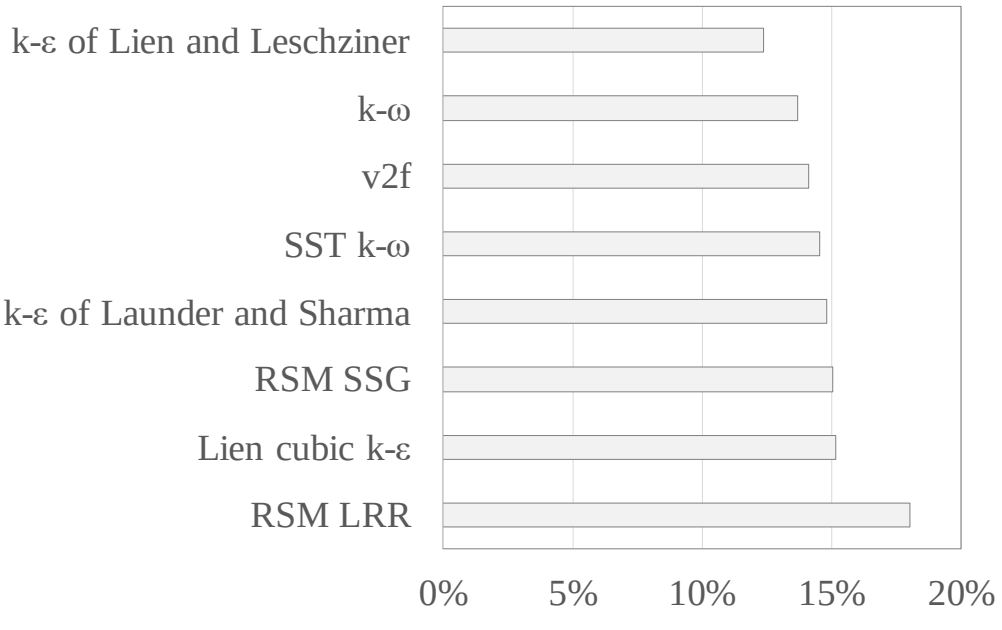
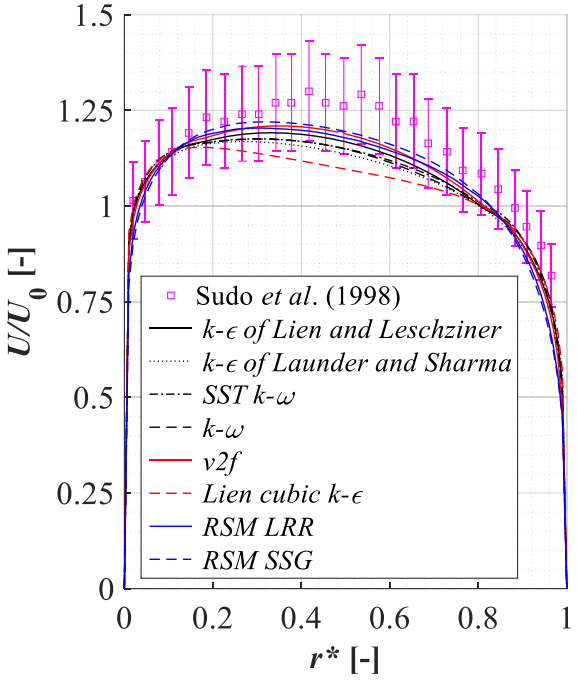
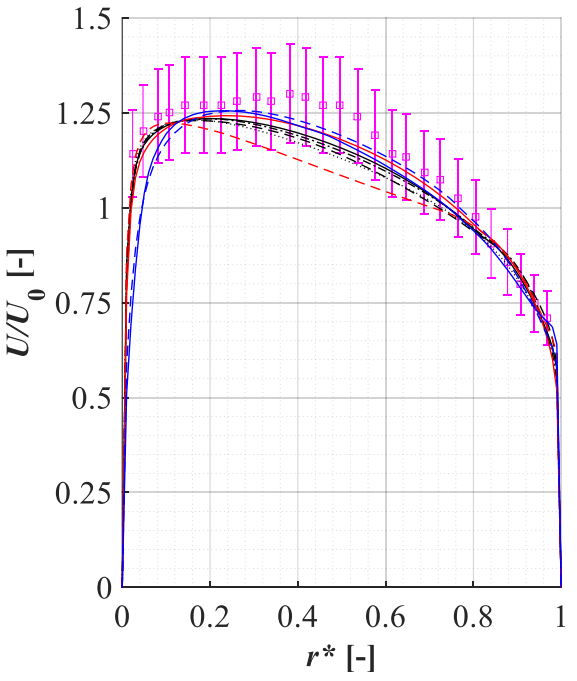


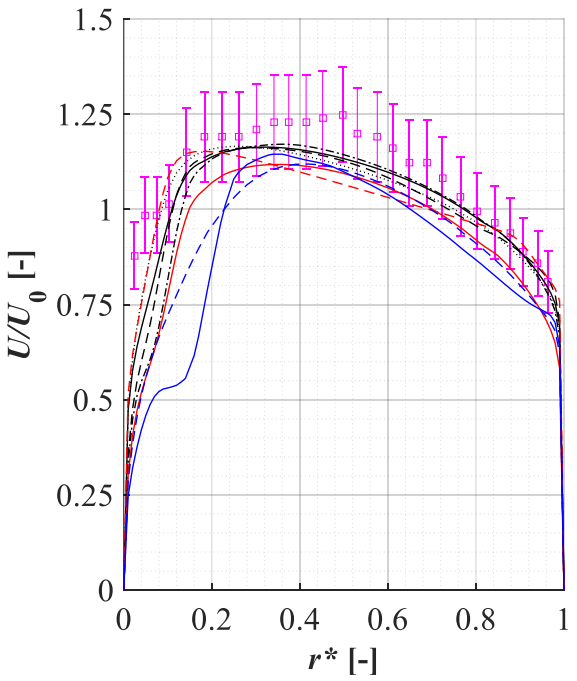
Figure 5.9 – Mean stream-wise velocity profiles at duct bend angles of (a) 0°, (b) 30°, (c) 60° and (d) 90°.



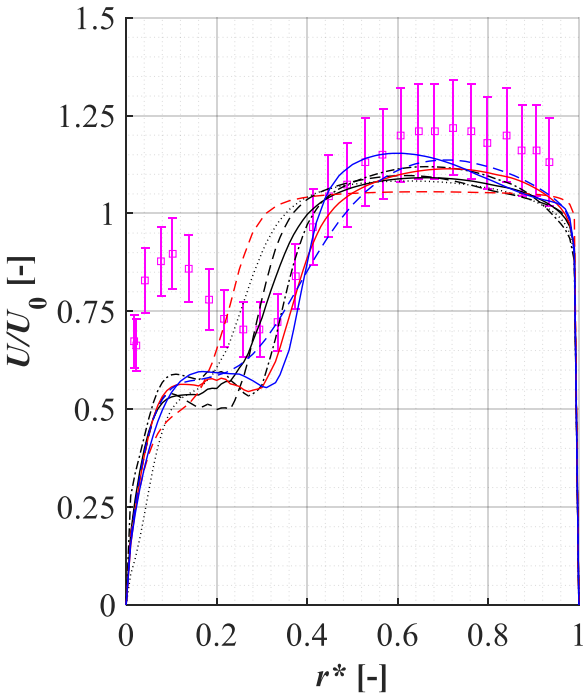
(a) $\alpha_b = 0°$.



(b) $\alpha_b = 30°$.



(c) $\alpha_b = 60°$.



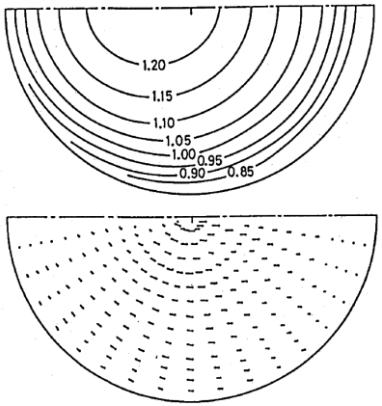
(d) $\alpha_b = 90°$.

The maximum velocity in the fully developed flow region in the upstream straight duct is located at the duct center. The flow profile changes near the bend inlet (Figure 5.10), a secondary flow appears and shifts the maximum velocity towards the inner bend wall (see the isocurve of $U/U_0 = 1.2$). The centrifugal forces appear due to the change in the direction of the flow following the wall curved geometry and set up a secondary motion (the Dean vortices) in the pipe cross-section (Figure 5.11). This motion increases the resistance to the flow and, consequently, the losses of energy due to the larger pressure gradients. Consequently, in the direction of the bend outlet the maximum velocity is shifted towards the outer bend wall and the distortion of the velocity profile is higher (e.g., compare the isocurve of $U/U_0 = 1.2$ at the bend outlet for the v^2f turbulence model of Figure 5.11f with the one at the bend inlet of Figure 5.10f).

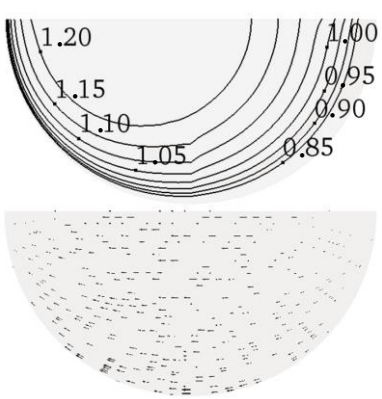
All turbulence models were able to reproduce qualitatively well the mean stream-wise velocity for both isocurves and vector fields at the bend inlet and the bend outlet as shown in Figure 5.10 and Figure 5.11, respectively. However, the turbulence models tested over-predicted the maximum velocity because the maximum isocurve of $U/U_0 = 1.2$ at the elbow inlet is much larger for all numerical solutions than the experimental data as presented in Figure 5.10 and we can assume that there are higher isocurves towards the maximum velocity as well as the elbow outlet for the isocurve of $U/U_0 = 1.05$ (see Figure 5.11).

The comparison of the turbulence intensity contours between numerical and experimental results at the bend inlet and the bend outlet are illustrated in Figure 5.12 and Figure 5.13, respectively. At the bend inlet, the EVM reproduced weaker turbulence intensities and the RSM reproduced higher turbulence regarding the experimental data. Nevertheless, all turbulence models reproduced well the flow behavior at the bend inlet regarding the turbulent intensities. On the other hand, all turbulence models did not reproduce well the flow behavior at the bend outlet. Similar to the bend inlet and outlet, the EVM reproduced weaker turbulence intensities and the RSM reproduced higher turbulence regarding the experimental data. Overall, the v^2f, the Lien cubic k-ε, and the RSM yielded better results of turbulence intensity in the stream-wise direction.

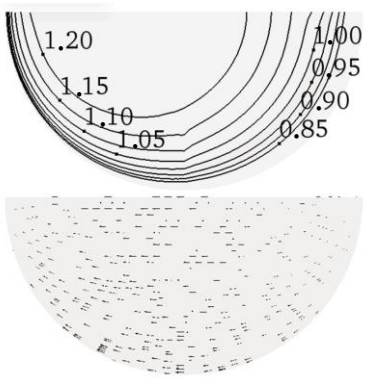
Figure 5.10 – Contours of the dimensionless mean stream-wise velocity (U/U_0) and velocity vectors at the bend inlet ($\alpha_b = 0°$).



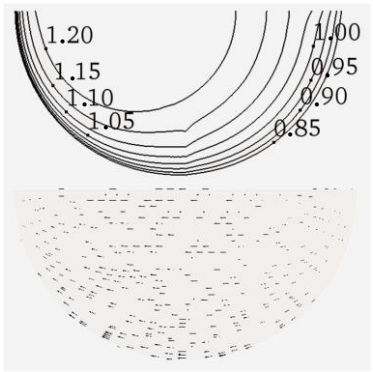
(a) Sudo *et al*. (1998).



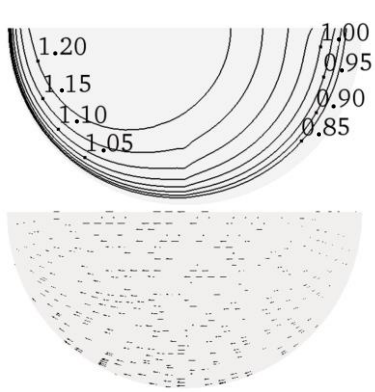
(b) *k-ε* of Lien and Leschziner.



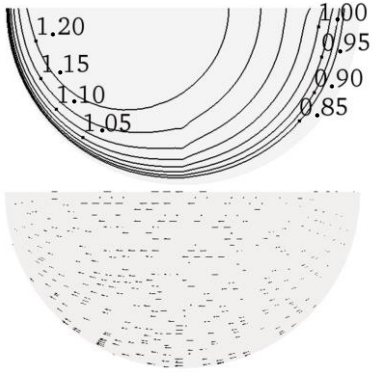
(c) *k-ε* of Launder and Sharma.



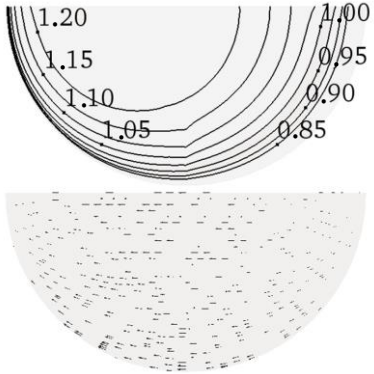
(d) SST *k-ω*.



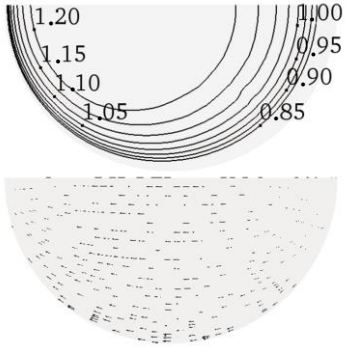
(e) *k-ω*.



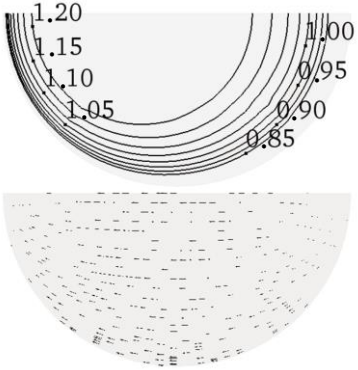
(f) *v2f*.



(g) Lien cubic *k-ε*.

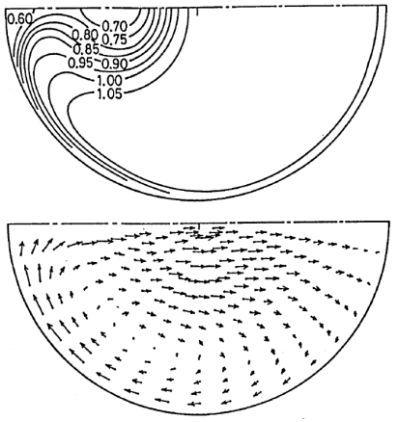


(h) RSM LRR.

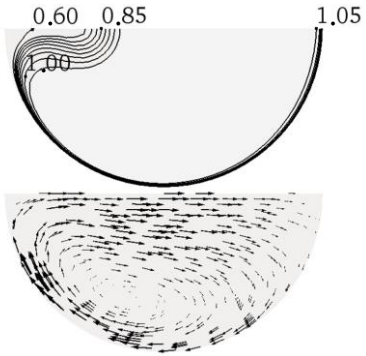


(i) RSM SSG.

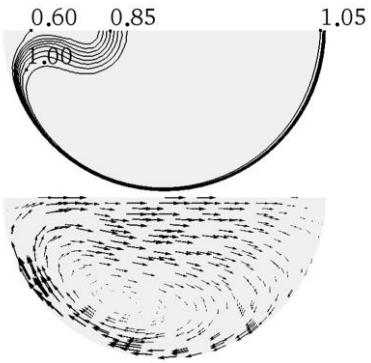
Figure 5.11 – Contours of the dimensionless mean stream-wise velocity (U/U_0) and velocity vectors at the bend outlet (α_b = 90°).



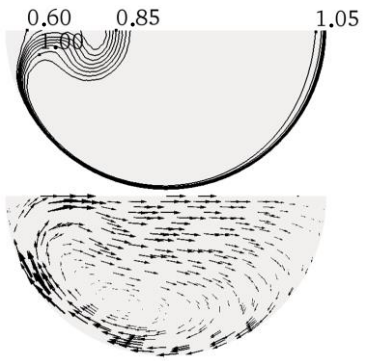
(a) Sudo *et al*. (1998).



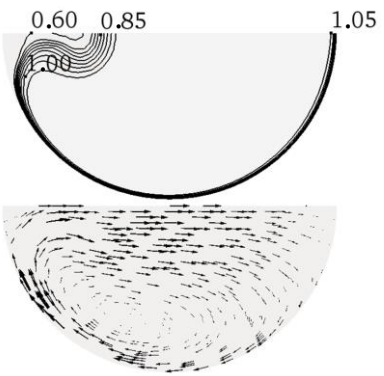
(b) *k*-*ε* of Lien and Leschziner.



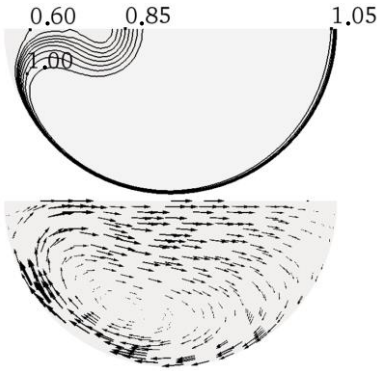
(c) *k*-*ε* of Launder and Sharma.



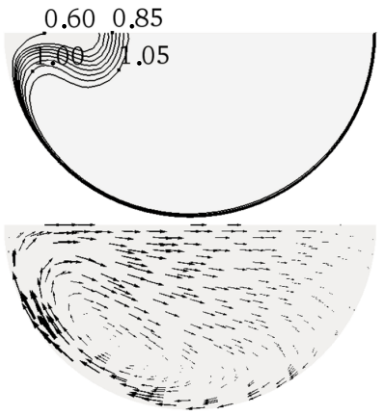
(d) SST *k*-*ω*.



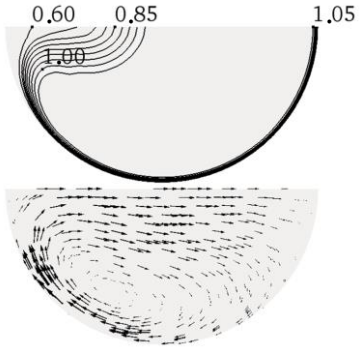
(e) *k*-*ω*.



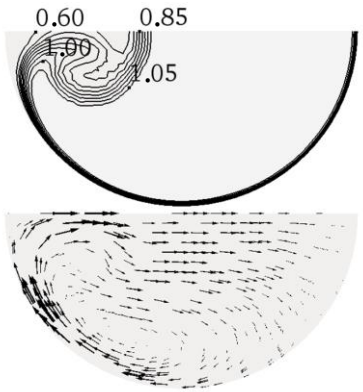
(f) *v2f*.



(g) Lien cubic *k*-*ε*.

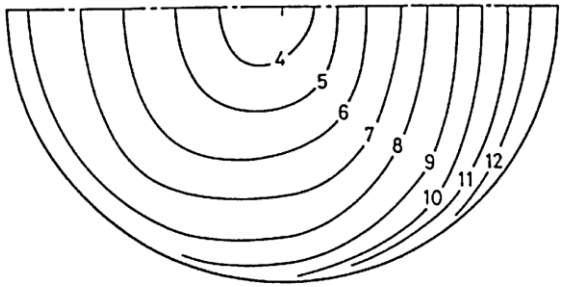


(h) RSM LRR.

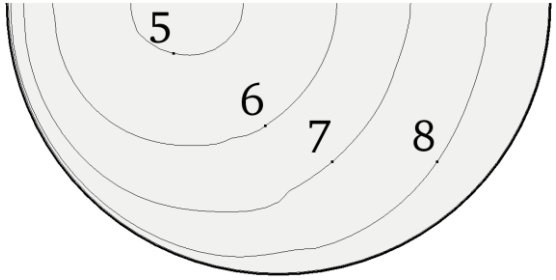


(i) RSM SSG.

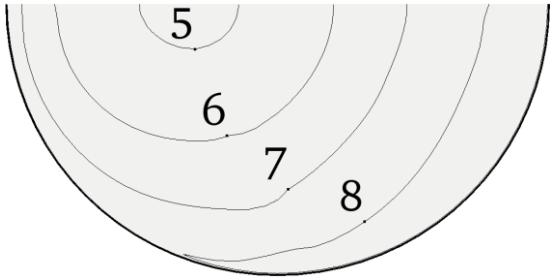
Figure 5.12 – Contours of the dimensionless turbulence intensity at the stream-wise flow direction ($\sqrt{u'^2}/U_0 \times 10^2$) at the bend inlet ($\alpha_b = 0°$).



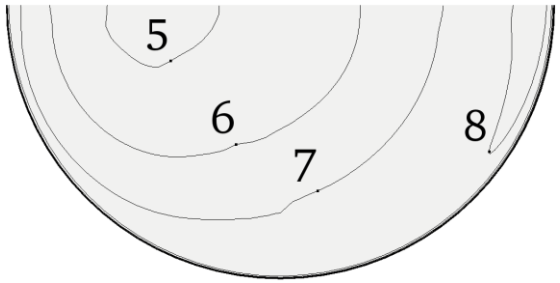
(a) Sudo *et al*. (1998).



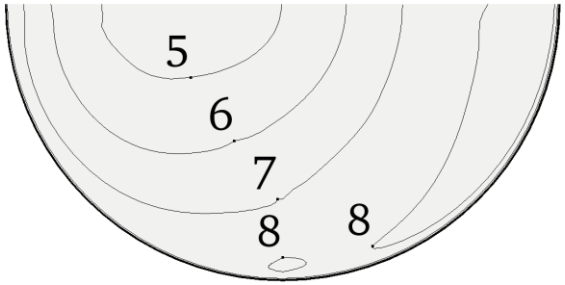
(b) *k-ε* of Lien and Leschziner.



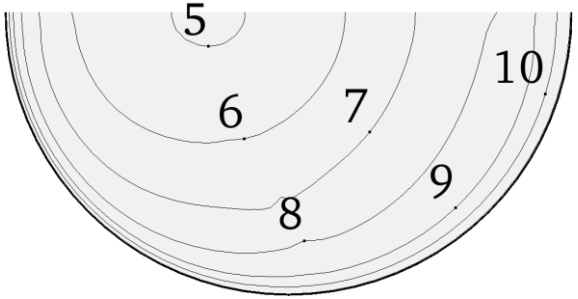
(c) *k-ε* of Launder and Sharma.



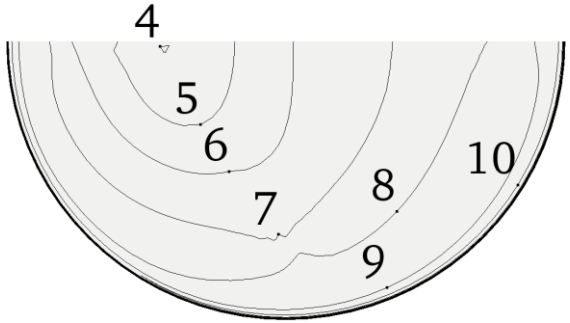
(d) SST *k-ω*.



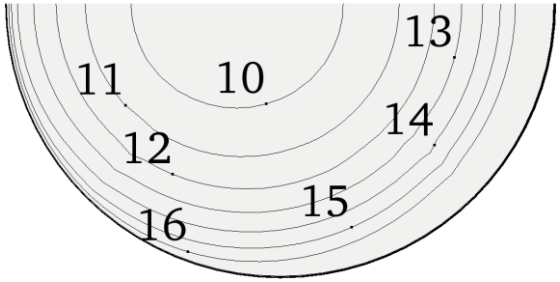
(e) *k-ω*.



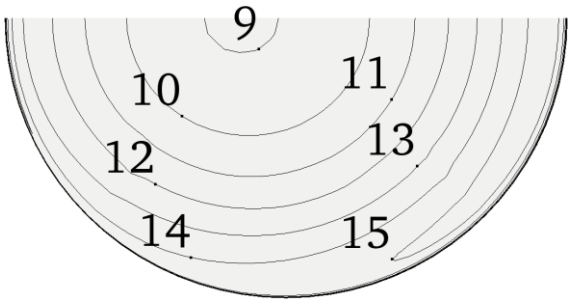
(f) *v2f*.



(g) Lien cubic *k-ε*.

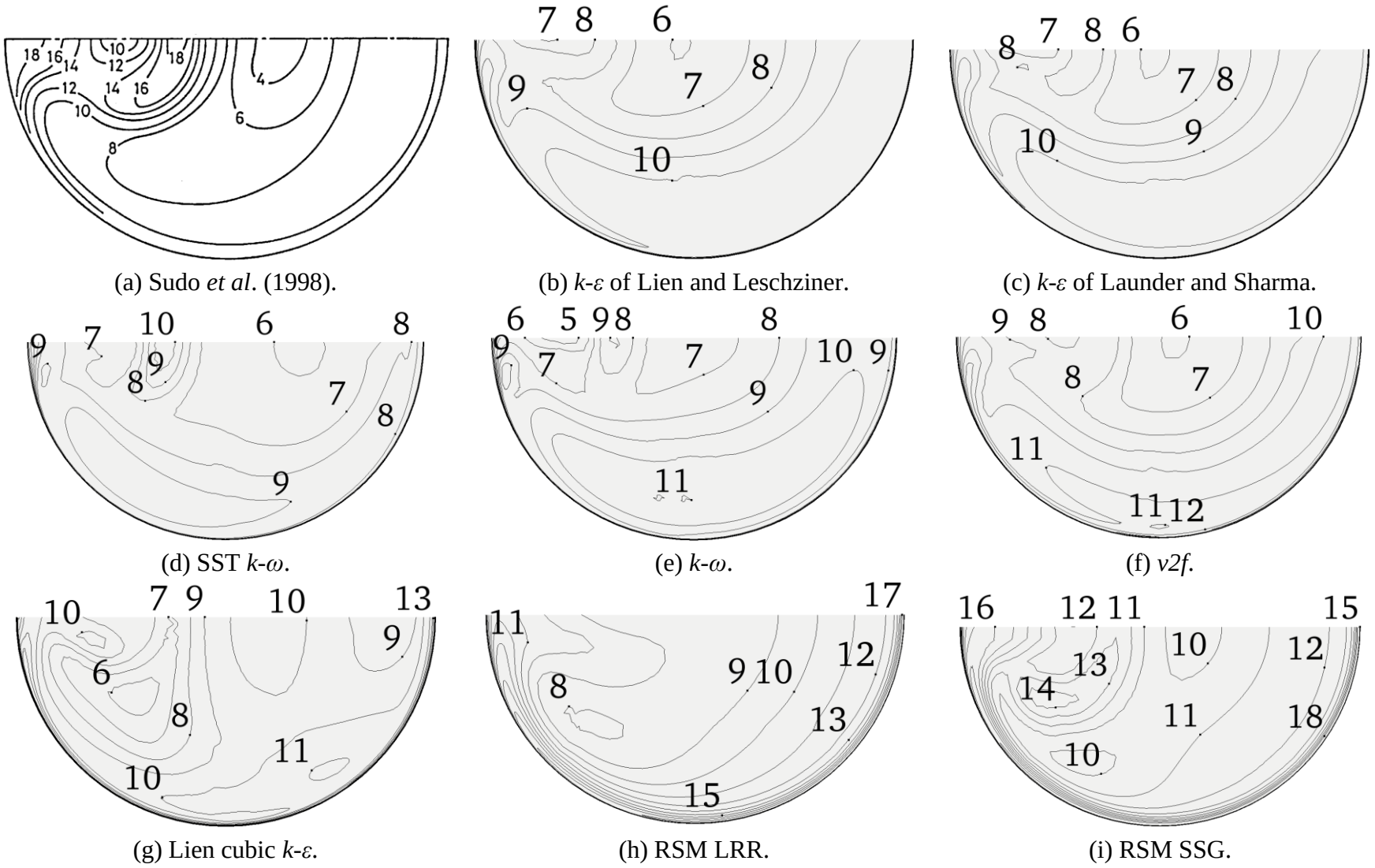


(h) RSM LRR.



(i) RSM SSG.

Figure 5.13 – Contours of the dimensionless turbulence intensity at the stream-wise flow direction ($\sqrt{u'^2}/U_0 \times 10^2$) at the bend outlet ($\alpha_b = 90°$).



(a) Sudo *et al*. (1998). (b) *k-ε* of Lien and Leschziner. (c) *k-ε* of Launder and Sharma.

(d) SST *k-ω*. (e) *k-ω*. (f) *v2f*.

(g) Lien cubic *k-ε*. (h) RSM LRR. (i) RSM SSG.

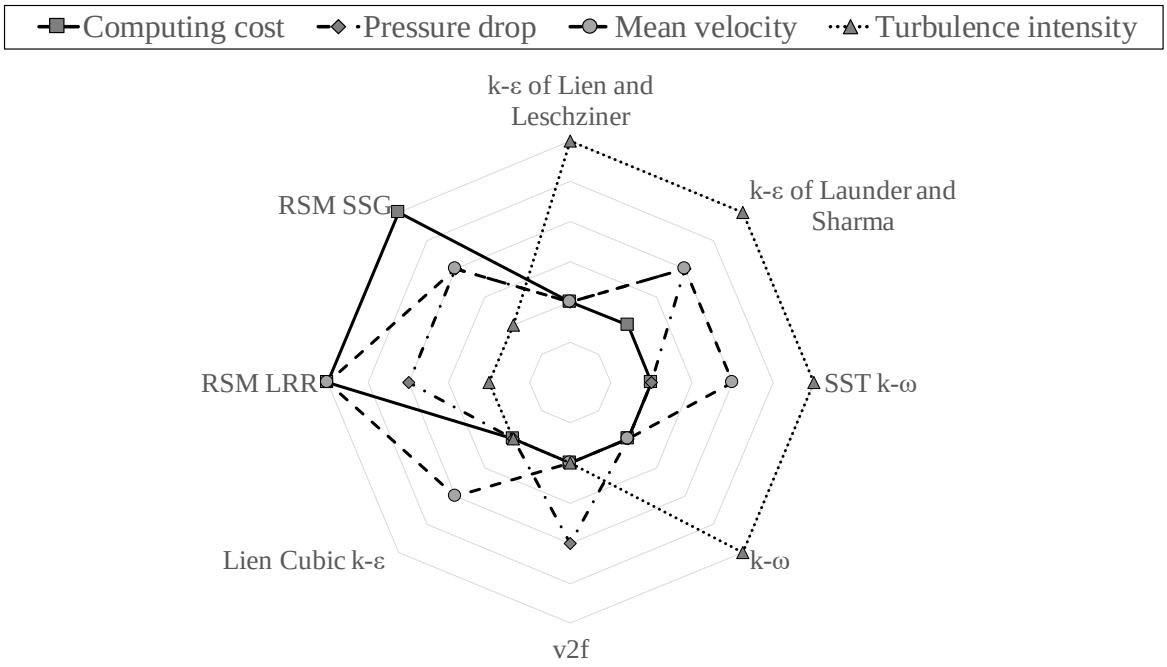
5.1.4 RANS turbulence model selection

In order to select the turbulence models, a qualitative criterion suggested in this study is illustrated in Figure 5.14. This is a radar chart in which the center represents the "target" value, i.e., when the given criteria are accurate enough, the values are close to the center, and when it is not the case the values are far from the center. Four criteria were selected for these analyses, according to the previous results: the computing cost (Table 5.3), the pressure drop (Table 5.3), the mean velocity (Figure 5.7-Figure 5.11) and the turbulence intensity (Figure 5.12 and Figure 5.13).

The linear EVM was better regarding the computing cost, while the non-linear EVM and the RSM demanded more computing cost because of the number of additional equations solved for each model. All turbulence models predicted quite well the pressure drop. The mean velocity was better predicted by the EVM. The RSM and the non-linear EVM better predicted the turbulence intensities than the linear EVM.

From Figure 5.14, the turbulence models selected were: the *k-ε* model of Launder and Sharma (the LRN *k-ε*), the SST *k-ω* model, the *v2f* model, and the RSM SSG. In the following paragraphs, we will show how each of these selected turbulence models performed to predict the airflow.

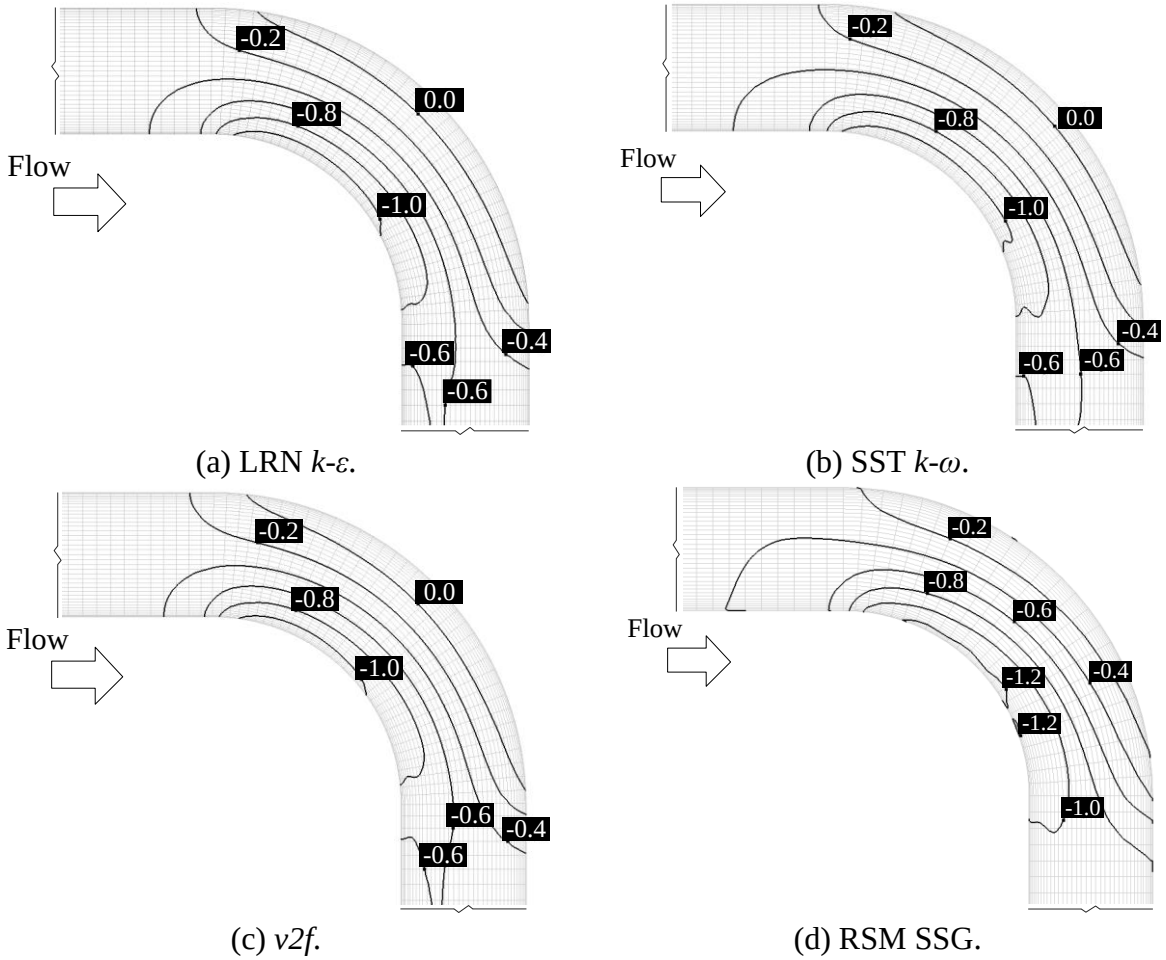
Figure 5.14 – RANS turbulence models selection.



As observed in Figure 5.15, all selected turbulence models described qualitatively well the pressure distribution in the elbow. As expected, the pressure is lower towards the inner bend wall and higher towards the outer bend wall, agreeing with the literature such as in the study of TANAKA *et al.* (2009).

The pressure coefficient, C_p, as defined in Equation (2.17), for the selected turbulence models against the experimental data of SUDO *et al.* (1998) is presented in Figure 5.16 in which the cross-section bend angle, ϕ_b, gives the position of the inner bend wall (ϕ_b = -90°), bottom bend wall (ϕ_b = 0°) and outer bend wall (ϕ_b = 90°). The reference position chosen by the authors was s_1/D = -17.6 (see Figure 5.2). The results for the LRN *k-ε*, *k-ω* SST and *v2f* turbulence models yielded very similar profiles of pressure coefficient with an *RMSE* up to 4%. The RSM SSG turbulence model also reproduced the behavior of the profile of pressure coefficient, however, the errors were higher than other turbulence models (especially in the downstream duct) that led to an *RMSE* up to 20%.

Figure 5.15 – Pressure coefficient at plane z = 0.

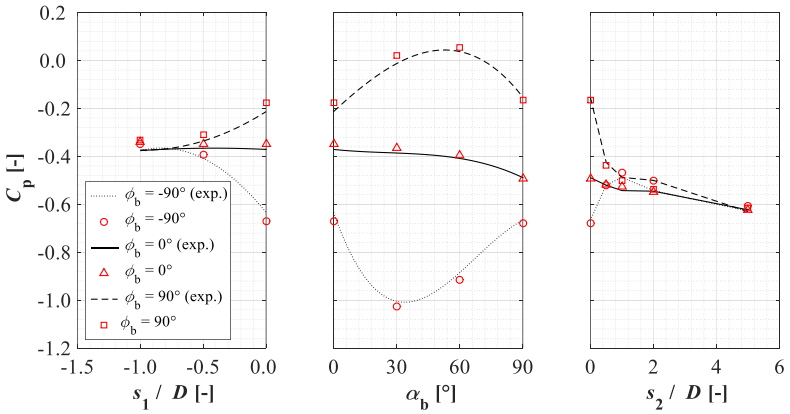


(a) LRN *k-ε*. (b) SST *k-ω*.

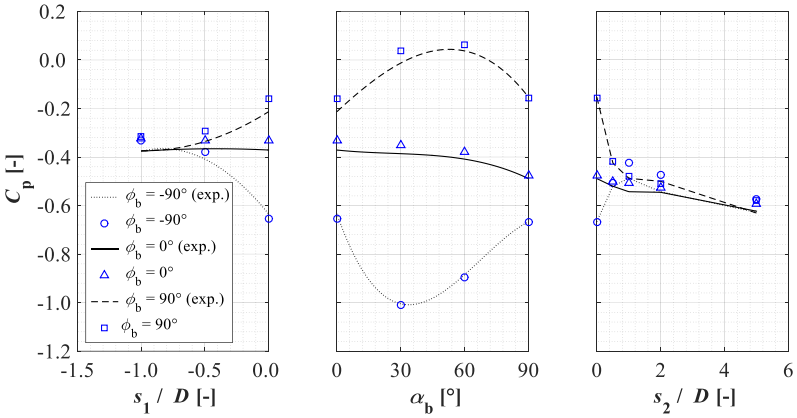
(c) *v2f*. (d) RSM SSG.

Figure 5.16 – Pressure coefficient profile.

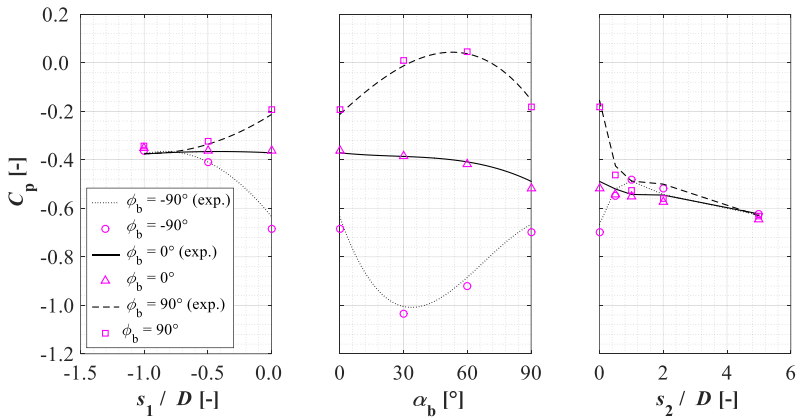
(a) LRN *k-ε*.



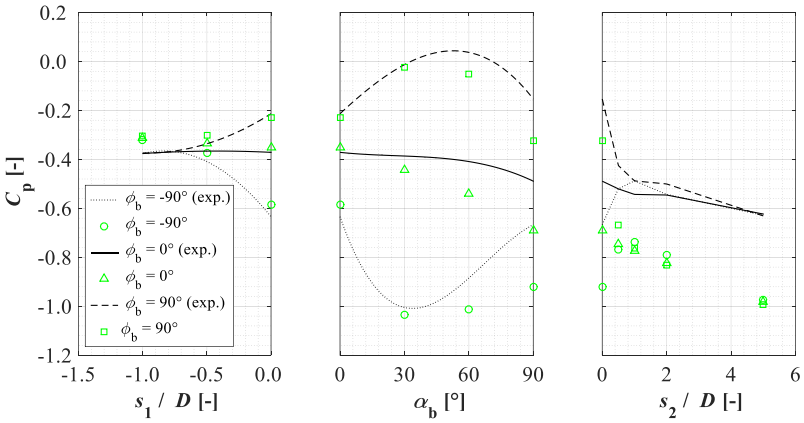
(b) SST *k-ω*.



(c) *v2f*.



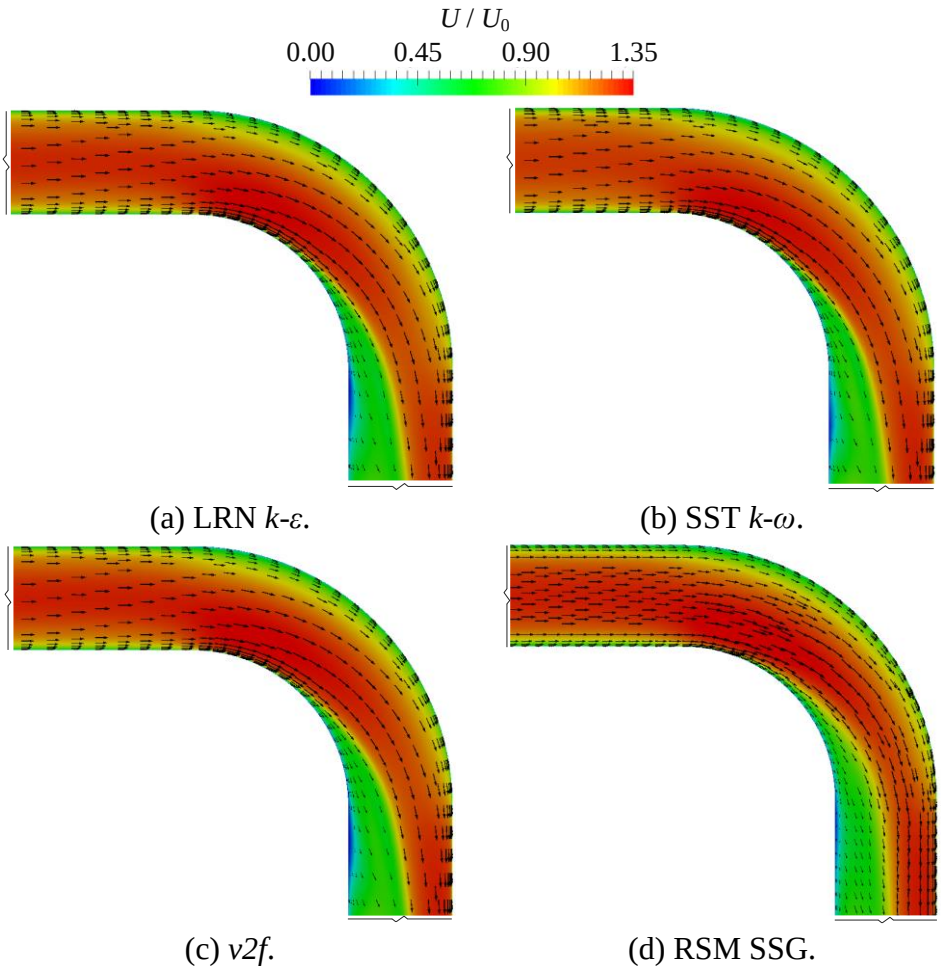
(d) RSM SSG.



The velocity field at the middle plane of the duct ($z = 0$) is presented in Figure 5.17. All selected turbulence models reproduced quite well the behavior of the airflow in the elbow and they agreed with results found in the literature such as the LES of TANAKA *et al.* (2009). The starting separation point near the bend outlet ($\alpha_b \approx 90°$) at the inner bend wall ($\phi_b = -90°$) and a reattachment point in the downstream straight duct ($s_2/D \approx 1$), agreed with the literature such as reported by TAKAMURA *et al.* (2012) for a higher Reynolds number. In addition, the vectors indicate an acceleration of the velocity field near the bend inner wall from the bend inlet to $\alpha_b \approx 30°$, followed by deceleration to the separation region and an acceleration of the velocity near the bend outer wall towards the bend outlet.

Figure 5.18 shows the airflow at the bend inlet ($\alpha_b = 0°$) and the bend outlet ($\alpha_b = 90°$) for all selected turbulence models. All of them were very similar and predict the Dean vortices at the bend outlet such as the LES of RÖHRIG *et al.* (2015) for almost the same Reynolds number.

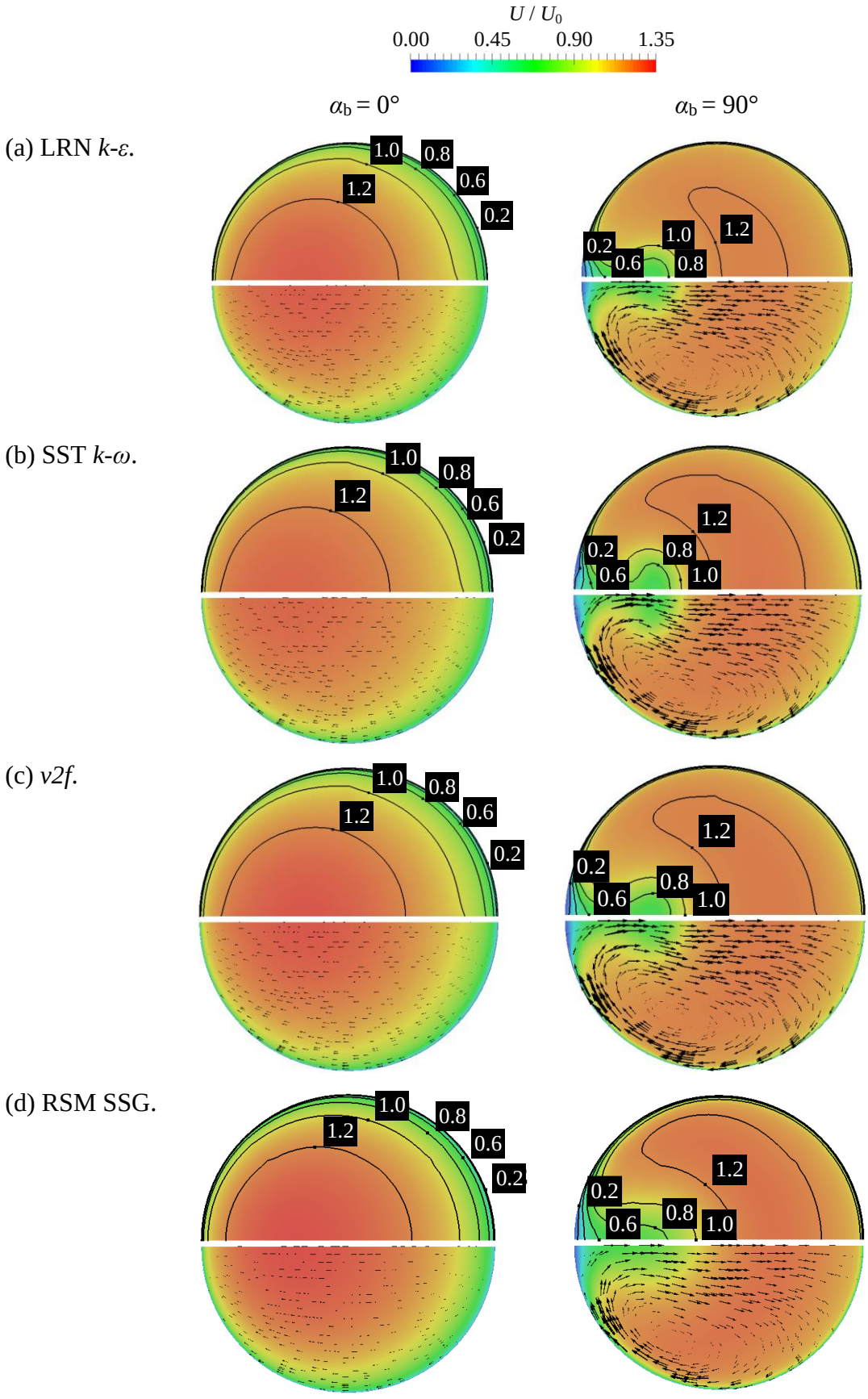
Figure 5.17 – Mean velocity at plane $z = 0$.



(a) LRN *k-ε*. (b) SST *k-ω*.

(c) *v2f*. (d) RSM SSG.

Figure 5.18 – Mean streamwise velocity at (a) bend inlet and (b) bend outlet.



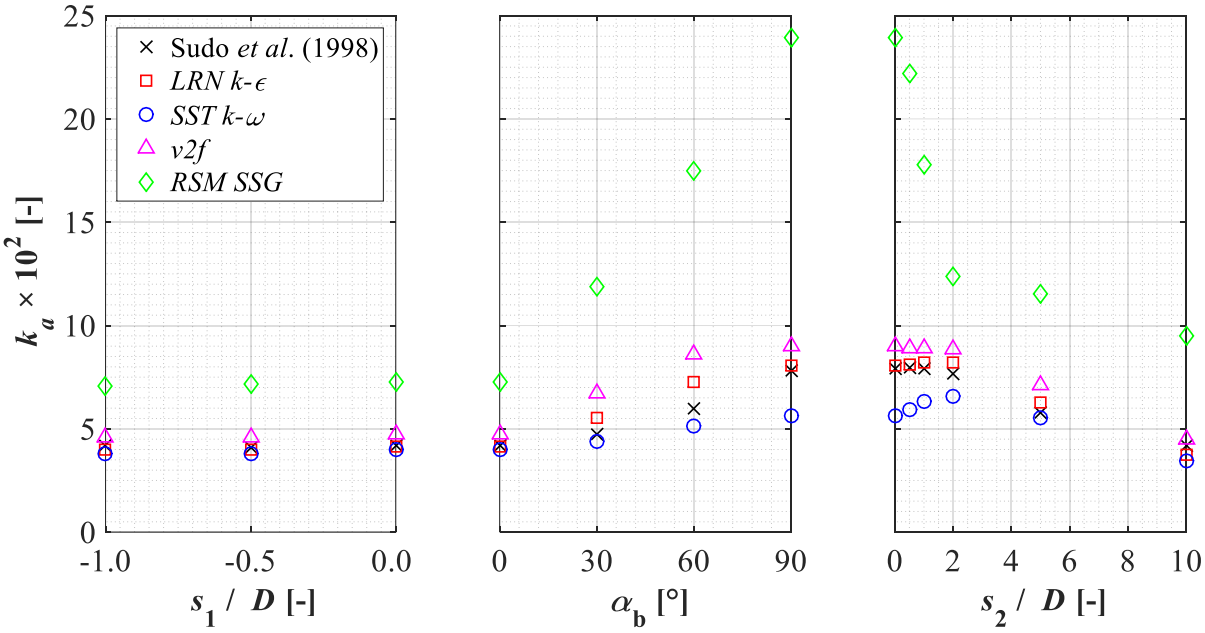
The total averaged dimensionless turbulent kinetic energy, ka, is given by (SUDO *et al.*, 1998):

$$ka = \frac{2}{D^2 {U_0}^2} \int_A kdA \ , \tag{5.13}$$

where A is the area of the duct cross-section.

The total averaged dimensionless turbulent kinetic energy for the selected turbulence models against the experimental data of SUDO *et al.* (1998) is presented in Figure 5.19. The RSM SSG turbulence model overestimated the turbulent kinetic energy corresponding to an *RMSE* of the order of 10%. On the other hand, the other turbulence models were very accurate leading to an *RMSE* of the order of 1%. However, the behavior of the profiles was like the experimental one, i.e., the increasing of the total averaged dimensionless turbulent energy along the elbow and the decreasing of it along the downstream duct.

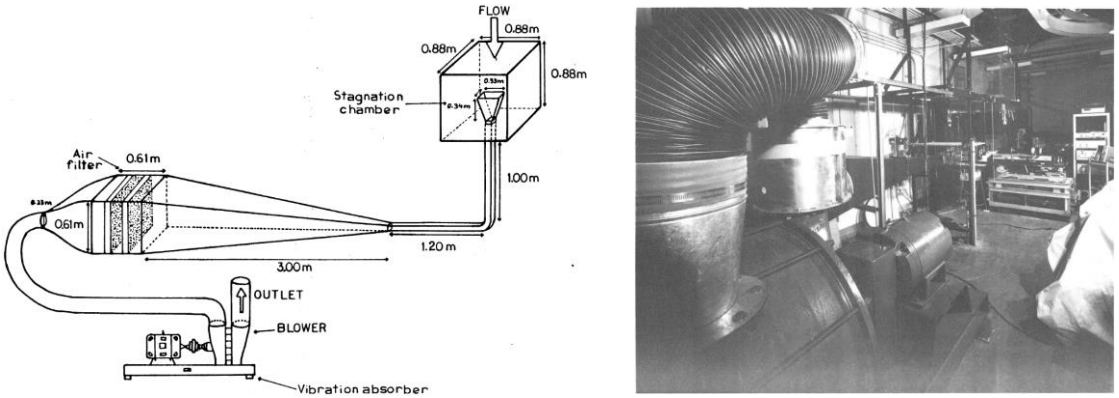
Figure 5.19 – Total averaged dimensionless turbulent kinetic energy.



5.2 THE INFLUENCE OF THE PWI MODELS TO PREDICT THE PARTICLE TRANSPORT IN A 90° BEND

KLIAFAS and HOLT (1987) performed experiments to measure the mean velocity and the associated Reynolds stress components for both the particulate and the fluid phases in the transport of micro glass spheres into the turbulent airflow through a vertical to horizontal square duct bend with a hydraulic diameter of 0.1 m and curvature ratio of 3.52, as shown in Figure 5.20. Air with glass spheres of 50 μm diameter and 2990 kg/m^3 particle density flows in the experimental duct system with a Reynolds number of 3.47×10^5.

Figure 5.20 – Schematic diagram and experimental apparatus of Kliafas and Holt (1987).



Source: KLIAFAS (1984).

Some of the numerical models from the literature based on the Eulerian-Lagrangian approach with a one-way coupling that chooses the benchmark of KLIAFAS and HOLT (1987) as validation cases are listed in Table 5.4. The modelling characteristics about the turbulence model, the particle-wall interaction (PWI) model and the geometry (if the model is two-dimensional, 2D, or three-dimensional, 3D) are presented in Table 5.4, as well as the bend angles for which results are available. As shown in Table 5.4, only the study of NJOBUENWU *et al*. (2013) provided results for bend angles up to 45°.

Table 5.4 – Literature numerical studies about results for different bend angles.

Reference	Turbulence Model	PWI model	2D/3D	α_b 0°	15°	30°	45°
TIAN *et al*. (2008)	RNG *k-ε*	Matsumoto & Saito	2D	☑	☑	☑	☒
SUN *et al*. (2011)	RNG *k-ε*	Brach & Dunn	2D	☑	☑	☑	☒
SUN *et al*. (2012)	RSM	Brach & Dunn	2D	☑	☑	☒	☒
NJOBUENWU *et al*. (2013)	RSM	Grant & Tabakoff	3D	☑	☑	☑	☑
ZAMANI *et al*. (2017)	RKE	Grant & Tabakoff	3D	☑	☑	☑	☒

In our study, the gas-phase of the flow was modeled by four different turbulence models: a low-Reynolds (LRN) k-ε model (the k-ε of Launder and Sharma), the SST k-ω model, the v^2f model, and the RSM SSG model. The particulate-phase was modeled by a Lagrangian particle tracking (LPT) formulation whose PWI was calculated using four different models: Brauer (BRAUER, 1980), Grant & Tabakoff (GRANT and TABAKOFF, 1975), Matsumoto & Saito (MATSUMOTO and SAITO, 1970) and Brach & Dunn (BRACH and DUNN, 1998) PWI models. The 2D and 3D simulations of mean streamwise velocities from the RANS-LPT/PWI combinations were performed and compared qualitatively and quantitatively to experimental and numerical data available in the literature.

5.2.1 Geometry and boundary conditions

The geometry and boundary conditions of this study are illustrated in Figure 5.21. The boundary conditions for the gas-phase were the same as those of the previous section (see section 5.1.1 for further information) with different airflow property values as provided in Table 5.5. In addition, as indicated in Figure 5.21 boundary conditions are different for the 2D and 3D simulations. As OpenFOAM® was designed as a 3D code and define all grids as such (GREENSHIELDS, 2018), 2D problems can be simulated in OpenFOAM® by applying the "empty" boundary condition, i.e., no boundary condition is required for the normal plane to this boundary condition (in this study, the plane $z = 0$). Concerning the 3D simulations, symmetric boundary condition was applied at the plane $z = 0$.

Figure 5.21 – Geometry and boundary conditions for the duct bend model.

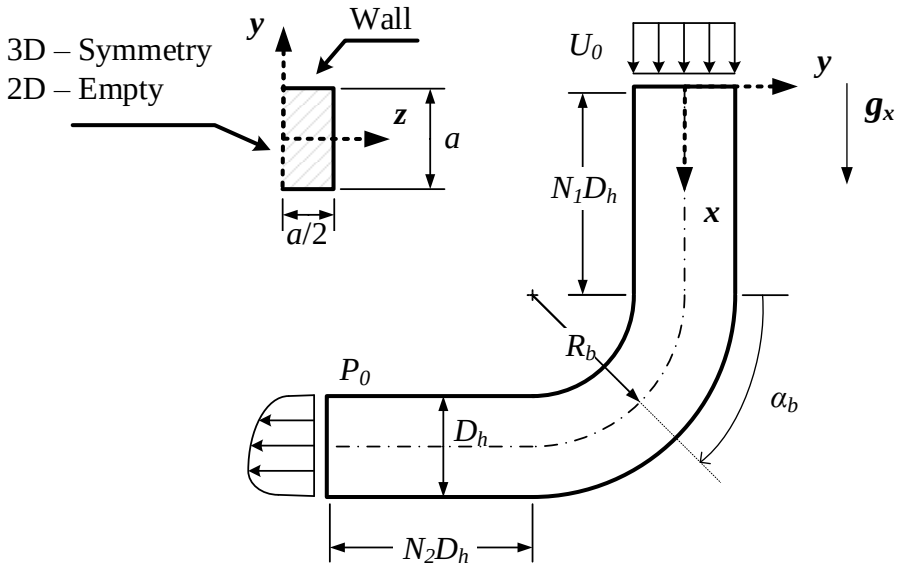


Table 5.5 – Elbow model: fluid and particulate phases.

Geometry	Fluid-phase* Air	Airflow	Particulate-phase* Particle
a = 100 mm	ρ = 1.2 kg/m^3	U_0 = 52.2 m/s	d_p = 50 μm
N_1 = 10 (50)	μ = 1.81×10^{-5} kg/ms	u_τ = 2.18 m/s	ρ_p = 2990 kg/m^3
N_2 = 12 (50)	ν = 1.51×10^{-5} m^2/s	f = 0.0141	Kn = 2.7×10^{-3}
R_b = 176 mm		Δp = 0.78 kPa (2.57 kPa)	τ_p = 2.3 ms
δ = 3.52		Re = 3.47×10^5	St = 2.4
α_b = 90°		De = 1.85×10^5	N_p = 1×10^5 (5×10^4)

* The values in parentheses refer to the 2D case.

The properties of the particulate and fluid phases are presented in Table 5.5. For the 2D cases, 5×10^4 monodispersed particles were injected once in a box from the position $x = 40D_h$ in the upstream pipe. For the 3D cases, 10^5 monodispersed particles were injected with the same local fluid velocity from the domain inlet during a given time interval. For both cases, the one-way coupled Lagrangian formulation was applied and the particle-fluid interactions considered were the drag, the shear-lift and the gravity forces (the direction of the gravity acceleration is indicated as g_x in Figure 5.21). APPENDIX B and APPENDIX C provide, respectively, a typical run of the Eulerian and Lagrangian OpenFOAM® solvers.

5.2.2 Mesh

The mesh was generated with the *blockMesh* tool of the OpenFOAM® code (see APPENDIX A). The model was divided into three parts as presented in Figure 5.22, Block 1 (the upstream duct), Block 2 (the elbow) and Block 3 (the downstream duct). We must have $N_z = 1$ for the 2D case, i.e., there is only one cell in the z-direction, and $N_z = N_y/2$ for the 3D case (the section of the duct is square, a symmetry boundary condition is applied on a vertical plane, and the other vertical plane is a wall).

The mesh generated for the 2D case is shown in Figure 5.23 (the number of cells in the stream-wise direction of the flow is $N_x = 5000$, and the number of cells in the transverse direction is $N_y = 200$). Close to the wall, the number of nodes increases in the normal direction from the inner wall with a cell-to-cell expansion ratio of 1.05 until the duct center (and then, it decreases with the same order of expansion ratio towards the outer wall). The cell length is uniform in the stream-wise direction and near the bend region, there is a lower cell length (where the size of the cells decreases smoothly with a cell-to-cell expansion ratio of 0.9, the cells are uniform in the bend and then the size of the cells increases after the bend outlet with a cell-to-cell expansion ratio of 1.1 until it reaches the same size as of the domain outlet).

Figure 5.22 – Block meshing strategy.

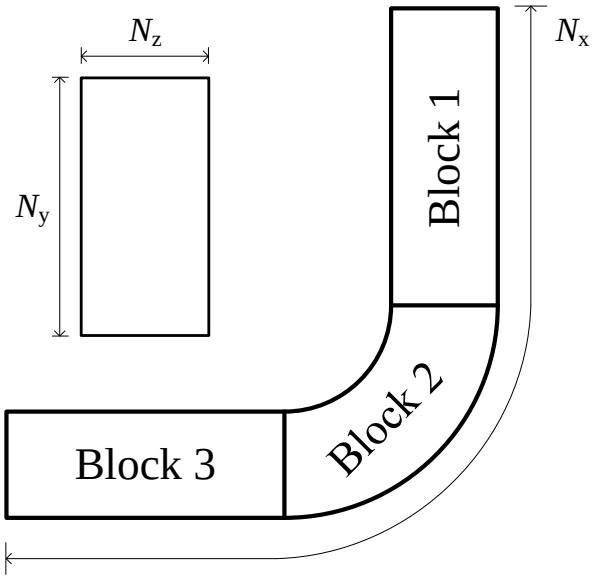
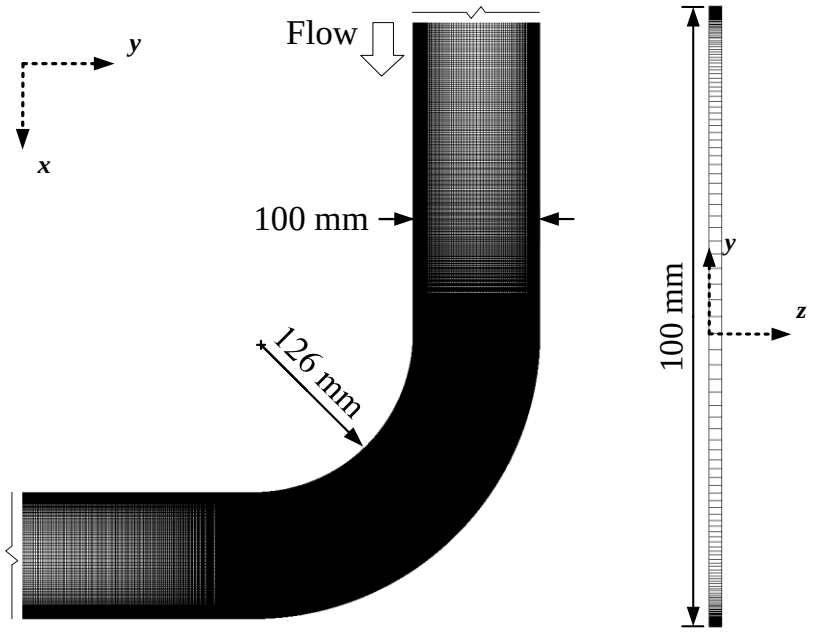


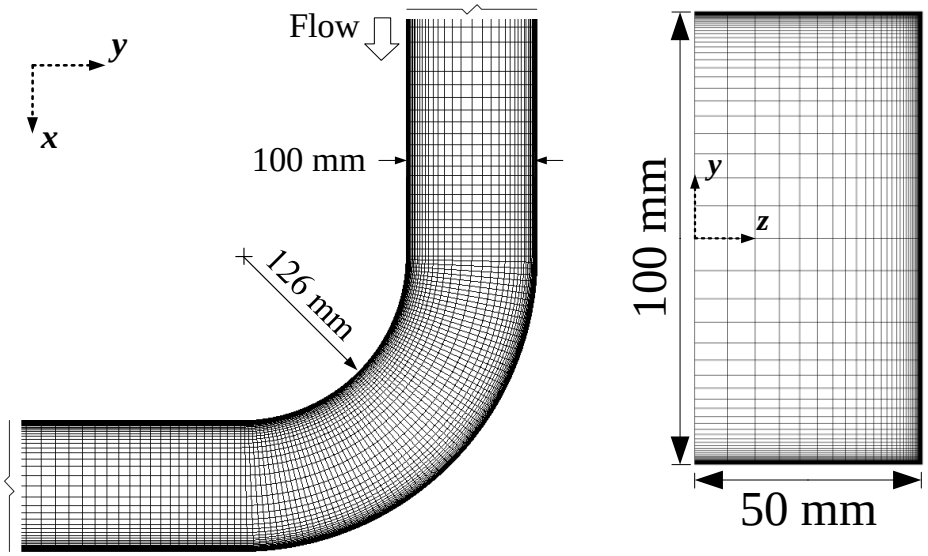
Figure 5.23 – 2D mesh (5000×200).



(a) Bend grid. (b) Domain inlet.

Similarly, the mesh generated for the 3D case is shown in Figure 5.24 with N_x = 264, N_y = 100 and N_z = 50. The length of the upstream duct was reduced from $50D_h$ (2D case) to $10D_h$ such as the downstream duct from $50D_h$ (2D case) to $12D_h$ to reduce the computing costs as well as the other authors. The expansion ratio in the wall-normal and stream-wise directions were similar to those adopted in the 2D case.

Figure 5.24 – 3D mesh (264×100×50).



(a) Bend grid. (b) Domain inlet.

5.2.3 Fluid-phase: V&V

The verification and validation (V&V) of this study were split into the fluid-phase and the particulate-phase. This section deals with the V&V of the fluid-phase following the systematic approach proposed in section 5.1.3 (V&V against the numerical data of literature listed in Table 5.4 and the experimental results regarding KLIAFAS and HOLT (1987)):

- the pressure drop, friction factor and computing efforts (Table 5.6);
- the stream-wise mean velocity profile in the elbow (Figure 5.25 and Figure 5.26);
- the RMSE relative to the mean velocity (Figure 5.27);
- the pressure coefficient contours (Figure 5.28);
- the mean velocity contours (Figure 5.29);
- the secondary flow (Figure 5.30).

The grid convergence study of the turbulent airflow was conducted with all turbulence models with the following grids: 5000×150, 5000×200 and 5000×250, for the 2D cases and, 264×80×40, 264×100×50 and 264×120×60, for the 3D cases[26]. The refinement method

[26] For the sake of brevity, the comparison between the grids will not be presented here.

followed the GCI method of ROACHE (1994) with a refinement ratio (r_m) near of 1.2. The comparison with respect to the pressure drop, friction factor and computing efforts between the turbulence models for both 2D and 3D cases with the selected grids 5000×200 and 264×100×50 for the 2D and 3D cases, respectively, is presented in Table 5.6.

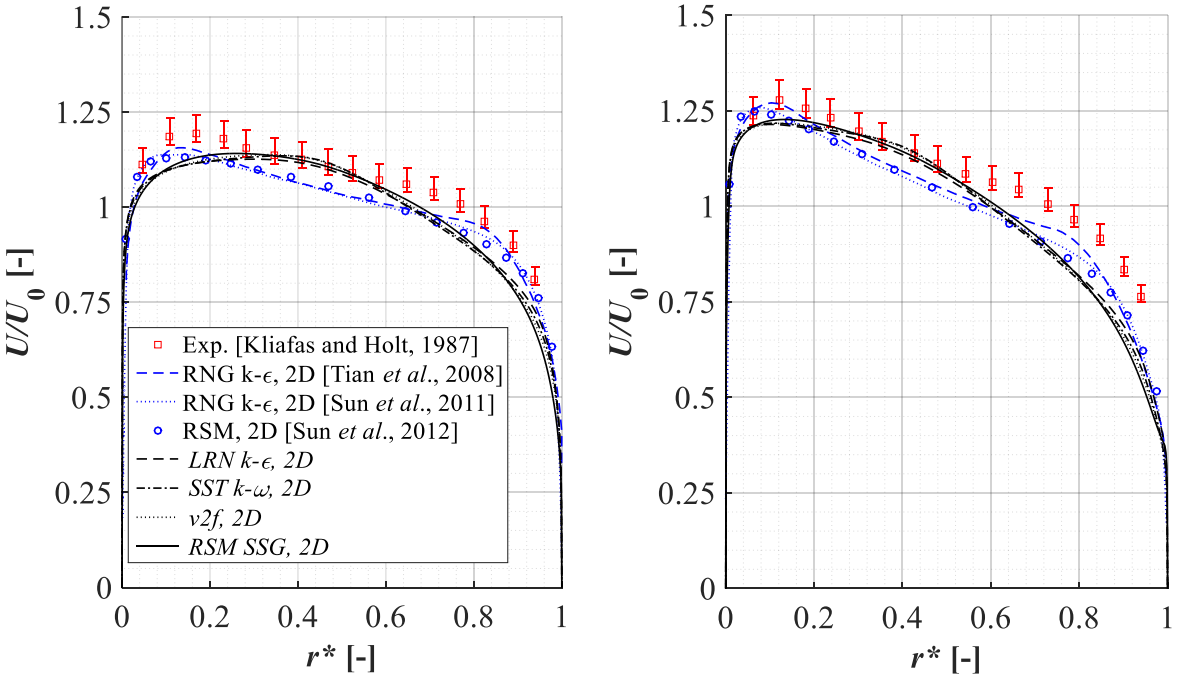
Table 5.6 – Comparison of the turbulence models tested.

Case	Turbulence model	Δp [d] [kPa] (E_r [e])	f [f] (E_r [e])	N° of iterations [g]	t [h] [h]
2D (5000×200) [a]	**LRN *k-ε***	1.34 (-48%)	0.0099 (-30%)	6×10^5	111
	SST *k-ω*	1.07 (-58%)	0.0095 (-32%)	5×10^5	128
	v^2f	1.10 (-57%)	0.0087 (-38%)	7×10^5	160
	RSM SSG	2.30 (-10%)	0.0209 (+48%)	8×10^5	473
3D (264×100×50) [b]	**LRN *k-ε***	0.80 (+3%)	0.0124 (-12%)	4×10^4	43
	SST *k-ω*	0.65 (-17%)	0.0092 (-35%)	6×10^4	65
	v²f	0.75 (-4%)	0.0109 (-23%)	1×10^5	73
	RSM SSG	0.83 (+6%)	0.0191 (+35%)	7×10^4	98

[a] $N_x\times N_y$.
[b] $N_x\times N_y\times N_z$.
[c] *k-ε* of Launder and Sharma.
[d] The calculated pressure drops with Equation (2.10) are 2.57 kPa (2D) and 0.78 kPa (3D).
[e] Relative error calculated with Equation (5.11).
[f] The calculated Darcy friction factor with Equation (2.11) is 0.0141.
[g] Number of iterations to achieve the convergence.
[h] Total computing time with parallel computation (2 Deca Core Intel Xeon CPU E5-2660 v3s).

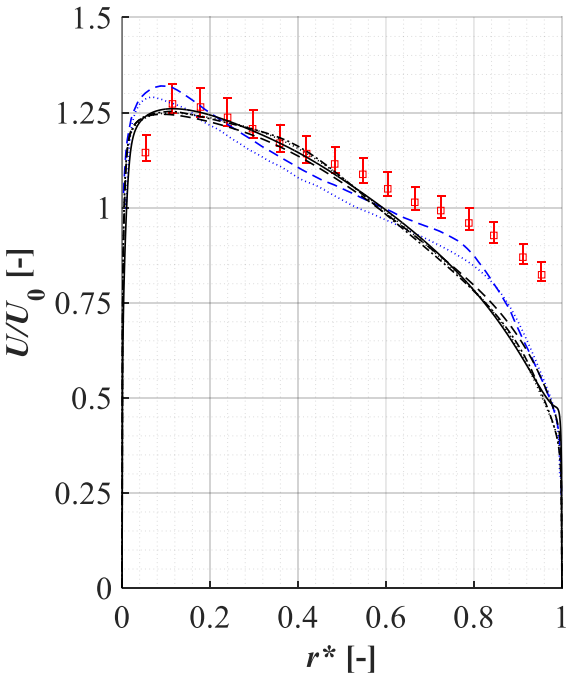
In general, all turbulence models of the 3D cases were accurate regarding the pressure drop with the absolute maximum relative error equal to 17%, while the maximum absolute relative error for the 2D cases was 58%. Additionally, regarding the Darcy friction factor, the 3D shows errors lower than 2D in general. The computational effort of the 3D cases demanded less computing cost than the 2D cases because the domain of the 2D is much bigger than the 3D and the 3D grid is only slightly bigger. The mean streamwise velocity of the fluid-phase is shown in Figure 5.25 and Figure 5.26.

Figure 5.25 –2D case: the fluid-phase mean streamwise velocity profiles through the elbow.

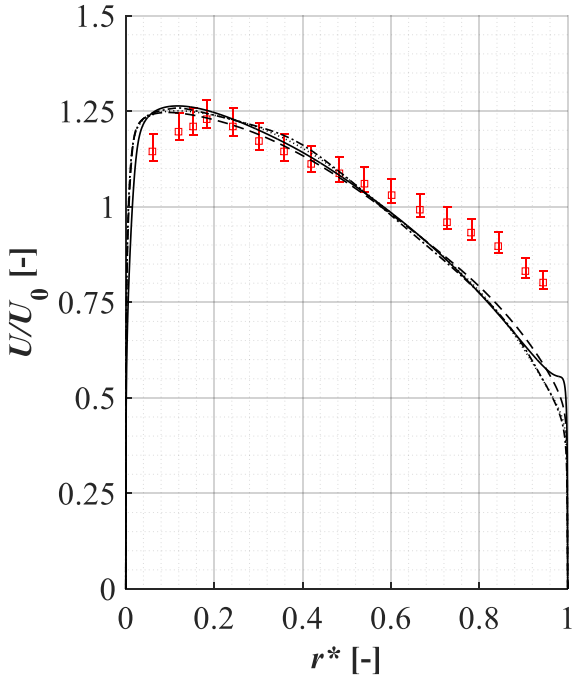


(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

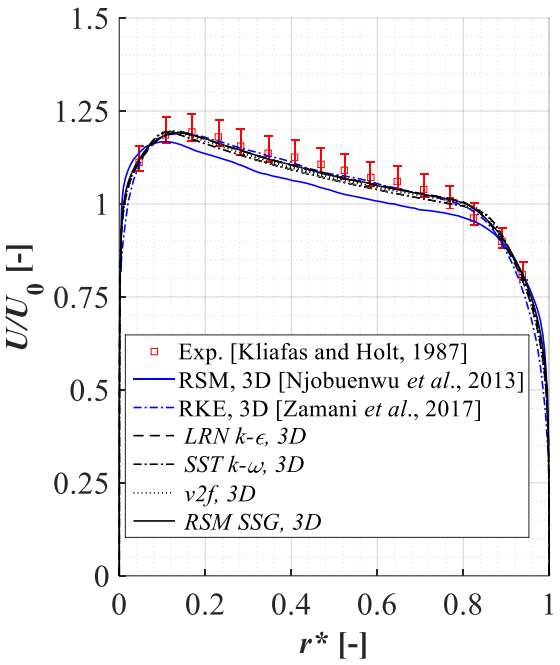


(c) $\alpha_b = 30°$.

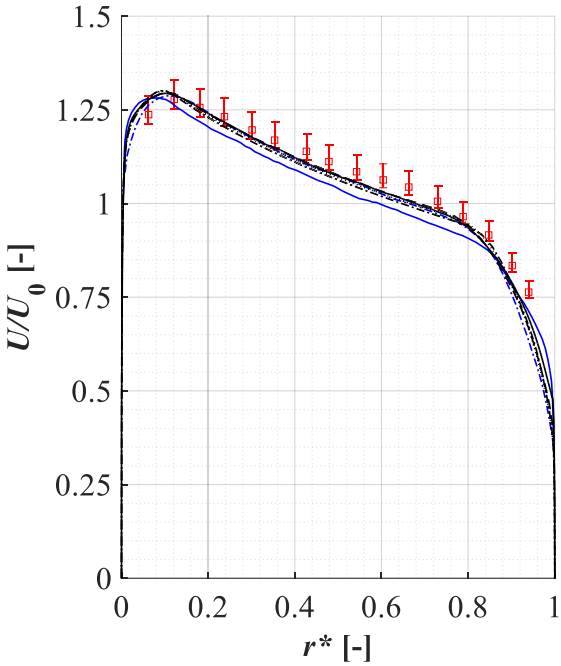


(d) $\alpha_b = 45°$.

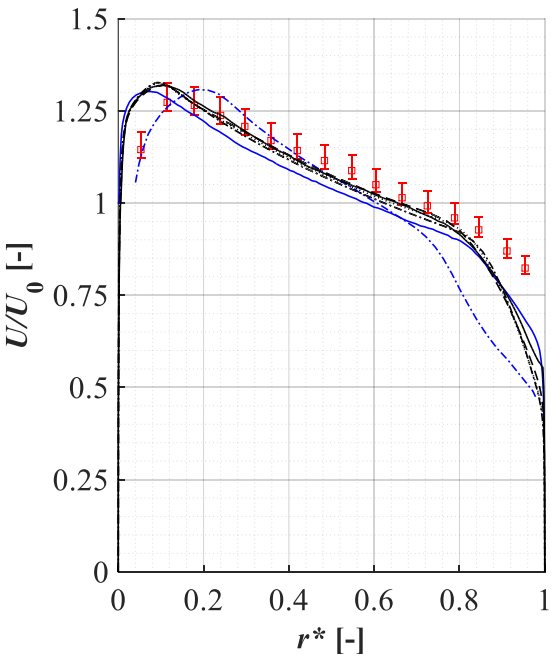
Figure 5.26 – 3D case: the fluid-phase mean streamwise velocity profiles through the elbow.



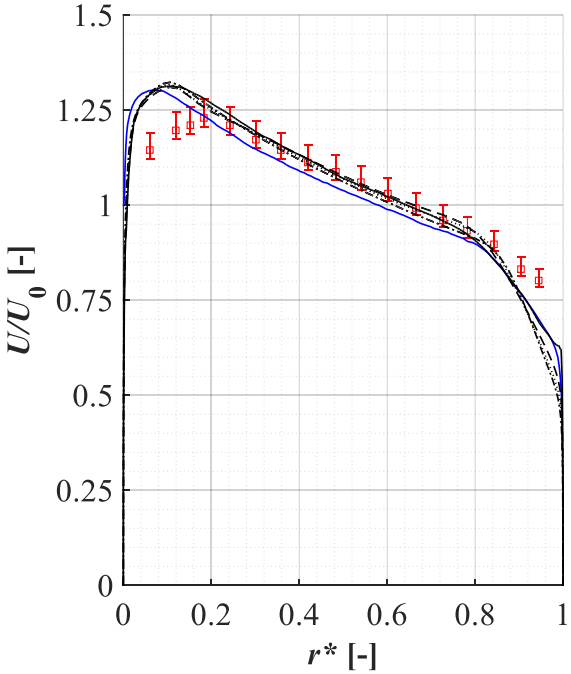
(a) $\alpha_b = 0°$.



(b) $\alpha_b = 15°$.



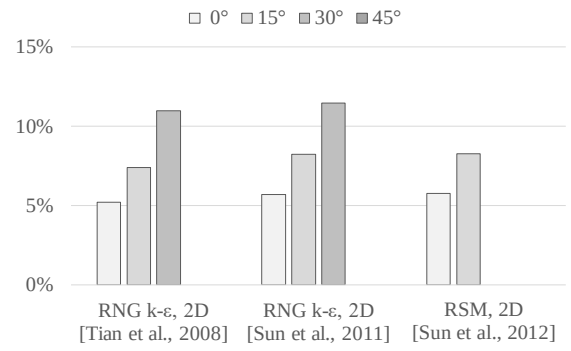
(c) $\alpha_b = 30°$.



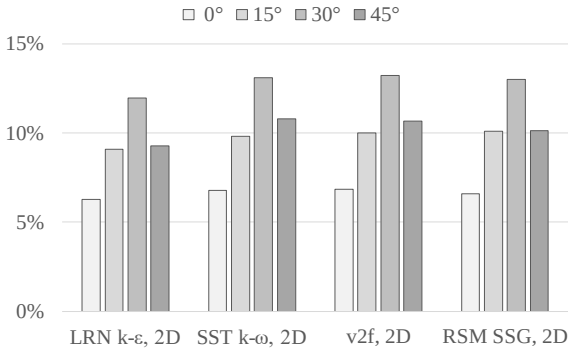
(d) $\alpha_b = 45°$.

The comparison between the measured and computed stream-wise fluid-phase mean velocity profiles along the plane $z = 0$ (the vertical midplane) of the bend at angles 0°(bend inlet), 15°, 30° and 45°, normalized with the bulk velocity ($U_0 = 52.19$ m/s), were presented in Figure 5.25 and Figure 5.26 for the 2D and 3D cases, respectively. The dimensionless radius, r^*, is determined according to Equation (2.3), ranging from the inner bend wall ($r^* = 0$) to the outer bend wall ($r^* = 1$). The corresponding RMSE of the fluid-phase mean stream-wise velocity is indicated in Figure 5.27.

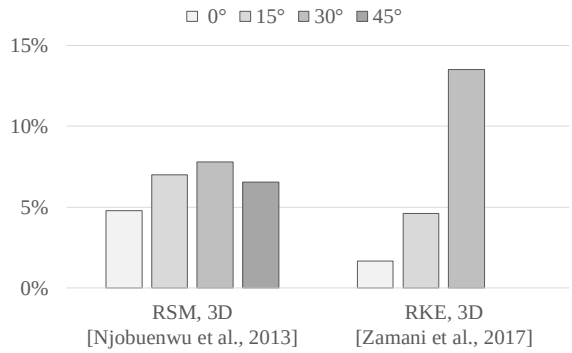
Figure 5.27 – RMSE for the fluid-phase mean stream-wise velocity through the elbow.



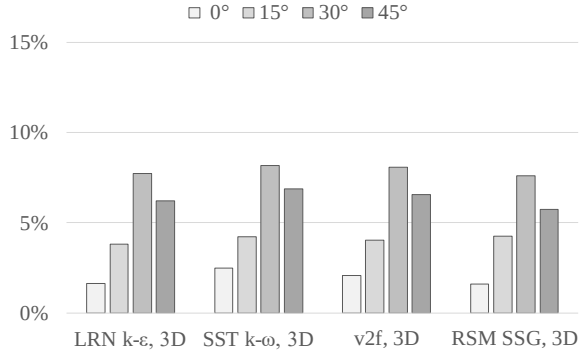
(a) Literature (2D cases).



(b) Present study (2D cases).



(c) Literature (3D cases).



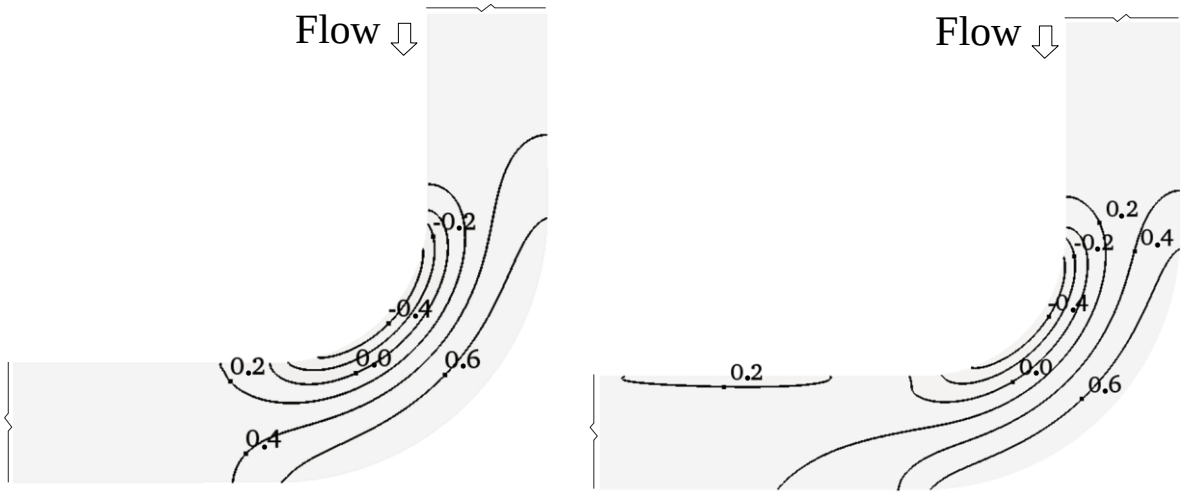
(d) Present study (3D cases).

In relation to the experimental data, both Figure 5.25 (2D cases) and Figure 5.26 (3D cases) indicate that the turbulence models tested are able to represent relatively well the acceleration of the flow along the inner bend wall. Nevertheless, the turbulence models of the 2D cases overestimated the deceleration of the flow along the outer bend wall, especially at bend angles of 30° and 45°. Still, the turbulence models of the 3D cases represented better this deceleration with a higher degree of accuracy.

As indicated in Figure 5.27, the RMSE regarding the mean stream-wise velocity of the fluid-phase of the 2D case was below 13% for all turbulence models tested, whereas for the 3D case the RMSE was below 8%, with the highest error being found at 30°. According to Figure 5.27b (the 2D cases), despite the behavior of the RMSE is very similar among the four turbulence models, the LRN *k*-*ε* is the turbulence model that produces the lowest errors, and according to Figure 5.27c (the 3D case) the RSM SSG is the best turbulence model with error even better as those from the 3D simulation with the RSM of literature (Figure 5.27a).

The isovalues of the pressure field at the vertical midplane ($z = 0$) for the LRN *k*-*ε* turbulence model are presented in Figure 5.28 (with the reference pressure being the atmospheric pressure)[27]. As expected, the pressure is lower towards the inner bend wall and higher towards the outer bend wall agreeing with the literature such as in the study of TANAKA *et al.* (2009) that deals with the computation of a turbulent flow in a pipe elbow with a LES model, revealing the presence of the adverse pressure gradient. It appears that for the pressure field there is no significant difference between the 2D and 3D cases.

Figure 5.28 – Pressure coefficient at the plane $z = 0$ with the LRN *k*-ε turbulence model.

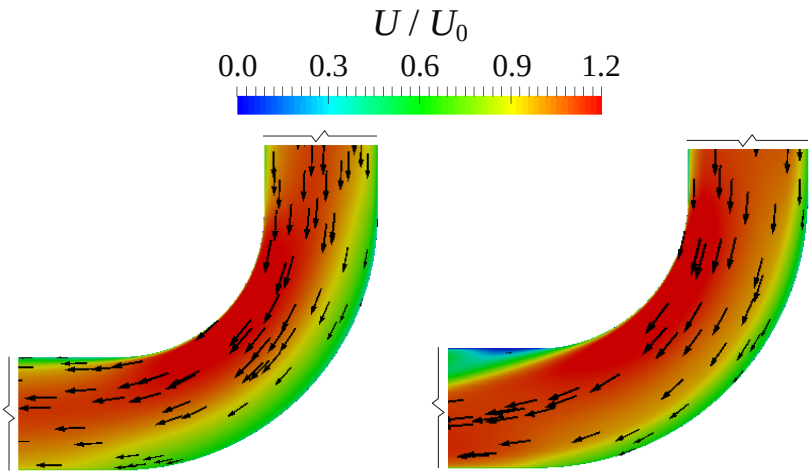


(a) 2D case. (a) 3D case.

The mean velocity field for the 2D and 3D cases is presented in Figure 5.29. The separation region near the bend outlet appears only with the 3D model (Figure 5.29b), therefore the turbulent flow in the downstream region predicted by the 2D model is not accurate because the influence of the side walls is not present.

[27] The other turbulence models for both 2D and 3D yielded very similar to the LRN *k*-*ε* turbulence model.

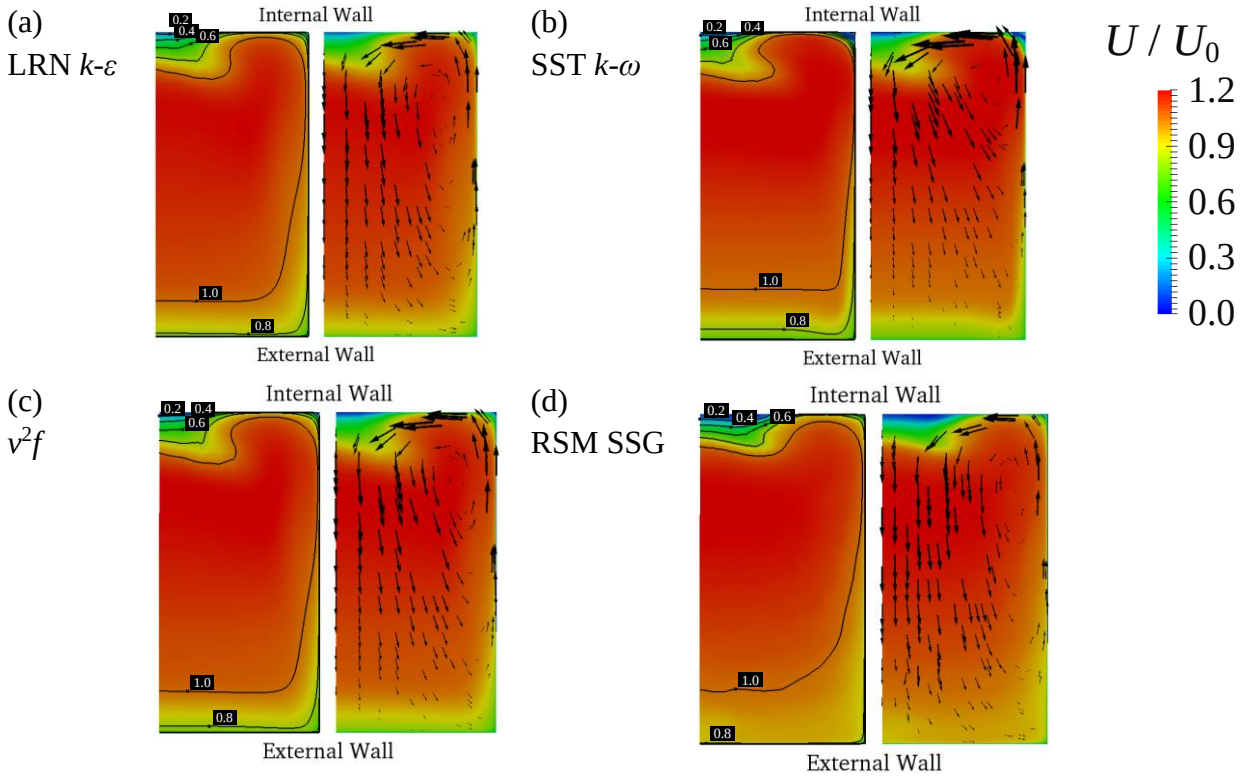
Figure 5.29 – Mean velocity at the plane $z = 0$ with the LRN k-ε turbulence model.



(a) 2D case. (b) 3D case.

Finally, the 3D cases revel another result that confirms the verification of the modelling: the Dean vortices as indicated in Figure 5.30 for all turbulence models tested at the bend outlet angle. As expected, we note the distortion of the flow contours and the maximum velocity contour shifting towards the outer bend wall such as the isocurve of $U/U_0 = 1.0$. The RSM model predicted a larger separation region along the inner bend wall when compared with the EVM, however, behaviors of the turbulence models are similar.

Figure 5.30 – Mean velocity at bend outlet ($\alpha_b = 90°$).



5.2.4 Particulate-phase: V&V

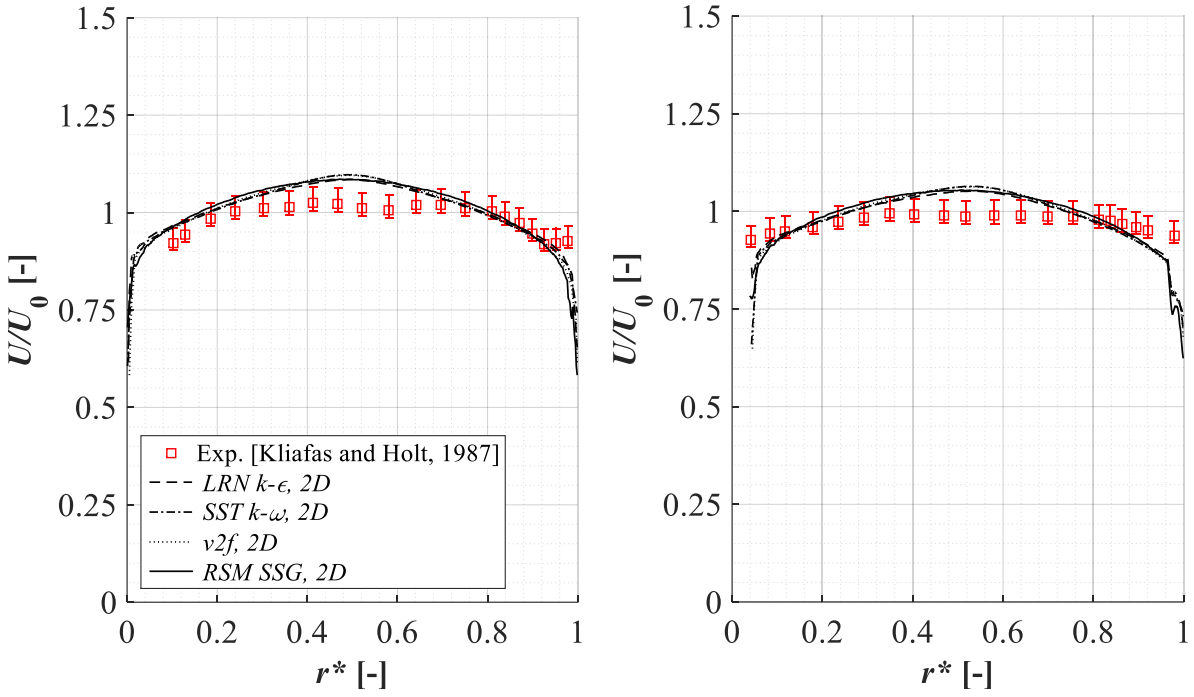
In order to implement the PWI models proposed in this study, the library of the standard LPT solver of OpenFOAM® code must be customized as described in APPENDIX C with a typical run of this new customized solver. Furthermore, to calculate the mean stream-wise velocity profiles of the particulate phase, the duct was divided into concentric bins, and the statistical results (mean stream-wise velocity profiles for the particulate phase) were calculated for each bin. The mean stream-wise velocity profiles for the particulate-phase from the simulation carried out with the PWI models of Brauer (BRAUER, 1980), Grant & Tabakoff (GRANT and TABAKOFF, 1975), Matsumoto & Saito (MATSUMOTO and SAITO, 1970) and Brach & Dunn (BRACH and DUNN, 1998) are presented in Figure 5.31-Figure 5.38.

According to Figure 5.31, the results from all RANS models combined with Brauer's PWI model for the 2D case agreed fairly well with the experimental data at bend angles up to 15°, although the numerical velocity profiles were not as uniform as the experimental ones for the relative radius position (r^*) between 0.2 and 0.8. On the other hand, for the 3D case (Figure 5.32) the mean velocity profiles fit better the experimental data at the lowest bend angles (0° and 15°). Regarding the highest bend angles (30° and 45°), the numerical models of the 2D case predicted lower particulate-phase mean velocity than the experiments in the inner and the outer bend walls.

Similar observations can be addressed regarding the numerical results from the RANS-LPT/PWI combinations with Grant & Tabakoff's PWI model. However, it can be noted in Figure 5.33 and Figure 5.34 that discrepancies between numerical and experimental results in the outer wall region of the bend for 30° and 45° are higher with this PWI model for both 2D and 3D cases than those with Brauer's PWI model (Figure 5.31 and Figure 5.32). At the highest bend angles, 3D simulation using the RSM turbulence models from literature (NJOBUENWU *et al.*, 2013) performed better than the turbulence models tested in this study regarding the particulate-phase mean velocities in the outer bend wall. In contrast, the 3D simulation using the RKE turbulence model (ZAMANI *et al.*, 2017) produced the greatest discrepancies for the deceleration of the particulate-phase in the inner and outer bend walls.

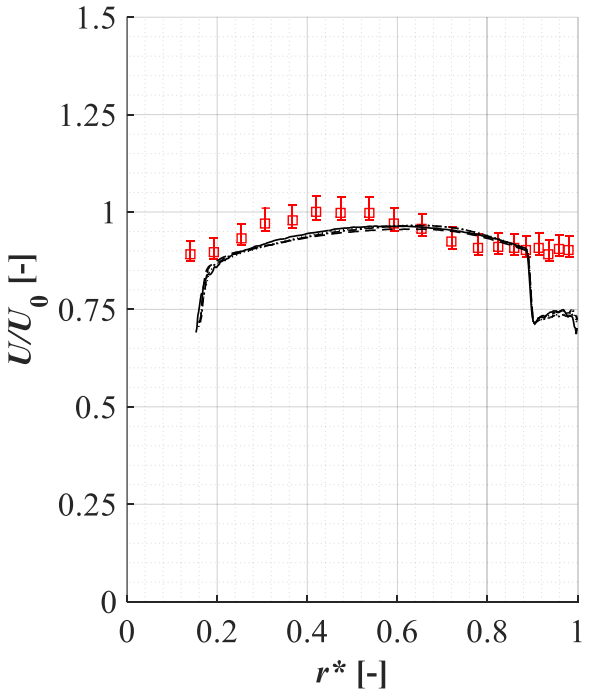
The mean streamwise velocity profiles for the particulate-phase resulting from the simulations performed with Matsumoto & Saito PWI model, presented in Figure 5.35 and Figure 5.36, were similar to those illustrated in Figure 5.31-Figure 5.34 at bend angles of 0° and 15° for both 2D and 3D cases.

Figure 5.31 – Particulate-phase (2D case): mean streamwise velocity profiles along the elbow, using Brauer's PWI model (BRAUER, 1980).

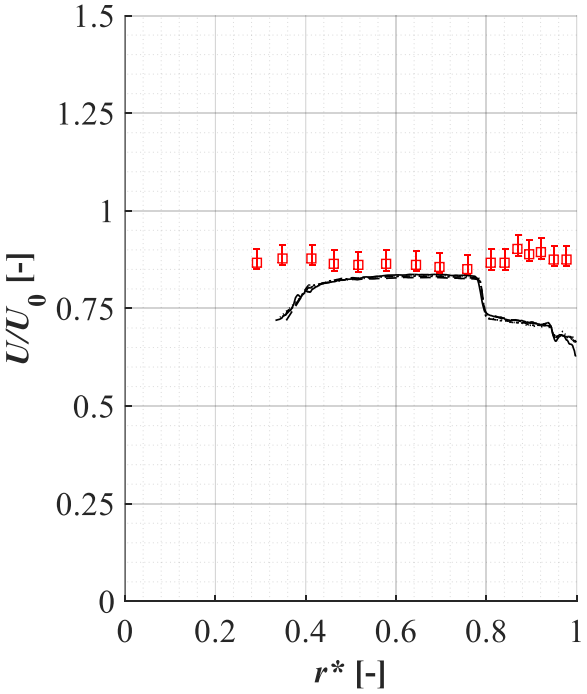


(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

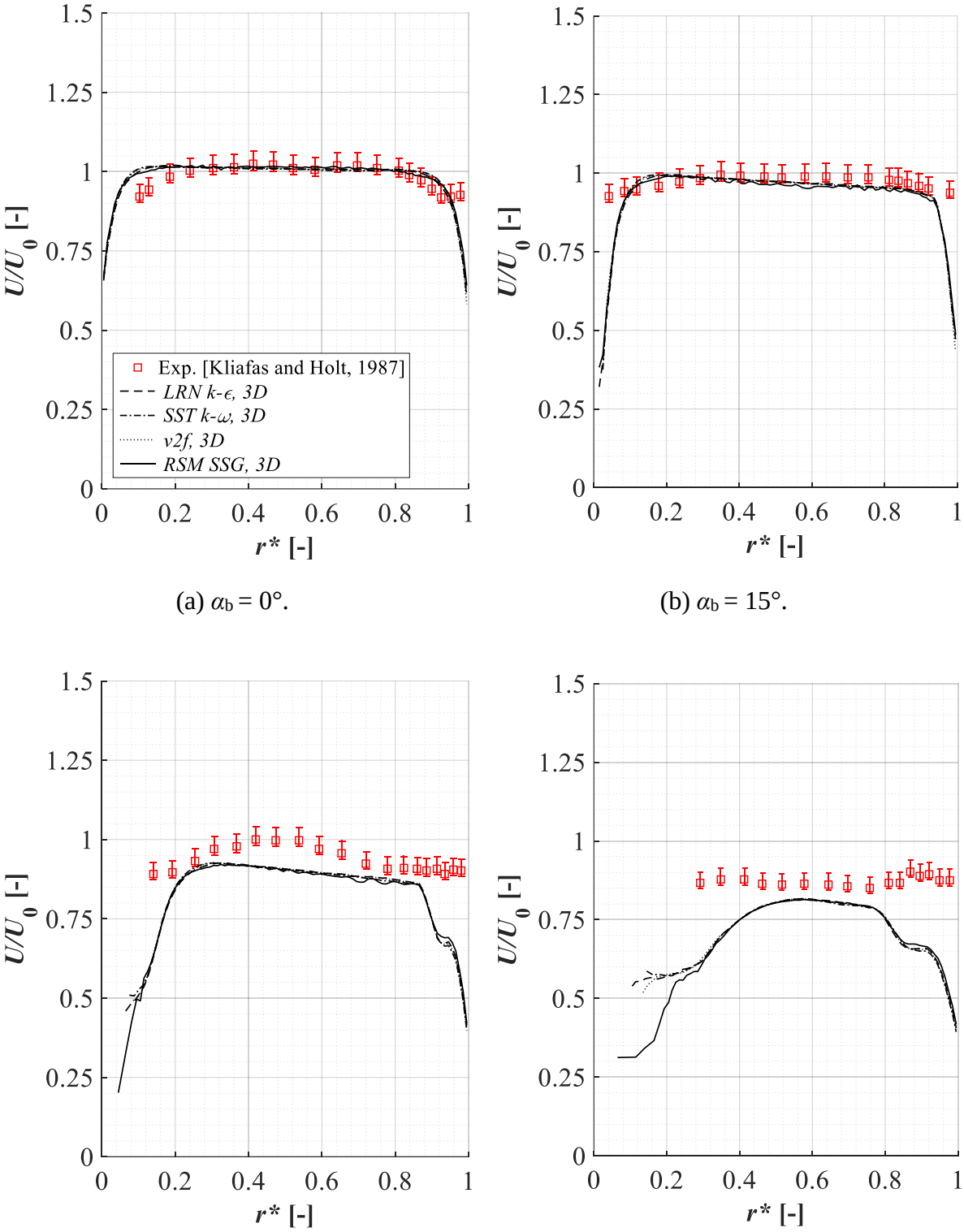


(c) $\alpha_b = 30°$.



(d) $\alpha_b = 45°$.

Figure 5.32 – Particulate-phase (3D case): mean streamwise velocity profiles along the elbow, using Brauer's PWI model (BRAUER, 1980).



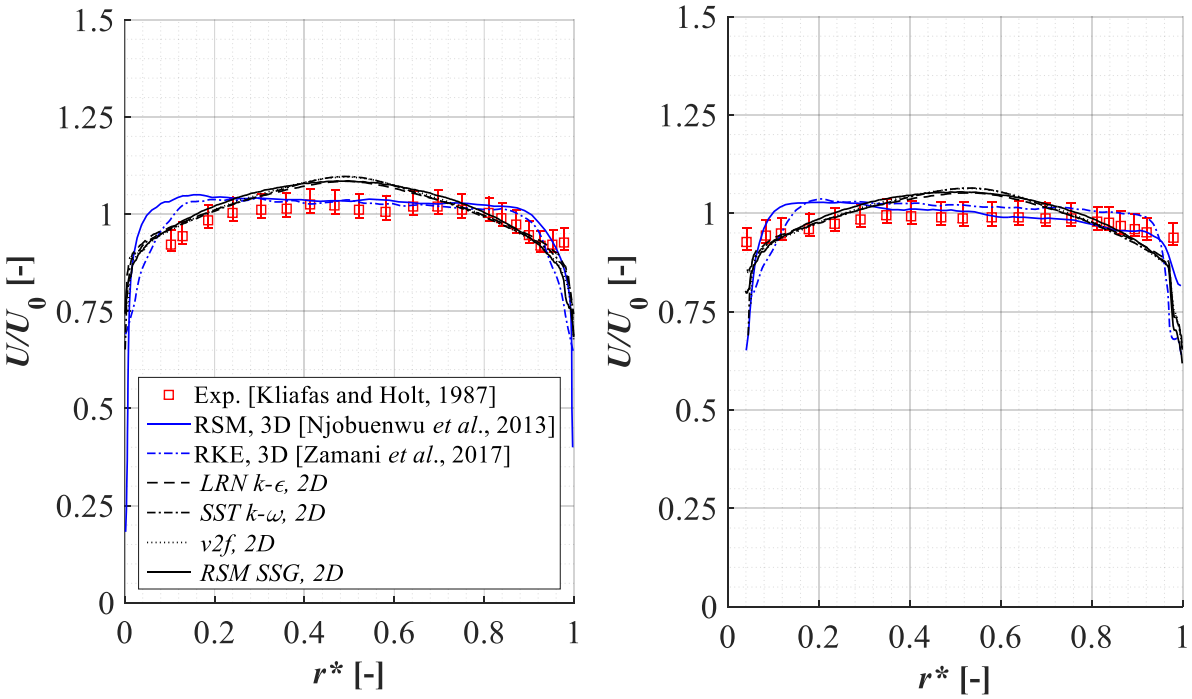
(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

(c) $\alpha_b = 30°$.

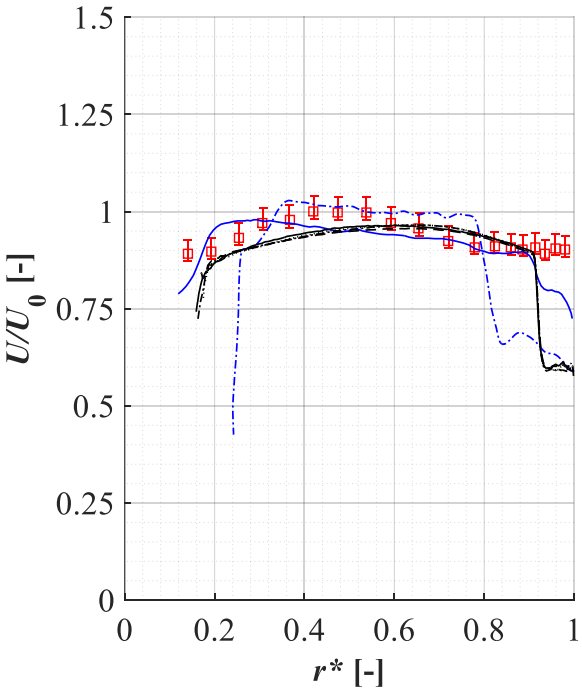
(d) $\alpha_b = 45°$.

Figure 5.33 – Particulate-phase (2D case): mean streamwise velocity profiles along the elbow, using Grant & Tabakoff's PWI model (GRANT and TABAKOFF, 1975).

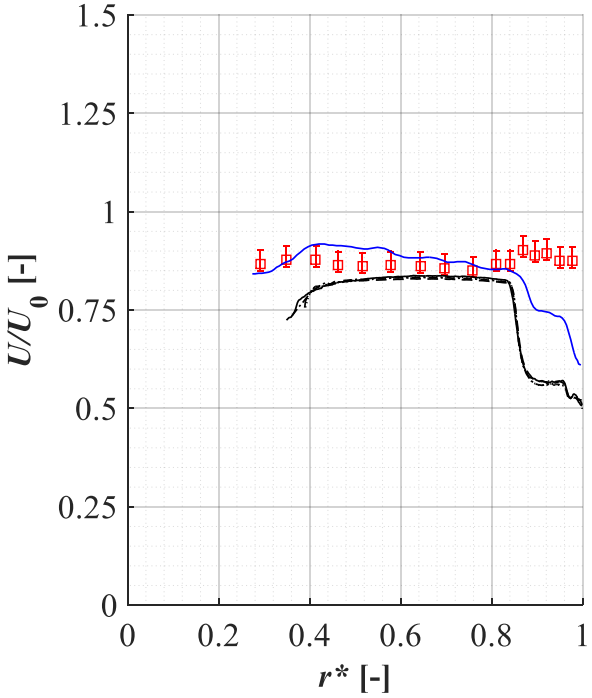


(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

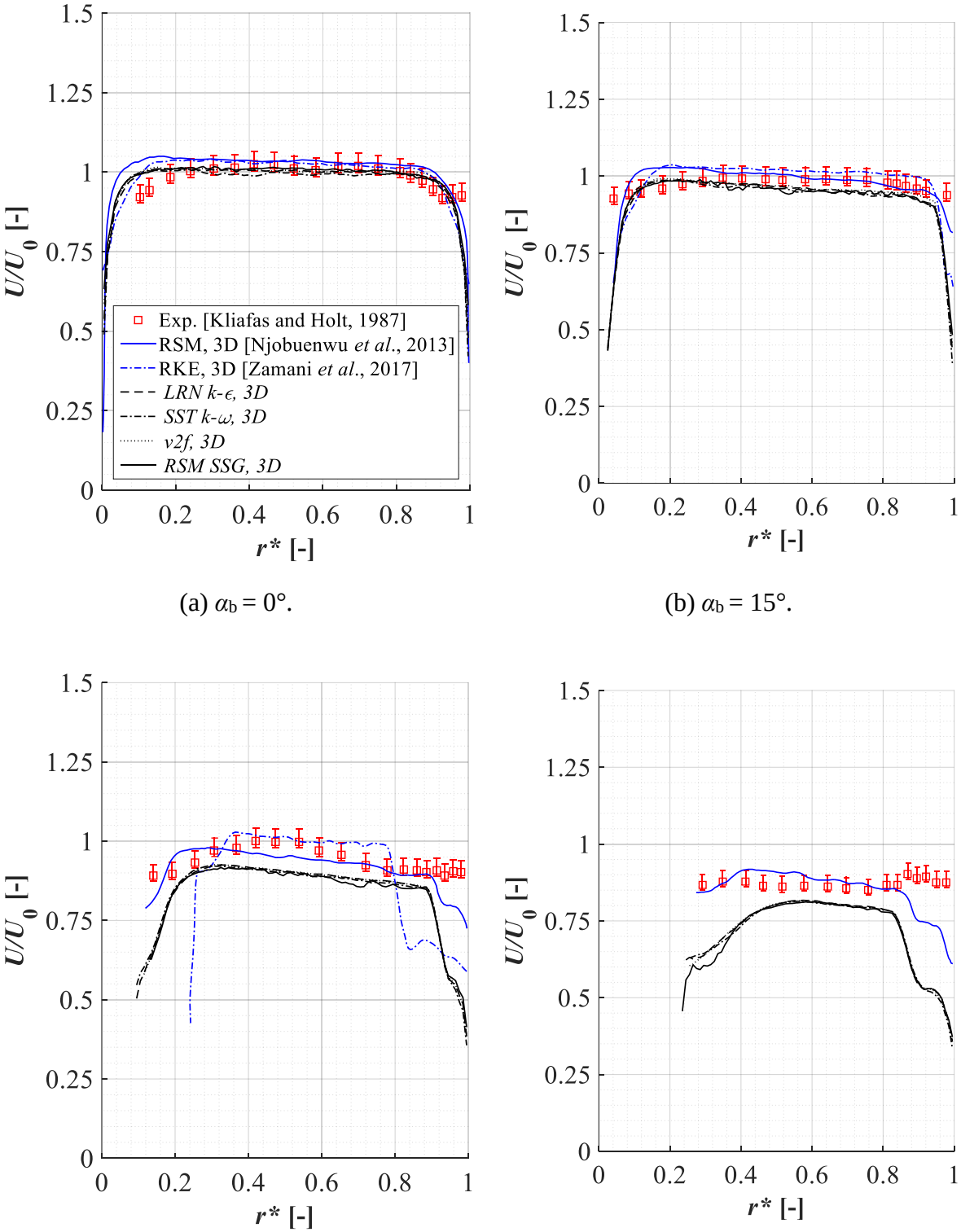


(c) $\alpha_b = 30°$.



(d) $\alpha_b = 45°$.

Figure 5.34 – Particulate-phase (3D case): mean streamwise velocity profiles along the elbow, using Grant & Tabakoff's PWI model (GRANT and TABAKOFF, 1975).



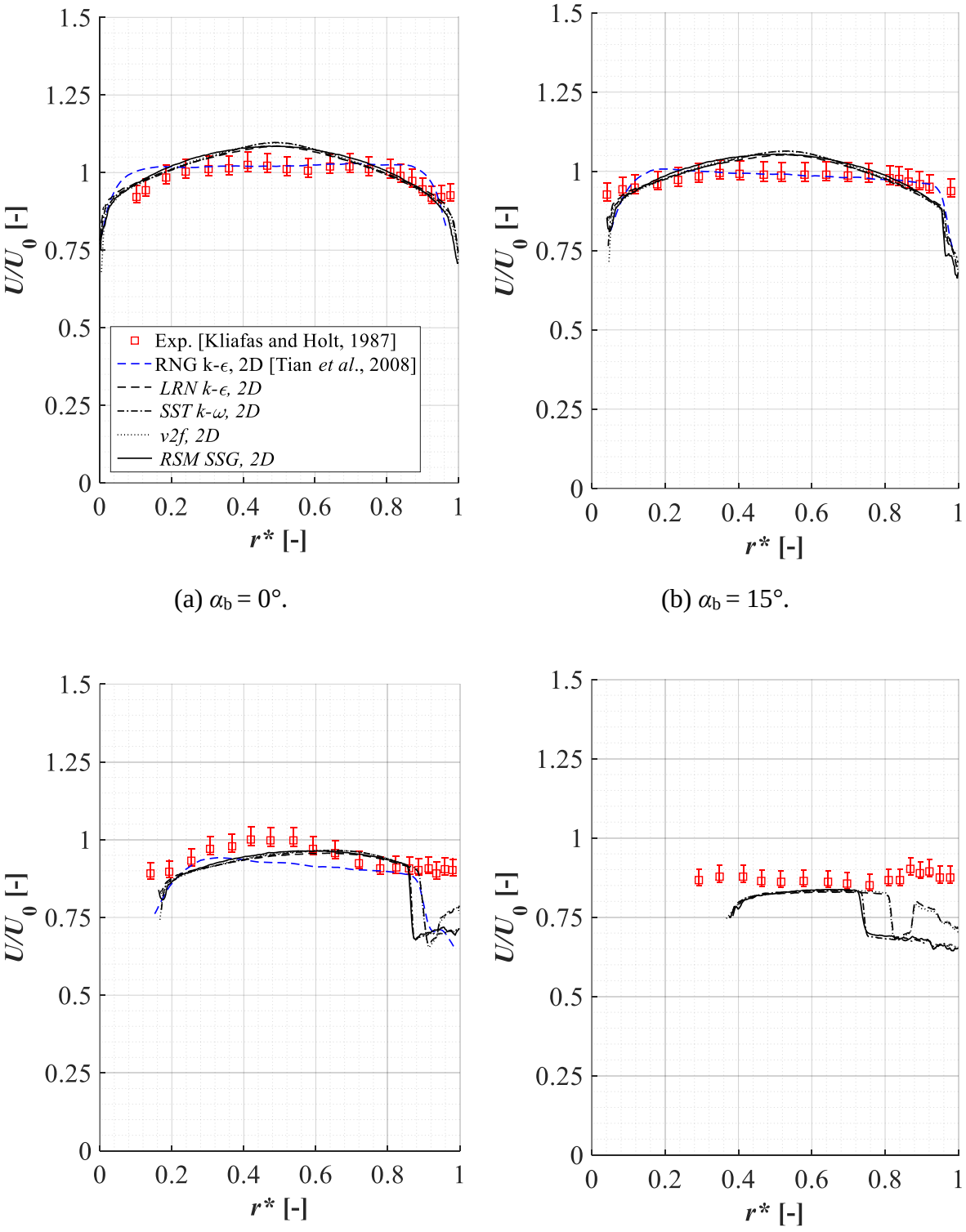
(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

(c) $\alpha_b = 30°$.

(d) $\alpha_b = 45°$.

Figure 5.35 – Particulate-phase (2D case): mean streamwise velocity profiles along the elbow, using Matsumoto & Saito's PWI model (MATSUMOTO and SAITO, 1970).



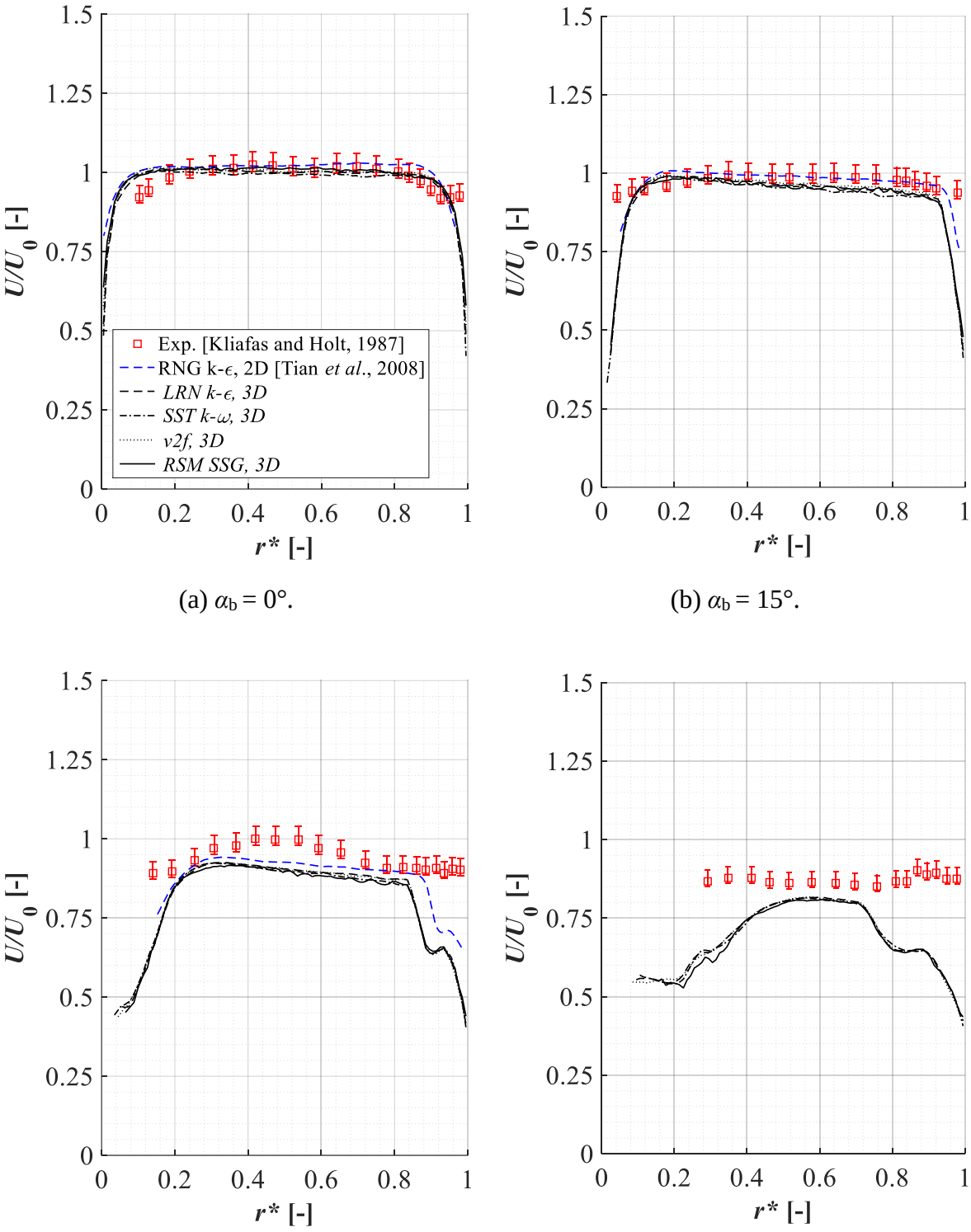
(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

(c) $\alpha_b = 30°$.

(d) $\alpha_b = 45°$.

Figure 5.36 – Particulate-phase (3D case): mean streamwise velocity profiles along the elbow, using Matsumoto & Saito's PWI model (MATSUMOTO and SAITO, 1970).



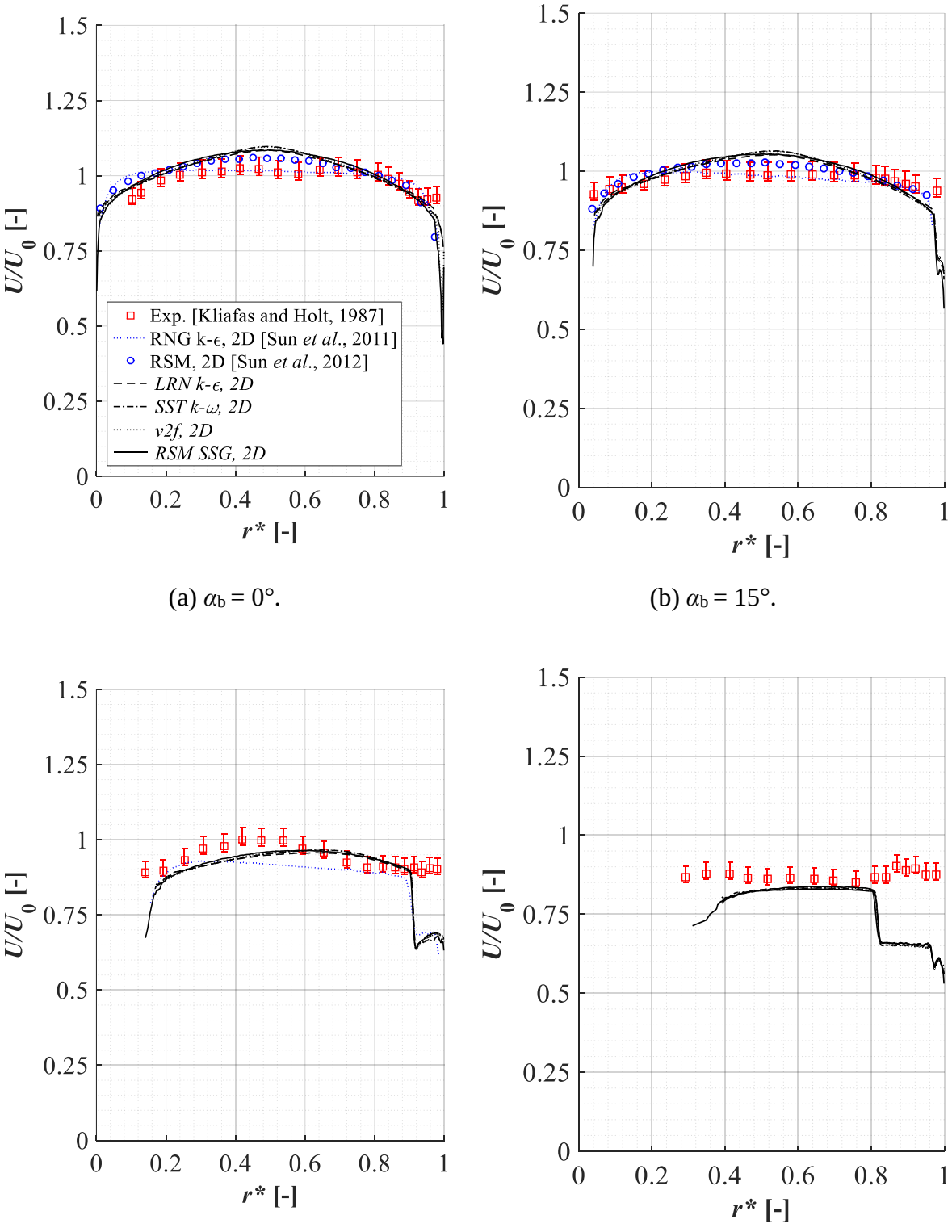
(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

(c) $\alpha_b = 30°$.

(d) $\alpha_b = 45°$.

Figure 5.37 – Particulate-phase (2D case): mean streamwise velocity profiles along the elbow, using Brach & Dunn's PWI model (BRACH and DUNN, 1998).



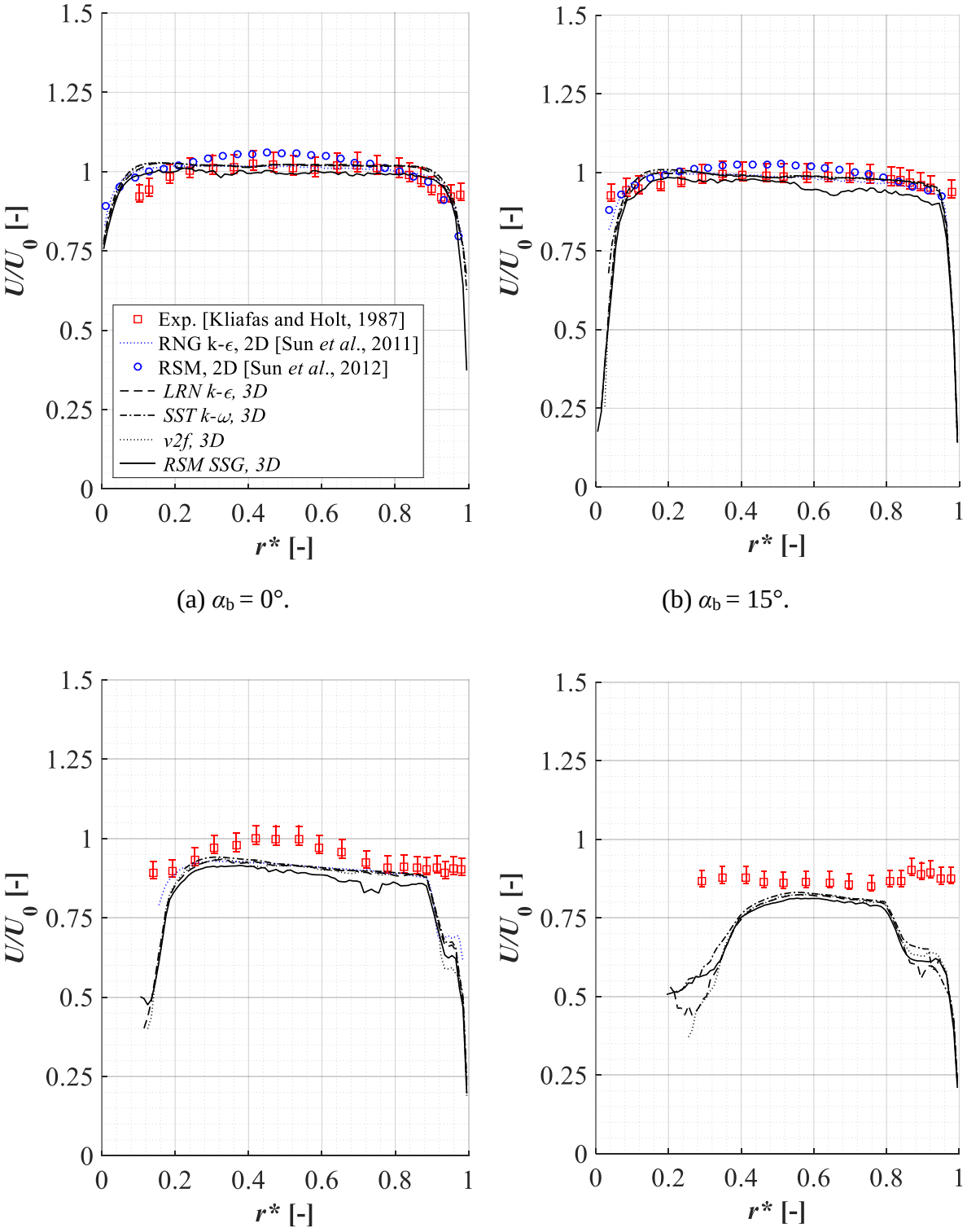
(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

(c) $\alpha_b = 30°$.

(d) $\alpha_b = 45°$.

Figure 5.38 – Particulate-phase (3D case): mean streamwise velocity profiles along the elbow, using Brach & Dunn's PWI model (BRACH and DUNN, 1998).



(a) $\alpha_b = 0°$.

(b) $\alpha_b = 15°$.

(c) $\alpha_b = 30°$.

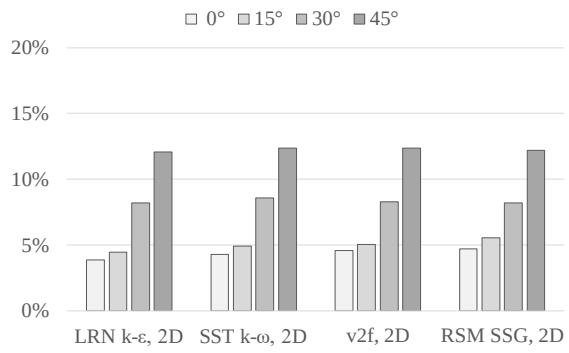
(d) $\alpha_b = 45°$.

For the 2D computations, at bend angles of 30° and 45°, the particle mean velocity profiles obtained with the SST k-ω and RSM SSG models at angles of 30° and 45° were different from those obtained with the LRN k-ε and the v^2f turbulence models near the outer bend wall. Nevertheless, for the 3D case of this study, all turbulence models combined with the Matsumoto & Saito PWI model followed a similar behavior within the bend. It can be observed in Figure 5.35 that the profiles from the 2D simulations with the LRN k-ε and the v^2f reduced the overestimation of the particulate-phase deceleration along the outer bend wall. Figure 5.35 also shows that the numerical results from TIAN *et al*. (2008) using the RNG k-ε turbulence model are in accordance with experimental data at the lowest bend angles (0° and 15°), while at 30° the discrepancies in the outer wall bend region were similar to those found with the LRN k-ε and the v^2f models for the 2D case.

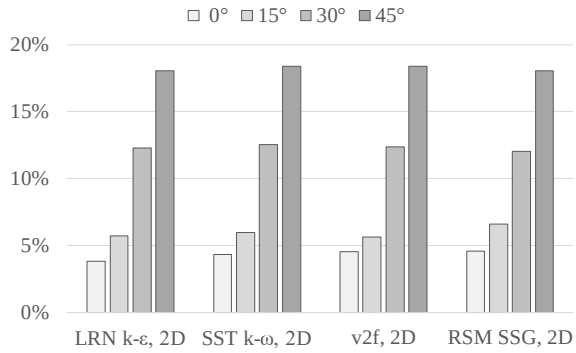
Concerning the Brach & Dunn's PWI model, the numerical mean stream-wise velocity profiles predicted by the four turbulence models tested were practically identical at the four bend angles for the 2D case as illustrated in Figure 5.37. The same observation can be done in this figure for the RSM turbulence model used by SUN *et al.* (2012) at bend angles up to 15°. Conversely, the 3D simulations with the RSM SSG (Figure 5.38) under-predict the particle stream-wise mean velocity through the elbow in comparison with the other turbulence models. As for the previous RANS-LPT/PWI combinations, the simulations using the Brach & Dunn's PWI model did not correctly predict the particulate-phase mean velocities at the outer bend wall. The deceleration of the particulate-phase in this region of the bend is again highly over-predicted at 30° and 45° for both 2D and 3D simulations of this study.

In order to quantify the discrepancies between the numerical and experimental results observed in Figure 5.31-Figure 5.38, the RMSE for the particulate-phase mean stream-wise velocity was calculated. As indicated in Figure 5.39 and Figure 5.40, independently of the turbulence model, the 2D and 3D simulations using Brauer, Grant & Tabakoff and Brach & Dunn's PWI models produced similar errors, i.e., the highest RMSE are about 12% for Brauer's with 2D model and 15% with the 3D model, 15% for Brach and Dunn's with the 2D model and 18% with the 3D model, and 18% for Grant & Tabakoff's with the 2D model and 19% with the 3D model. Regarding the 2D simulations using Matsumoto & Saito PWI model, the prediction errors were lower using the LRN k-ε and the v^2f turbulence models (with highest RMSE ~ 10%) than those using the SST k-ω and the RSM SSG, highest RMSE (with highest RMSE ~ 15%). On the contrary, the 3D simulations with Matsumoto & Saito PWI model were again independently of the turbulence model with the highest RMSE of 18%.

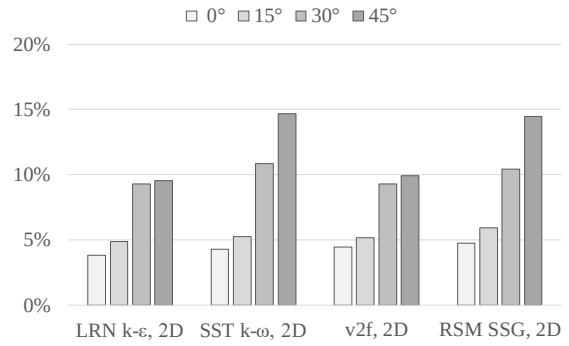
Figure 5.39 – Case 2D: RMSE for the particle-phase mean stream-wise velocity for the PWI models.



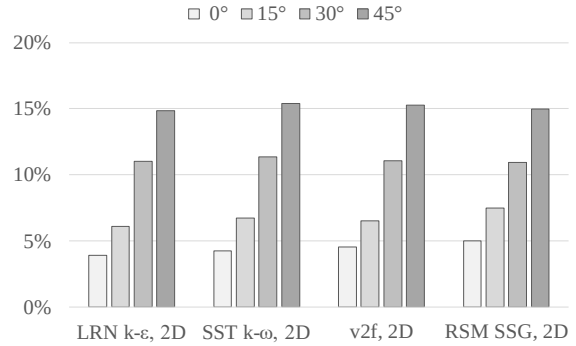
(a) Brauer's PWI model.



(b) Grant & Tabakoff's PWI model.

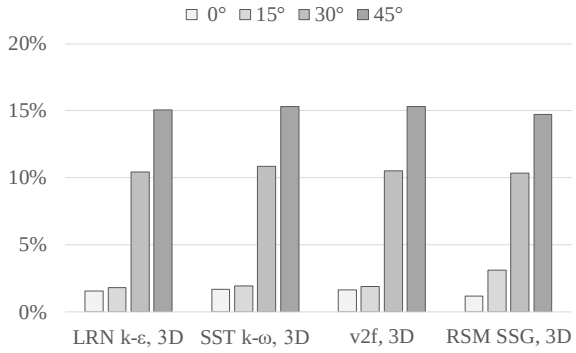


(c) Matsumoto & Saito's PWI model.

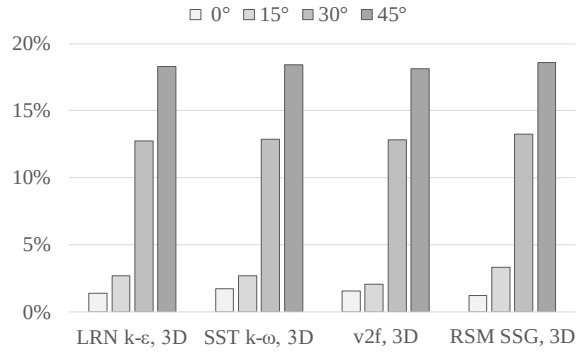


(d) Brach & Dunn's PWI model.

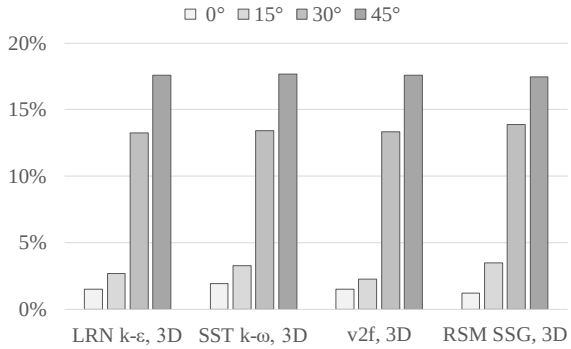
Figure 5.40 – Case 3D: RMSE for the particle-phase mean stream-wise velocity for the PWI models.



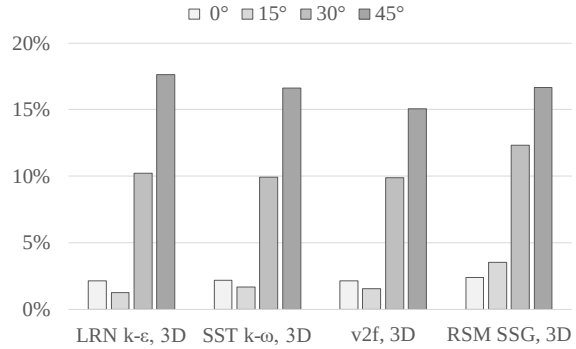
(a) Brauer's PWI model.



(b) Grant & Tabakoff's PWI model.



(c) Matsumoto & Saito's PWI model.



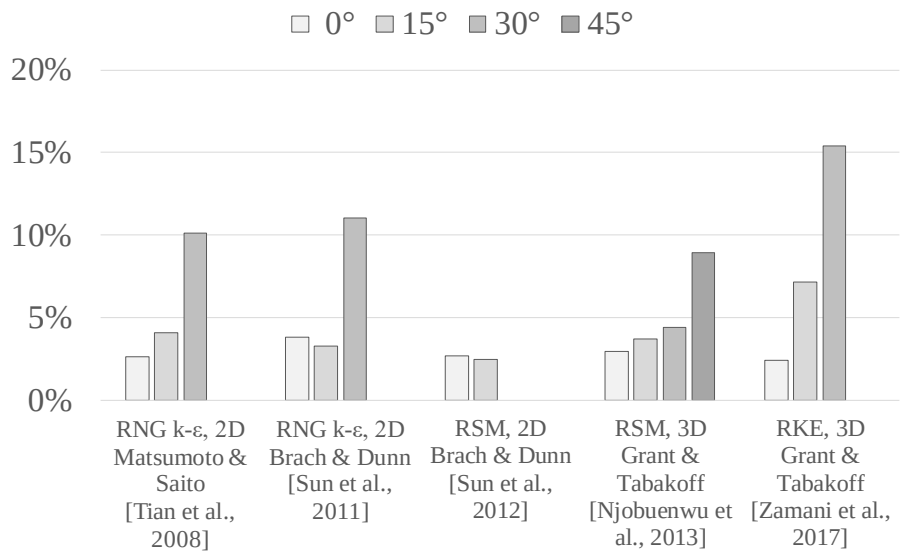
(d) Brach & Dunn's PWI model.

The RMSE at 45° were similar to those at 30° (~10%) for the LRN k-ε and the v^2f combined with Matsumoto & Saito PWI model with the 2D model (Figure 5.39c), while for all other RANS-LPT/PWI combinations the RMSE is much higher at 45° than at 30°. On the other hand, the same was not observed with the 3D simulations (Figure 5.40c). In addition, for all RANS-LPT/PWI combinations with the 3D simulations at bend angles of 0° and 15° the highest RMSE was 3%, whilst the highest RMSE was 7% for the 2D simulations at lowest bend angles (0° and 15°). For the highest bend angles (30° and 45°) we observed the opposite, the highest RMSE for all the RANS-LPT/PWI combinations with the 3D simulation was 19%, whilst for the 2D simulation was 18%.

Brach & Dunn's model gives results that are similar to Brauer's model. At a bend angle of 30°, the results are the same. In addition, the best agreement is obtained with the *v2f* model combined with Brach & Dunn's model. Brach & Dunn's model is similar to Matsumoto & Saito's model, but the restitution coefficient model in Brach & Dunn's model includes the effect of the adhesion force. It can also be noticed that Brauer's model, where the normal and tangential restitution coefficients are constant, gives results that are similar to those obtained with the Brach & Dunn's model. However, with Brauer's model, as explained in CROWE *et al.* (2012), the restitution coefficients depend on the contact point between the particle and the wall, this does not cause any problem for spherical particles, but it is problematic for non-spherical particles.

Comparatively to the other RANS-LPT/PWI combinations found in the current literature (Figure 5.41), the predictions from 3D simulations using the RSM turbulence model (NJOBUENWU *et al.*, 2013) combined with Grant & Tabakoff PWI model provided RMSE inferior to 10% at all bend angles, while RMSE reached 15% with the RKE (ZAMANI *et al.*, 2017) at bend angle of 30°. As shown in Figure 5.41, the predictions from 2D simulations using the RNG k-ε (TIAN *et al.* (2008) and SUN *et al.* (2011)) produced errors similar to the predictions using the LRN k-ε and the v^2f at the analyzed angles of the bend for the 2D case. The predictions at the lowest bend angles (0° and 15°) from 2D simulations using an RSM turbulence model (SUN *et al.*, 2012) produced errors slightly lower than those using the four turbulence models tested combined with the same PWI model.

Figure 5.41 – Literature: RMSE for the particle-phase mean streamwise velocity for the PWI models.



5.2.5 RANS-LPT/PWI combinations selection

Overall, the comparative analysis has shown that all RANS-LPT/PWI combinations had some difficulties to predict the behavior of the dilute gas-solid flow at the outer wall region of the bend, especially at the highest angles (30° and 45°).

The 2D simulations could predict the fluid flow with errors not higher than 13% and the predictions from 3D simulations produced errors lower than 8%. The radar chart of Figure 5.42 in which the center represents the "target" value (see section 5.1.4 for further information) indicates that the 3D case is better than the 2D case regarding the three criteria considered: the computing cost, the pressure drop, and the mean velocity.

However, the LRN k-ε and v^2f turbulence models when combined with Matsumoto & Saito PWI model reproduce the particulate-phase flow with errors of approximately 10% for the 2D case, while for the 3D case the errors were approximately 18%. Predictions from 3D simulations using an RSM turbulence model combined with Grant & Tabakoff PWI model (NJOBUENWU *et al.*, 2013) provide results with an error less than 10% for both phases. To summarize the RANS-LPT/PWI combinations with the qualitative analysis, Figure 5.43 is proposed (such as Figure 5.42 where the center represents the "target" value). The RANS-LPT/PWI combinations with Brauer's PWI model produced the lowest errors for both 2D and 3D simulations and the RANS-LPT/PWI combinations with the Matsumoto & Saito's PWI model for the 2D simulations.

Figure 5.42 – Fluid-phase: comparison between 2D and 3D cases

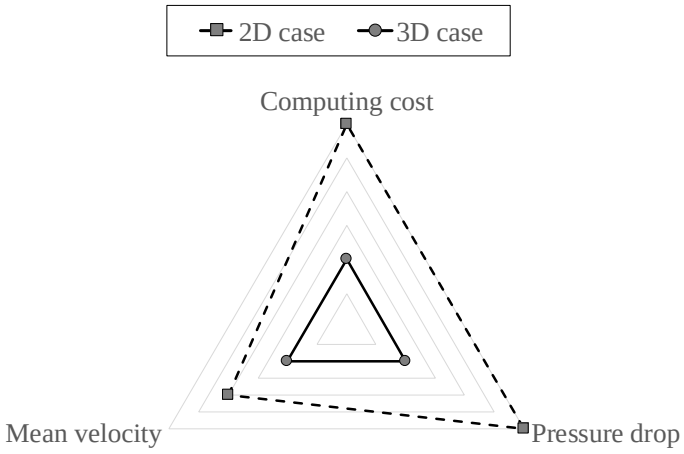
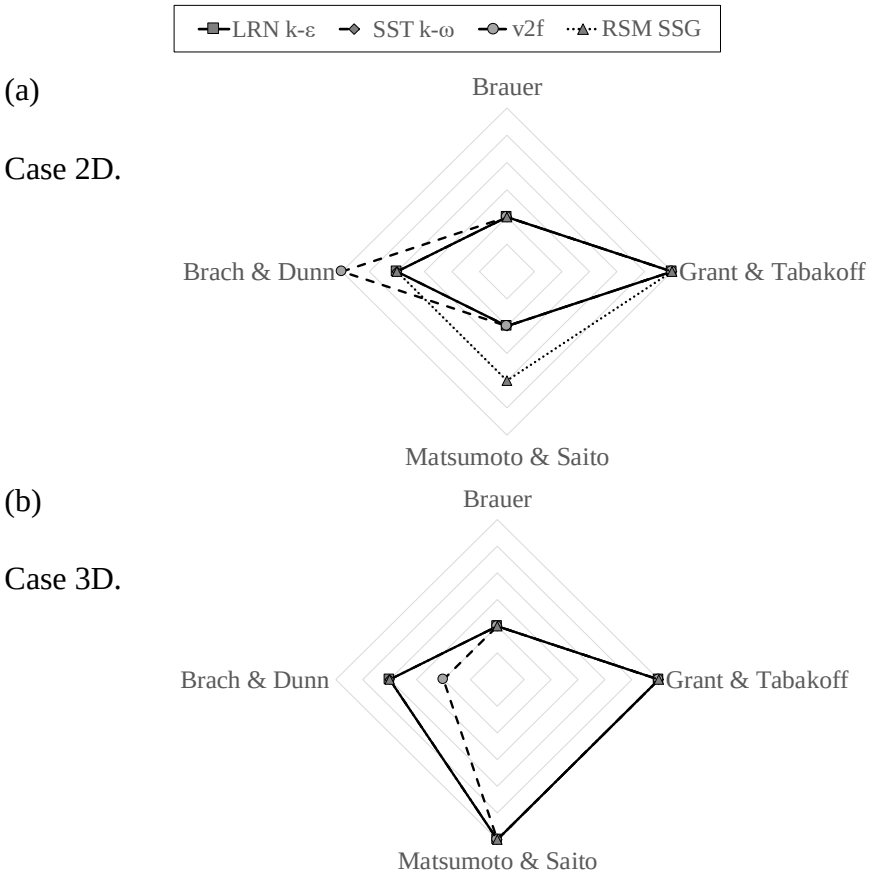


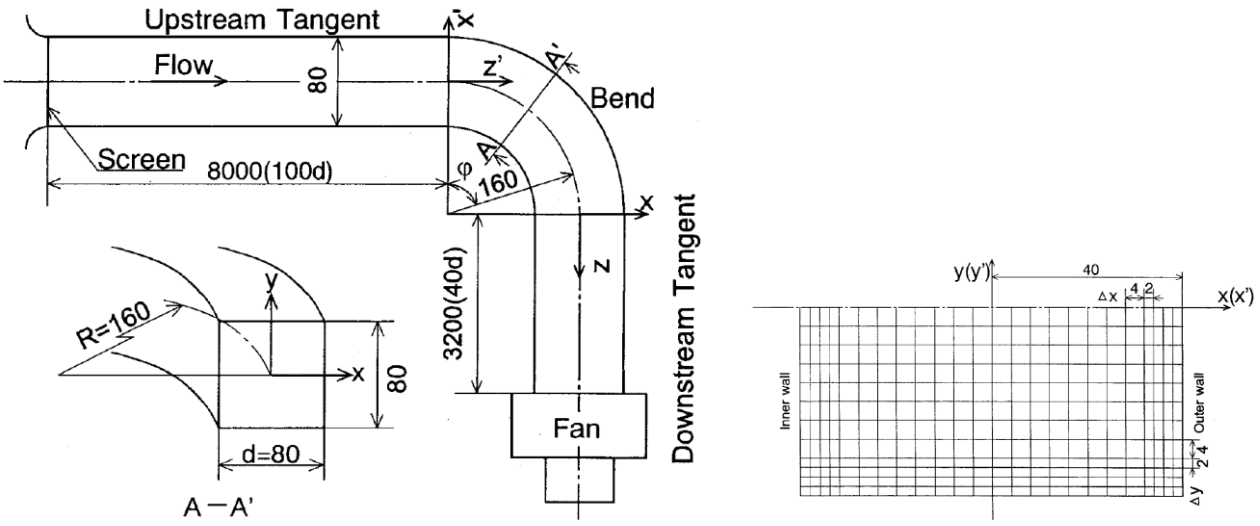
Figure 5.43 – Particulate-phase: comparison of RANS models for the (a) 2D and (b) 3D cases.



5.3 PARTICLE DEPOSITION IN CURVED DUCTS WITH VARIATION OF THE DUCT SHAPE

SUDO *et al.* (2001) performed experiments, schematically illustrated in Figure 5.44a, to measure (see the grid with measuring points in Figure 5.44b) principally the mean and fluctuating velocities in the turbulent airflow (without aerosol) through a horizontal to horizontal square-sectioned duct with a diameter of 0.08 m, an elbow with a curvature ratio of 4 and a Reynolds number of the flow equal to 4×10^4. This benchmark is similar to the work studied in section 5.1 (SUDO *et al.*, 1998), which had been chosen as a validation case for the fluid-phase because of the dimensions and flow conditions corresponding to those of HVAC applications.

Figure 5.44 – Schematic diagram of Sudo *et al.* (2001) experimental system (dimensions in [mm]).



(a) Test bend. (b) Measuring points.

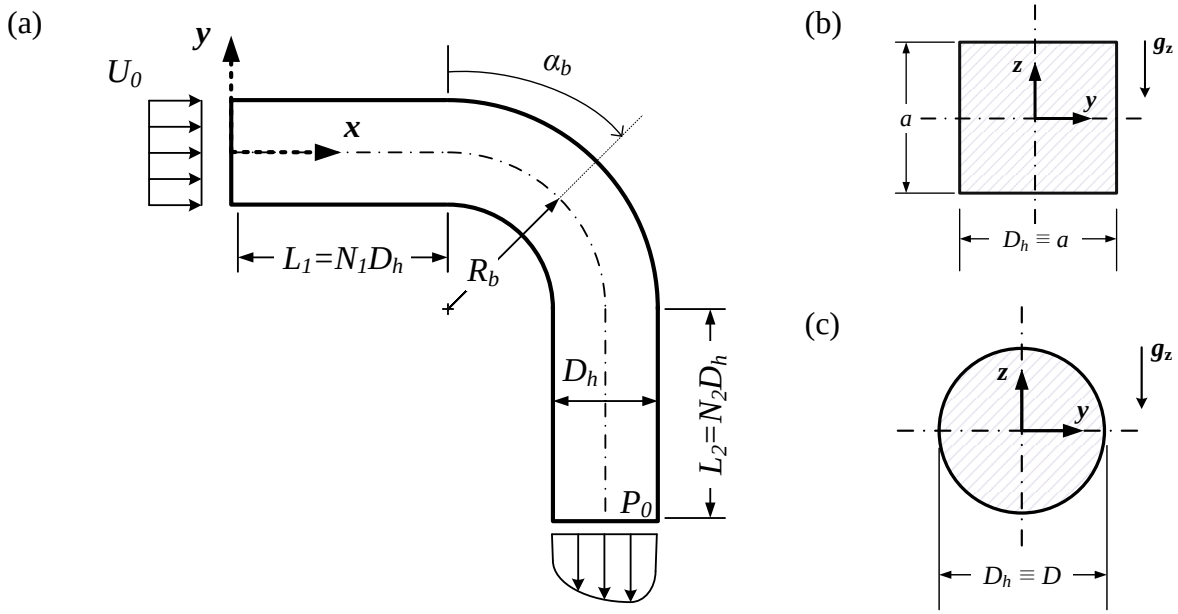
Source: SUDO *et al.* (2001).

The objective of this section is to evaluate the impact of the duct shape on the particle deposition in elbows. In order to reach this goal, the particle deposition rates in a geometry similar to the experiment of SUDO *et al.* (2001) are calculated and compared to the deposition rates in a circular-sectioned duct operating in the same conditions, i.e., the same Reynolds number and monodisperse aerosols. The PWI models are in this case reduced to a null coefficient of restitution, i.e., the particles are deposited once in contact with the duct walls.

5.3.1 Geometry and boundary conditions

The physical model is an elbow that connects a horizontal to horizontal straight duct as illustrated in Figure 5.45a (the gravity is in the *z*-direction, i.e., g_z). In this figure: D_h is the hydraulic diameter, R_b is the bend radius, α_b is the duct bend angle, L_1 and L_2 are, respectively, the length of the straight duct upstream and downstream the elbow. Figure 5.45b and Figure 5.45c present, respectively, the square and circular cross-sections.

Figure 5.45 – Geometry and boundary conditions for the duct bend model: (a) top view, (b) square and (c) circular cross-sections.



The boundary conditions for the gas-phase were the same as in the previous section (see section 5.1.1 for further information) with distinct airflow property values as provided in Table 5.7. The SST *k*-ω turbulence model was chosen in this study due to its good performance as shown in the previous studies in sections 5.1 and 5.2.

Table 5.7 – Elbow model: fluid and particulate phases.

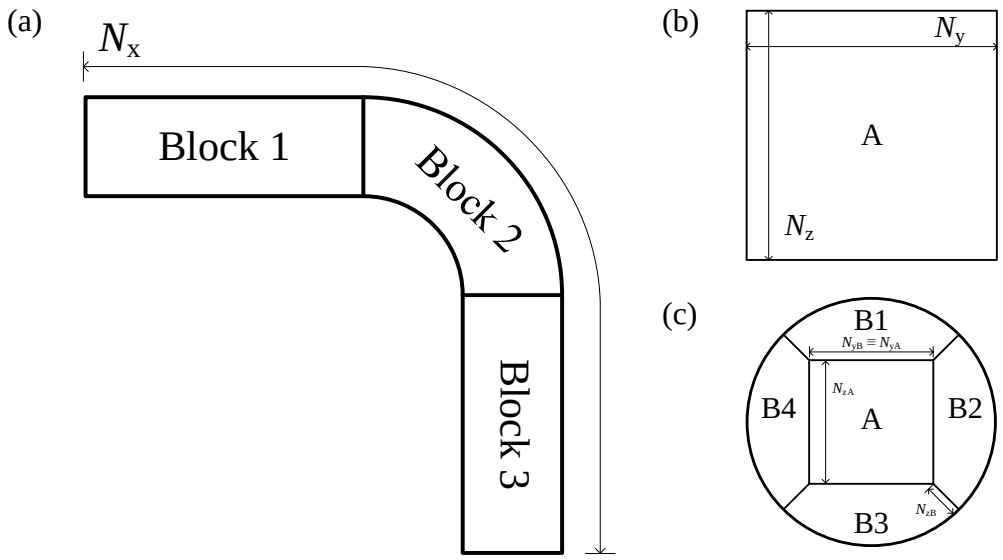
	Fluid-phase		Particulate-phase
Geometry	Air	Airflow	Particle
D = 80 mm	ρ = 1.2 kg/m^3	U_0 = 7.4 m/s	d_p = 1-100 μm
N_1 = 50	μ = 1.81×10^{-5} kg/ms	u_τ = 0.39 m/s	ρ_p = 2500 kg/m^3
N_2 = 50	ν = 1.51×10^{-5} m^2/s	f = 0.0220	Kn = 0.1-0.001
R_b = 160 mm		Δp = 82 Pa	τ_p = 0.01-77 ms
δ = 4		Re = 4×10^4	St = 0.002-14
α_b = 90°		De = 2×10^4	N_p = 10^5

The properties of the particulate-phase are also presented in Table 5.7. The simulations were performed for a particle size ranging from 0.1 μm to 100 μm. One cloud with 10^5 monodisperse particles of 2500 kg/m^3 density was injected at once with the same local fluid velocity in the upstream duct near the bend inlet ($x = 40D_h$) and the particles were tracked during 0.35 s with a time step of 10^{-6} s. The one-way coupled Lagrangian formulation was applied and the particle-fluid interactions considered were the drag, the shear-lift, the gravity and the Brownian forces. APPENDIX C provides a typical run of the LPT OpenFOAM® solver.

5.3.2 Mesh

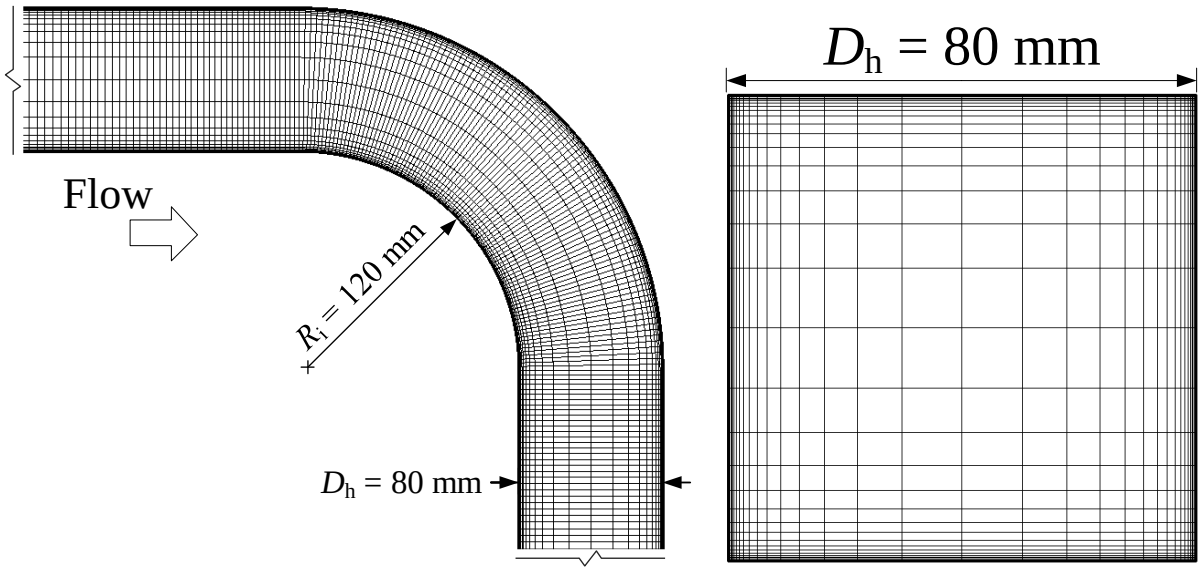
The mesh was generated with the *blockMesh* tool of the OpenFOAM® code (see APPENDIX A). The model was divided into three parts as presented in Figure 5.46a: Block 1 (the upstream duct), Block 2 (the elbow) and Block 3 (the downstream duct). The square-sectioned duct needs only the information of the section A from Figure 5.46b to generate the mesh. On the other hand, the circular-sectioned duct must have five additional parts to follow the O-grid method for the mesh generation (TU *et al.*, 2013) as shown in Figure 5.46c. Therefore, we have a total of 15 blocks (instead of the 3 blocks of the square-sectioned grid) to generate this mesh with the divisions indicated as N_x for the stream-wise direction and N_{yA}, N_{yB}, N_{zA}, and N_{zB} for the wall-normal directions.

Figure 5.46 – Block meshing strategy.



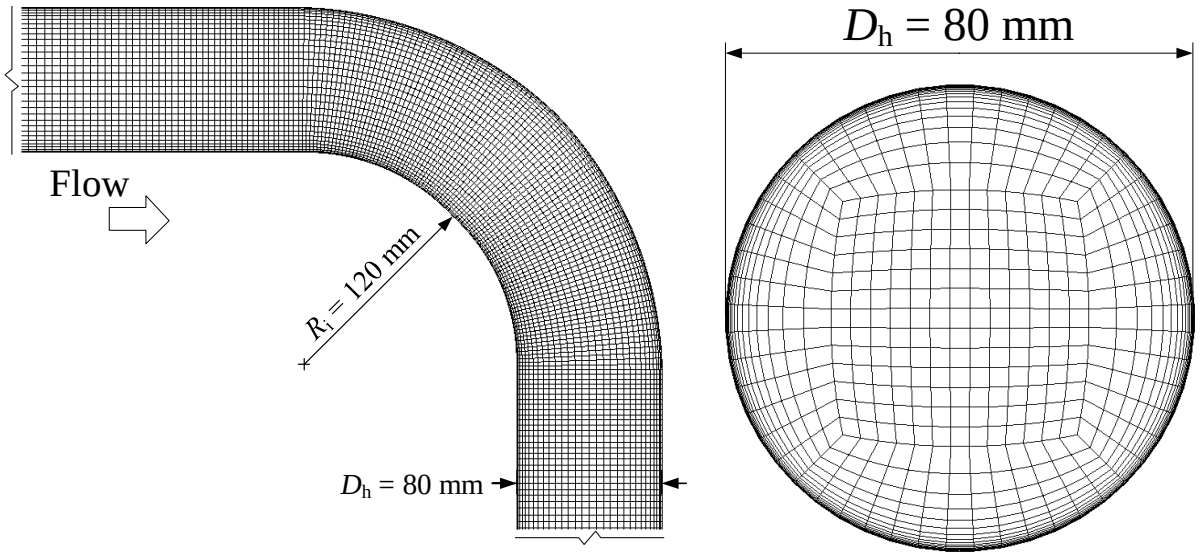
The meshes generated for the duct with square and circular cross-sections are shown respectively in Figure 5.47 and Figure 5.48 (the total length of the upstream and downstream ducts meshes were hidden in both cases). The wall-normal expansion ratio was approximately 1.2 and the cell length is uniform along the stream-wise direction.

Figure 5.47 – Square duct mesh: Grid 2.



(a) Top view. (b) Cross-section.

Figure 5.48 – Circular duct mesh: Grid 2.



(a) Top view. (b) Cross-section.

5.3.3 Fluid-phase: V&V

The verification and validation (V&V) of this study were split into the fluid-phase and the particulate-phase. This section handles the V&V of the fluid-phase following the systematic approach proposed in section 5.1.3, concerning:

- the grid convergence study (Table 5.8);
- the pressure drop, friction factor and computing cost (Table 5.8);
- the fully developed turbulent flow profile (Figure 5.50 and Figure 5.52a);
- the stream-wise mean velocity profile at the bend outlet (Figure 5.51 and Figure 5.52b);
- the mean velocity contours (Figure 5.53, Figure 5.56 and Figure 5.57);
- the turbulence intensities (Figure 5.54).

The grid convergence study of the turbulent airflow was conducted with the SST *k-ω* turbulence model and the selected grids are listed in Table 5.8 in which the refinement follows the GCI method of ROACHE (1994) with a refinement expansion ratio (r_m) near 1.2.

Table 5.8 – Grid convergence study.

Cross-section	Grid [a]		y^+_{max}	θ_{max} [b] [°]	Δp [c] [Pa] (E_r [d])	f [e] (E_r [d])	N° of iterations [f]	t [g] [h]
Square	*Grid 1*	480×30×30	0.62	1.82	66.30 (-19%)	0.0125 (-43%)	3×10^4	8
	Grid 2	576×36×36	0.55	1.59	68.92 (-16%)	0.0131 (-40%)	4×10^4	25
	Grid 3	692×44×44	0.47	1.33	71.52 (-13%)	0.0136 (-38%)	4×10^4	30
Circular	*Grid 1*	480×[13×13 +4(13×16)]	0.98	21.92	66.86 (-19%)	0.0129 (-41%)	3×10^4	11
	Grid 2	576×[16×16 +4(16×20)]	0.80	23.95	70.10 (-15%)	0.0135 (-39%)	4×10^4	27
	Grid 3	692×[20×20 +4(20×24)]	0.68	25.64	70.77 (-14%)	0.0137 (-38%)	5×10^4	33

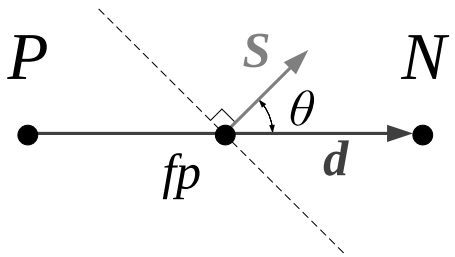
[a] The grid for the square duct is $N_x\times N_y$ and for the circular duct is $N_x\times[N_{yA}\times N_{zA}+4(N_{yB}\times N_{zB})]$.
[b] Maximum grid non-orthogonality angle.
[c] The calculated pressure drop with Equation (2.10) is 82.23 Pa.
[d] Relative error calculated with Equation (5.11).
[e] The calculated Darcy friction factor with Equation (2.11) is 0.0220.
[f] Number of iterations to achieve the convergence.
[g] Total computing time with parallel computation (2 Deca Core Intel Xeon CPU E5-2660 v3s).

The maximum value of y^+ along the duct walls are calculated from the converged numerical solution for each grid and the values were almost the same for all grids and adequate for discretizing the viscous sublayer. The average pressure drop (Δp) and the Darcy-Weisbach friction factor (f) were also calculated. There is no significant variation regarding these properties for both square and circular cross-sections. Moreover, these flow properties have converged for a given result by reducing the relative errors, confirming the convergence for an accurate numerical solution with the grid refinement. The computing effort was similar for both square and circular cross-sections and it increases with the mesh refinement, as expected.

Additionally, a grid quality property is introduced in Table 5.8, the maximum grid non-orthogonality angle (θ_{max}). The non-orthogonality angle (θ) is illustrated in Figure 5.49, i.e., the angle at the face *fp* between the normal face vector $\vec{S}$ and the vector $\vec{d}$ that connected the center of the control volume center *P* and *N*, defined as follows:

$$\theta = \cos^{-1}\left(\frac{\vec{d}\cdot\vec{S}}{\|\vec{d}\|\|\vec{S}\|}\right). \tag{5.14}$$

Figure 5.49 – Non-orthogonality angle.



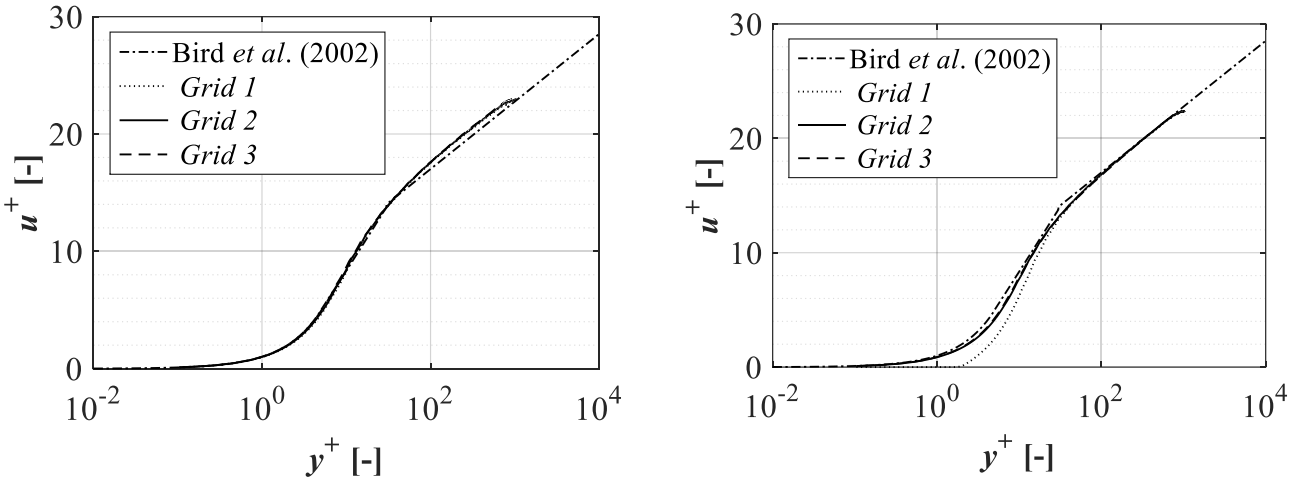
According to GREENSHIELDS (2018) for a non-orthogonality above 80° the convergence is generally hard to achieve. The maximum grid non-orthogonality of the square-sectioned duct is only 2° while that of the circular-section duct is up to 26°. It is expected that the square-sectioned duct model must provide a better numerical convergence than the circular-sectioned duct model, with no significant convergence issue for both cross-sections.

Figure 5.50 compares the dimensionless velocity u^+ at x/D_h = -1 from the bottom wall to the duct centerline, represented with the non-dimensional normal distance y^+, with that for a fully developed turbulent flow. All meshes for both model predictions were fairly close to the experimental data fit from BIRD *et al.* (2002).

The mean stream-wise velocity at the bend outlet normalized with the bulk velocity (U_0 = 7.4 m/s) is shown in Figure 5.51. Only the results at bend outlet are presented because

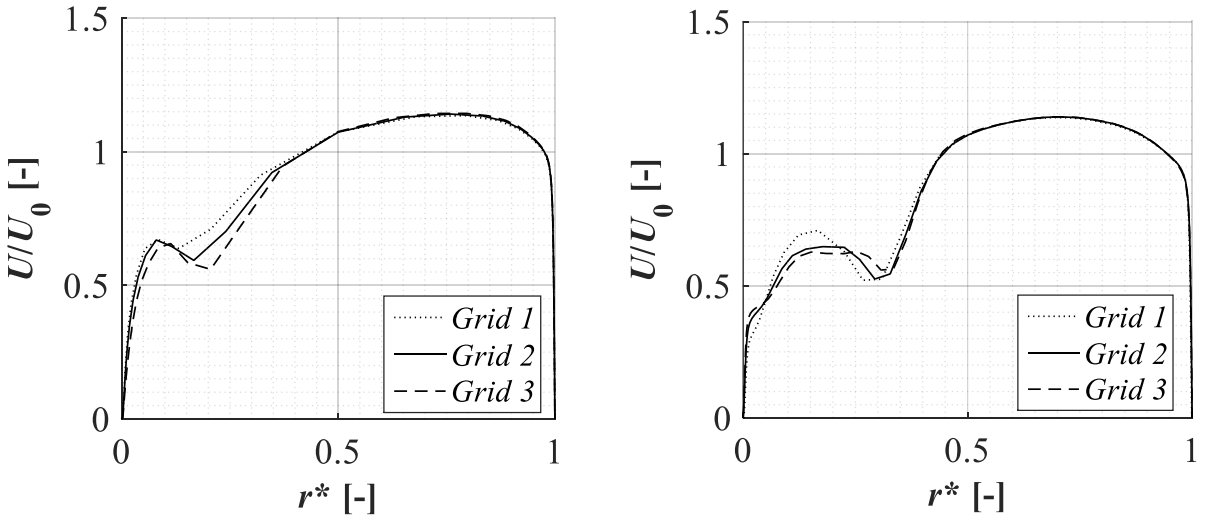
this region may have the highest distortion of the velocity profile and if the refinement solution converged in this bend angle, consequently, the other bend angles will also yield a refinement solution such as proved in section 5.1 (e.g., Figure 5.6). For both square-sectioned (Figure 5.51a) and circular-sectioned (Figure 5.51b) ducts, the coarse (*Grid 1*) and intermediate (*Grid 2*) meshes converged towards the fine grid numerical solution (*Grid 3*). No variation was observed for the outer bend wall region ($r^* > 0.5$) independently of the grid for both models. In addition, no significant variation was noted near the inner bend wall region ($r^* < 0.4$) for both models. Therefore, the *Grid 3* was selected for further analysis.

Figure 5.50 – Fluid-phase: dimensionless mean velocity profile along the duct radius at $x/D_h = 45$.



(a) Square duct. (b) Circular duct.

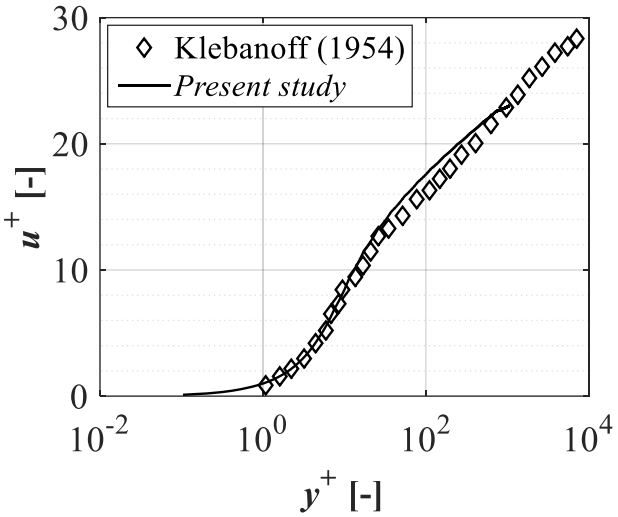
Figure 5.51 – Fluid-phase: mean streamwise velocity at the bend outlet ($\alpha_b = 90°$).



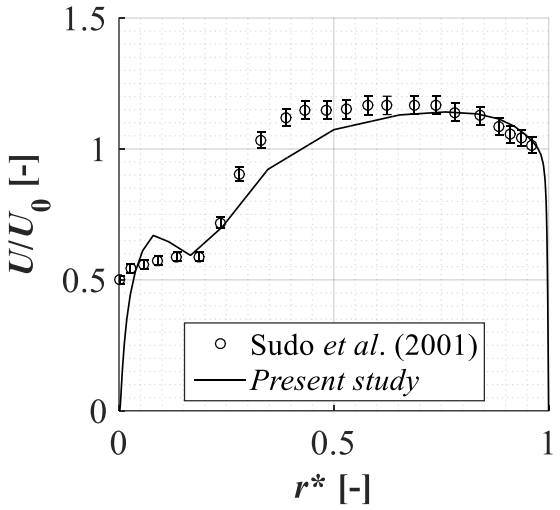
(a) Square duct. (b) Circular duct.

Once the selected benchmark for validation was a square-sectioned duct and the duct geometry and airflow conditions were based on this experimental data the validation of the fluid-phase was analyzed only for the square duct. The square duct model predicted fairly close the fully developed turbulent flow profile from KLEBANOFF (1954) as indicated in Figure 5.52a. This model also predicts accurately the mean stream-wise velocity profile of SUDO *et al.* (2001) at the bend outlet as observed in Figure 5.52b (corresponding RMSE of 8%).

Figure 5.52 – Fluid-phase (square duct): (a) fully developed turbulent profile at the upstream duct close to the bend inlet and (b) mean stream-wise velocity at the bend outlet.



(a) x/D_h = 45



(b) α_b = 90°.

A qualitative validation of the secondary motion is proposed in Figure 5.53 for the square-sectioned model against the experimental data of SUDO *et al.* (2001) at bend inlet and bend outlet. The calculated isocurve $U/U_0 = 1.21$ at the bend inlet (Figure 5.53a) shifting towards the inner bend wall as in the experimental data and the vectors indicated the start of the secondary motion likewise to the experimental result. Regarding the bend outlet (Figure 5.53b), the isocurve of $U/U_0 = 1.12$ was shifted towards the outer bend wall and the distortion of the velocity profile is higher than at the bend inlet such as in the experimental data. Additionally, the center location of the Dean vortices at the bend outlet is almost the same as those from the experimental data (Figure 5.53b).

Nevertheless, the turbulence intensities predicted with the SST *k-ω* were weaker than those of SUDO *et al.* (2001) as illustrated in Figure 5.54 at the bend inlet and outlet. Still, the square-sectioned model reproduced the flow behavior in bend inlet regarding the turbulence intensities.

Figure 5.53 – Fluid-phase: contours of the dimensionless mean stream-wise velocity (U/U_0) at (a) the bend inlet and (b) the bend outlet.

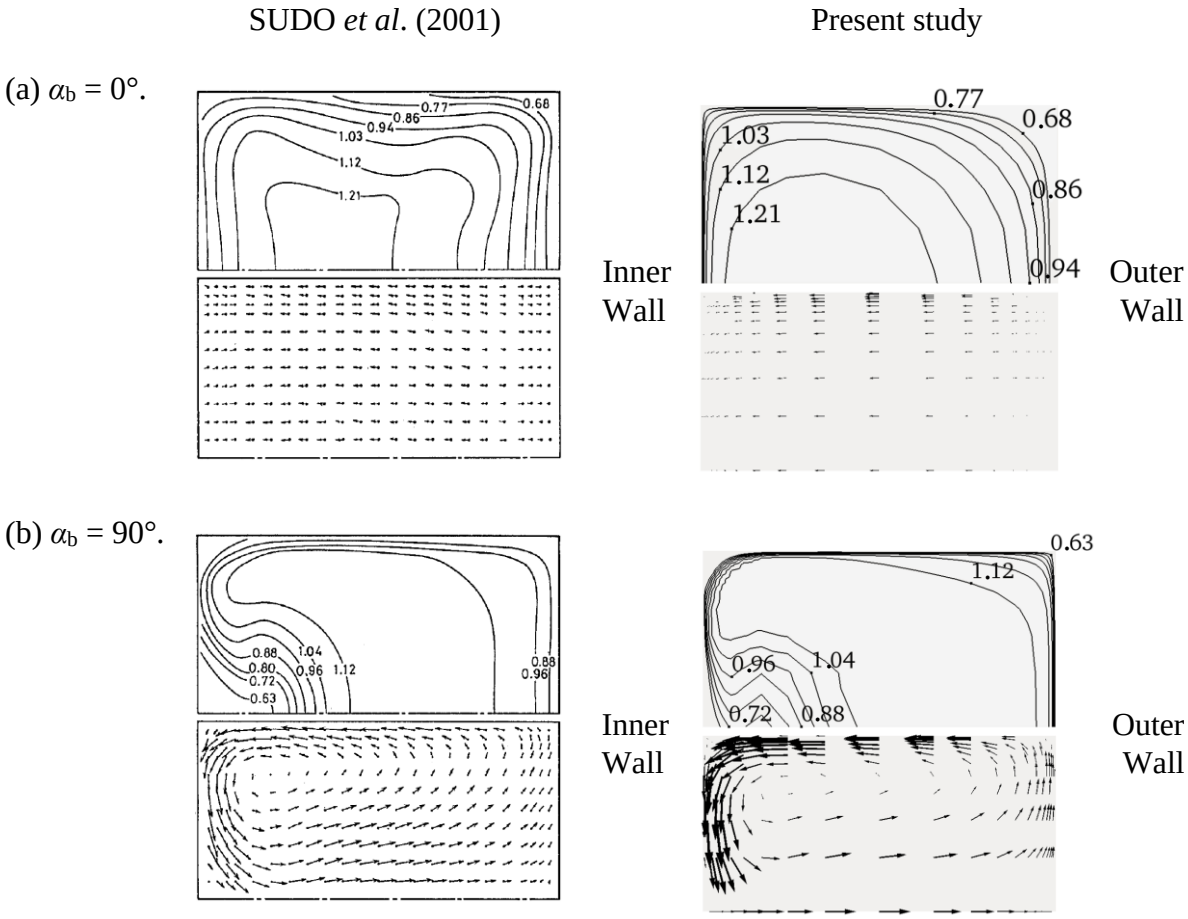
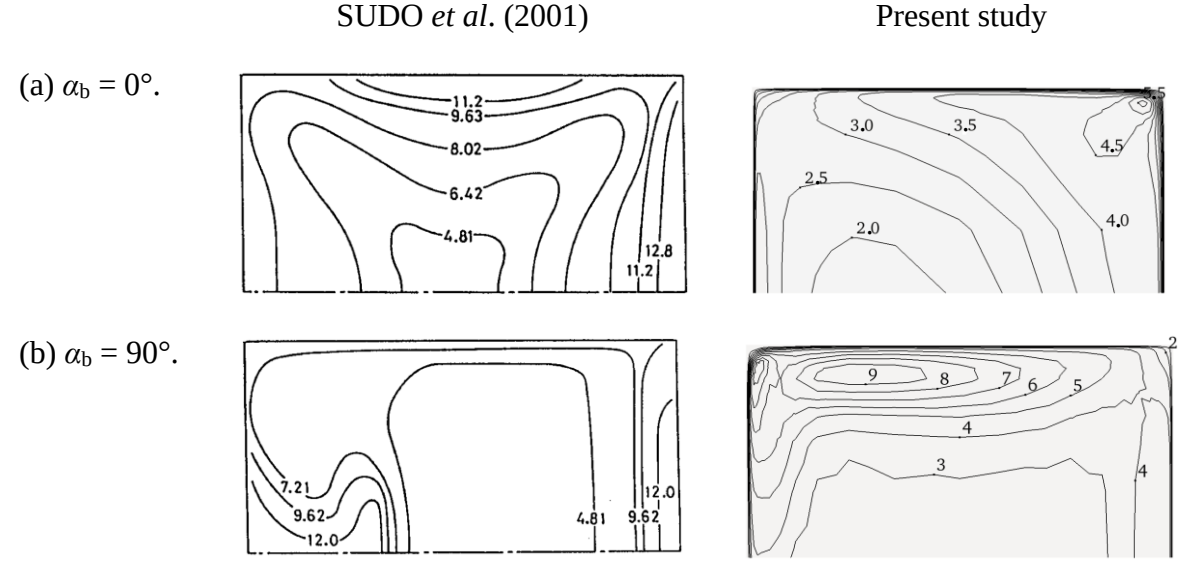
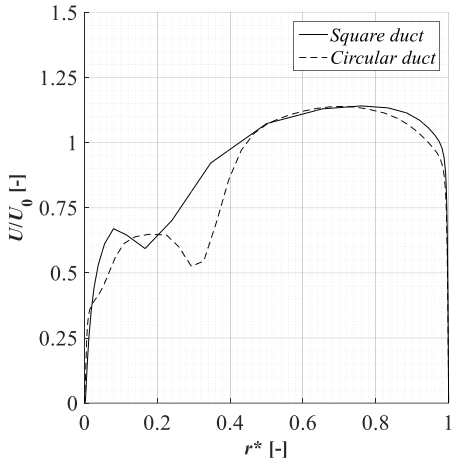


Figure 5.54 – Fluid-phase: contours of the dimensionless turbulence intensity at the principal flow direction at (a) the bend inlet ($\sqrt{u'^2}/U_0 \times 10^2$) and (b) the bend outlet ($\sqrt{v'^2}/U_0 \times 10^2$).



We can propose some comparisons between the turbulent airflow within the square-sectioned and circular-sectioned duct bends. The normalized mean stream-wise velocity for both models at the bend outlet is presented in Figure 5.55. Both models reached almost the same maximum velocity of $U/U_0 \approx 1.14$ at $r^* \approx 0.7$. Near the bend outer wall, the circular duct produced lower acceleration than the square duct. The deceleration near the bend inner wall was higher for the circular duct than for the square duct.

Figure 5.55 – Fluid-phase: comparison of the mean stream-wise velocity at the bend outlet ($\alpha_b = 90°$).



The contours of mean stream-wise velocity at the bend inlet and outlet for the square and circular ducts are shown in Figure 5.56. As presumed, in both cross-sections the maximum mean velocity was offset towards the inner bend wall at the bend inlet and offset towards the outer bend wall at the bend outlet deforming the turbulent velocity profile. A stagnation region appeared along the outer bend wall in the corners of the square duct at the bend inlet, while at the bend outlet the stagnation regions appeared at the corners along the inner bend wall. Due to the round geometry of the circular duct, there is no stagnation region as observed for the square duct excepted the separation region at the bend outlet along the inner bend wall that is larger than that of the square duct.

Figure 5.57 and Figure 5.58 present, respectively, the mean stream-wise velocity and the turbulent kinetic energy for both square and circular ducts for the plane $z = 0$ (the middle plane) and the planes at the upstream duct (at $x/D_h = 45$ and $x/D_h = 47.5$), the elbow (at bend angles 0°, 45° and 90°) and the downstream duct (at $y = -[R_b+2.5D_h]$ and $y = -[R_b+5D_h]$). Regarding the mean velocity, the separation region is larger in the circular duct (Figure 5.57a) than in the square duct (Figure 5.57b). The turbulent kinetic energy is higher at $z = 0$ in the downstream duct for the circular duct (Figure 5.58b) than for the square duct (Figure 5.58a).

Figure 5.56 – Fluid-phase: comparison of the mean stream-wise velocity contours at bend (a) inlet and (b) outlet.

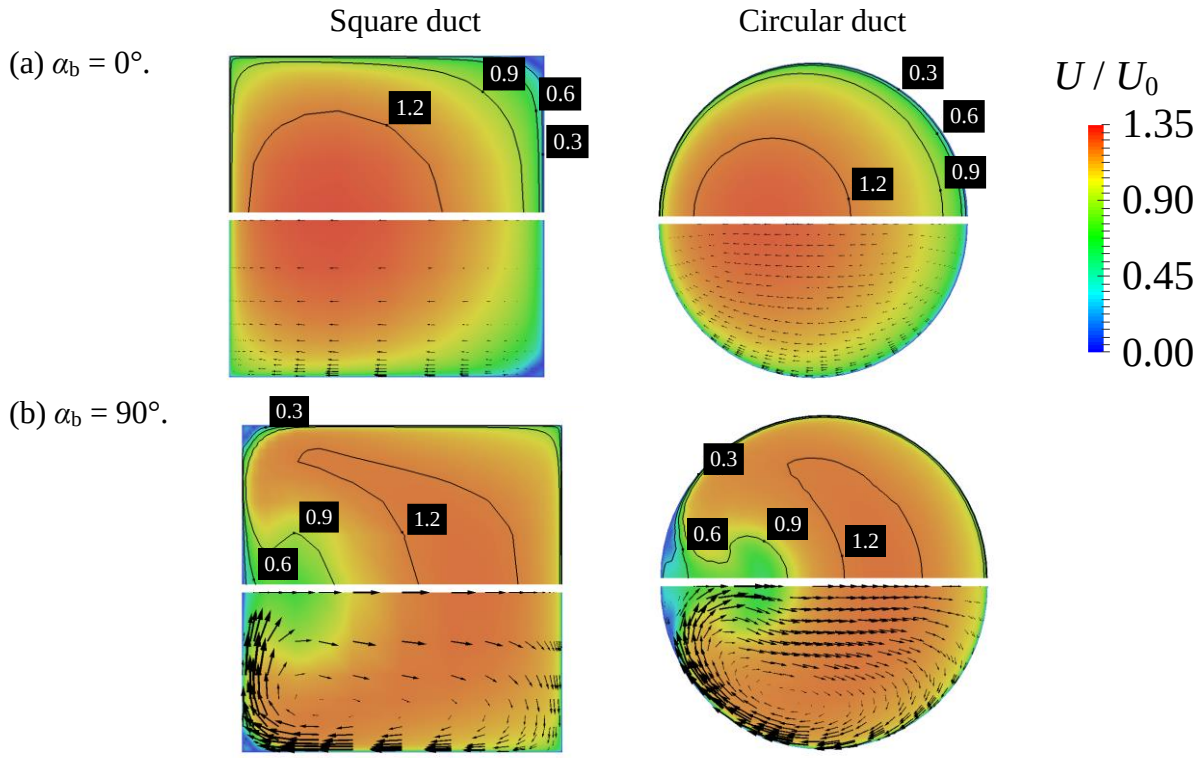


Figure 5.57 – Fluid-phase: comparison of the mean velocity contours at some planes for (a) square and (b) circular ducts.

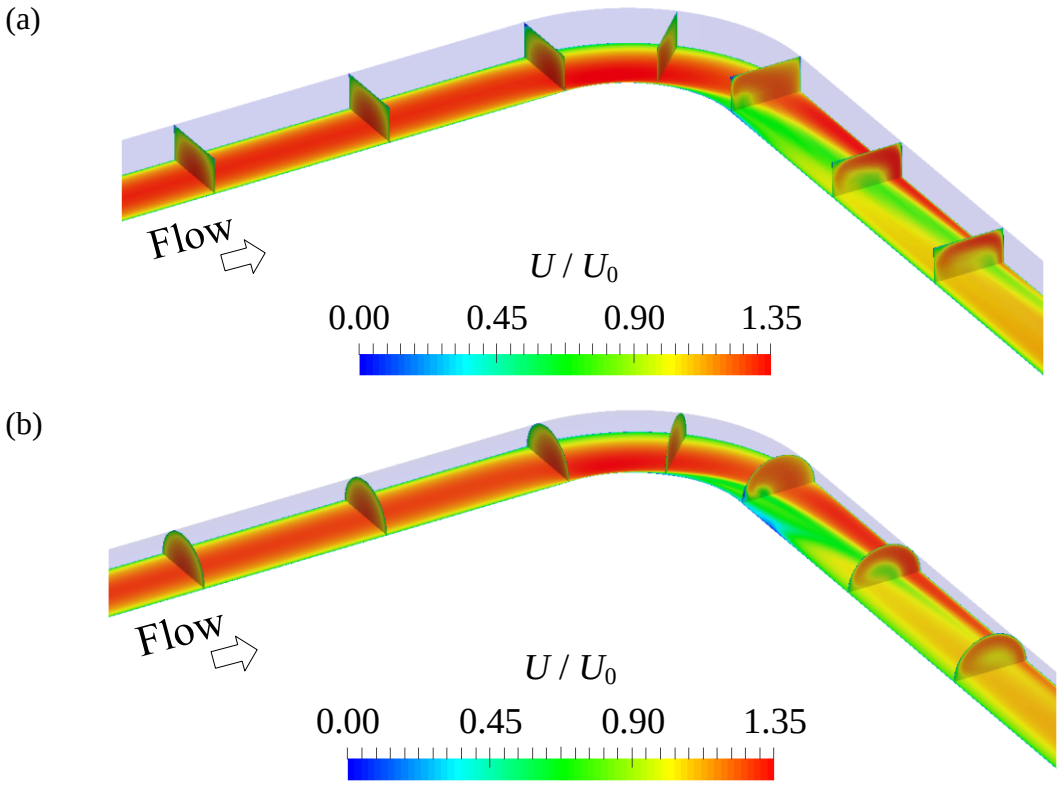
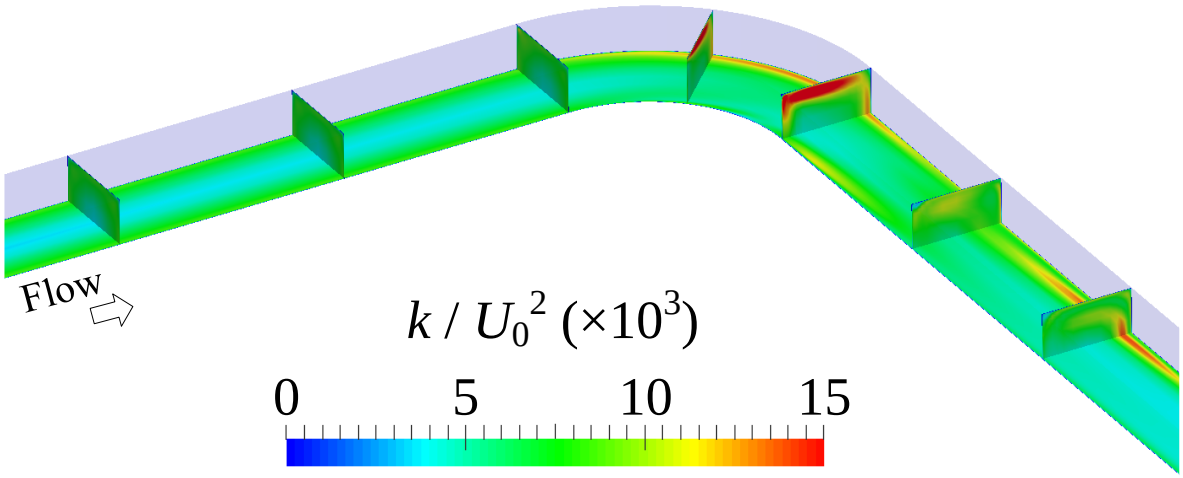
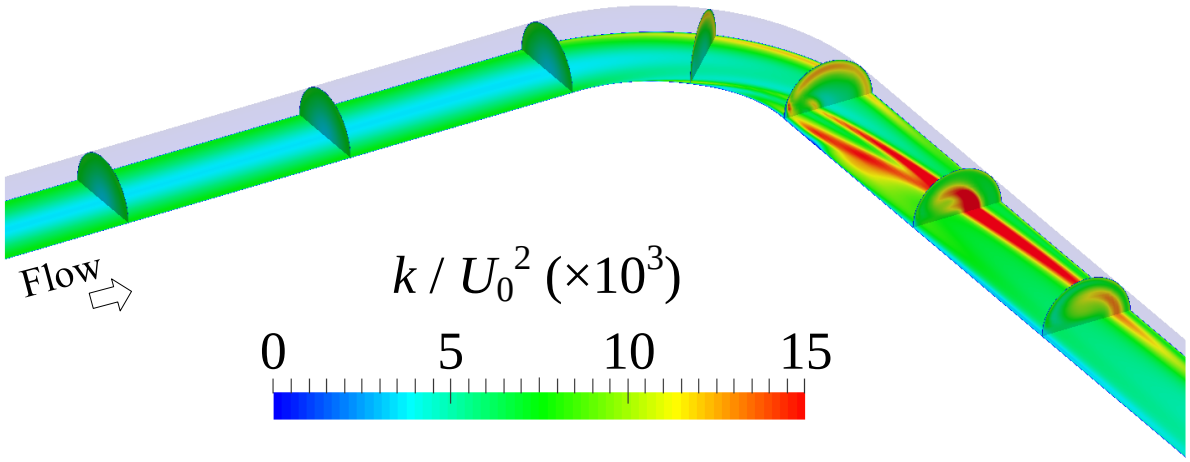


Figure 5.58 – Fluid-phase: comparison of the turbulent kinetic energy contours at some planes for (a) square and (b) circular ducts.



(a) Square duct.



(b) Circular duct.

5.3.4 Particulate-phase: V&V

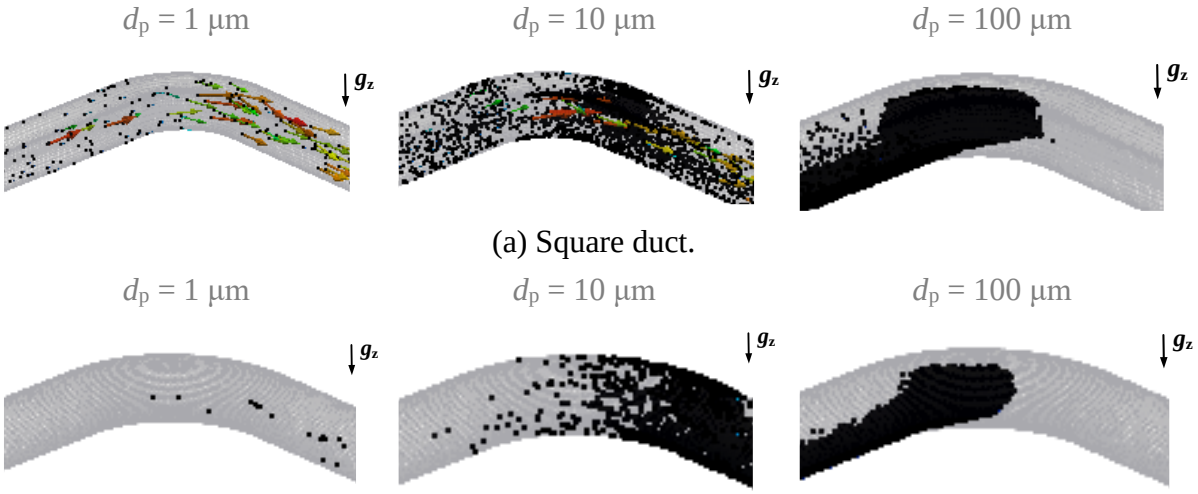
A monodisperse cloud with 10^5 particles was injected at once in the upstream duct at $x = 40D_h$ and tracked with a time interval of 0.35 s. The particle position tracked for the square and circular ducts at the end of the simulation time for particle diameters ranging from 1 μm to 100 μm are presented in Figure 5.59.

According to Figure 5.59, independently of the cross-section and for the same operating flow conditions, the particles with 1 μm of diameter were totally transported by the turbulent airflow and almost no deposition in the elbow was observed. Contrarily, heavier particles (with the diameter of 100 μm) were almost all deposited mostly along the outer bend wall, due to

their higher inertia. The Stokes number of these particles (calculated by Equation (2.34) that leads to (SIPPOLA, 2002): $St = \frac{C_c \rho_p {d_p}^2 U_0}{9 \mu D_h}$) is higher than the unit and the particles tend not to follow the flow streamlines impacting directly against the outer wall of the elbow.

The 10 μm particles have an intermediate behavior compared to the lightest particles (1 μm) and the heaviest particles (100 μm), i.e., a portion of the particles was deposited on the duct walls and the remaining particles were transported by the flow for both cases (square and circular ducts). However, we noted a higher deposition of the 10 μm particles in the square upstream duct than in the circular upstream duct.

Figure 5.59 – Particle transport and deposition for particle diameters ranging from 1 μm to 100 μm during 0.35 s after the injection for the (a) square and (b) circular ducts.



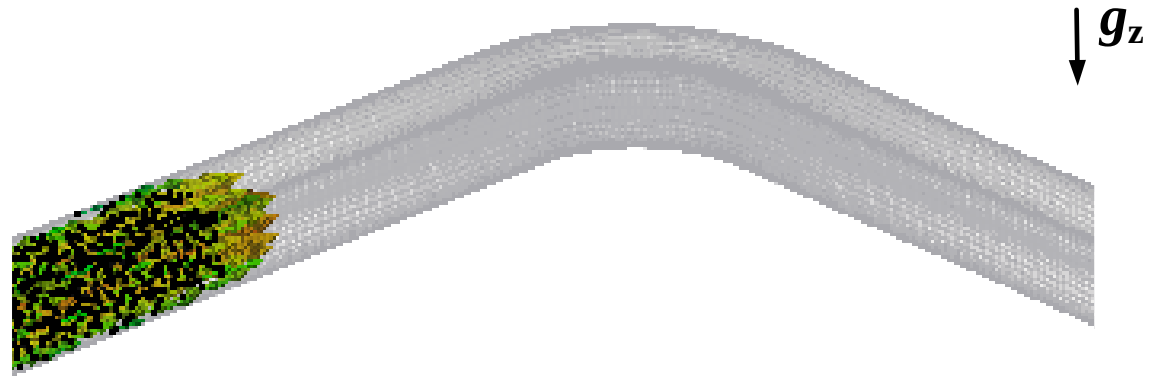
(a) Square duct.

(b) Circular duct.

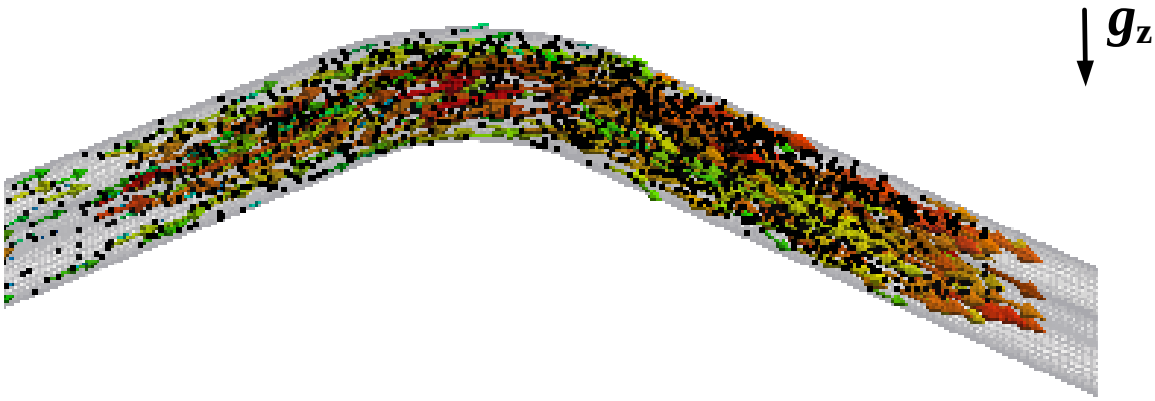
An important question remains: how do the particles evolve during the particle-laden flow? To answer this question the Figure 5.60 and Figure 5.61 present an instantaneous picture of the transport of the 10 μm particles at three times for the square and the circular ducts, respectively. At the initial time, the particles moved towards the elbows with the same local fluid velocity, 0.14 s after the injection the particles crossed the elbow and at 0.28 s after the injection the not deposited particles remained transported by the airflow.

Comparing the duct cross-section, for the circular duct at 0.14 s almost no particle was deposited on the upstream circular duct wall (Figure 5.61b), whereas a higher particle deposition was observed on the upstream square duct at the same time (Figure 5.60b). It seems that at 0.28 s more particles were deposited on the square bend wall (Figure 5.60c) than on the circular bend wall (Figure 5.61c).

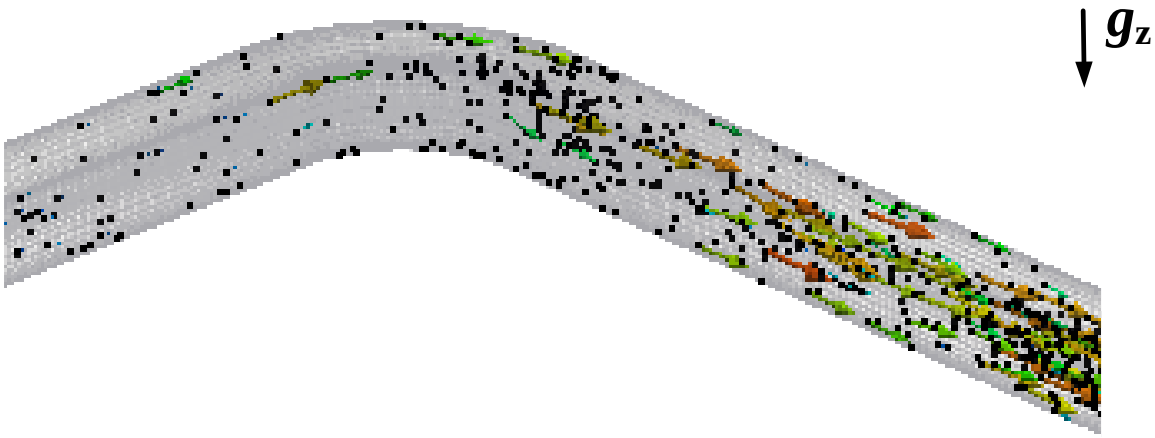
Figure 5.60 – Particle transport and deposition for the 10 μm particle in the square duct for the following times: (a) injection, (b) 0.14 s and (c) 0.28 s after the injection.



(a) t = 0.00s.

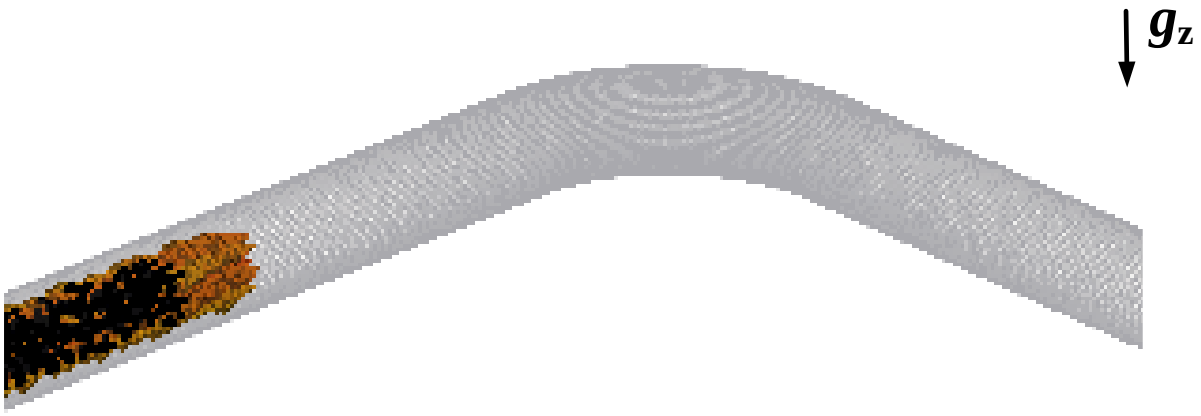


(b) t = 0.14s.

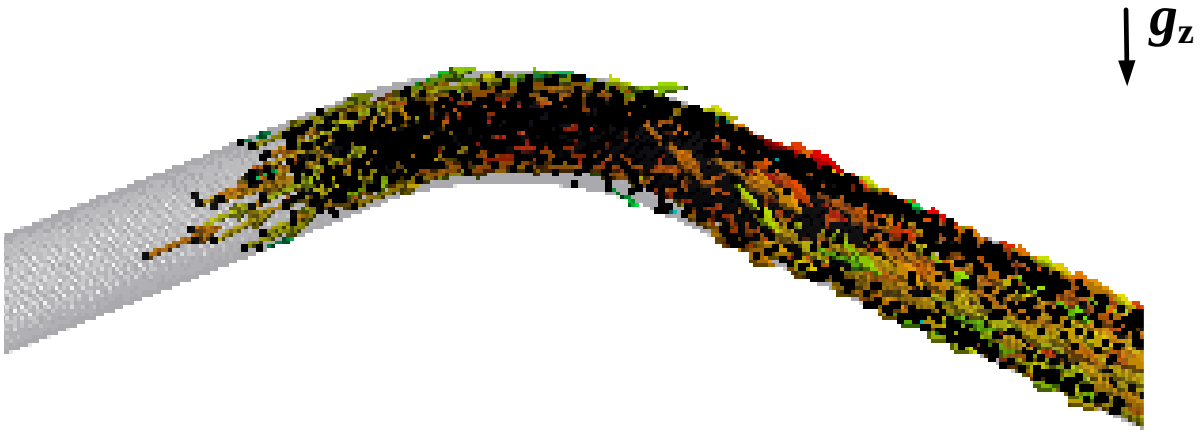


(c) t = 0.28 s.

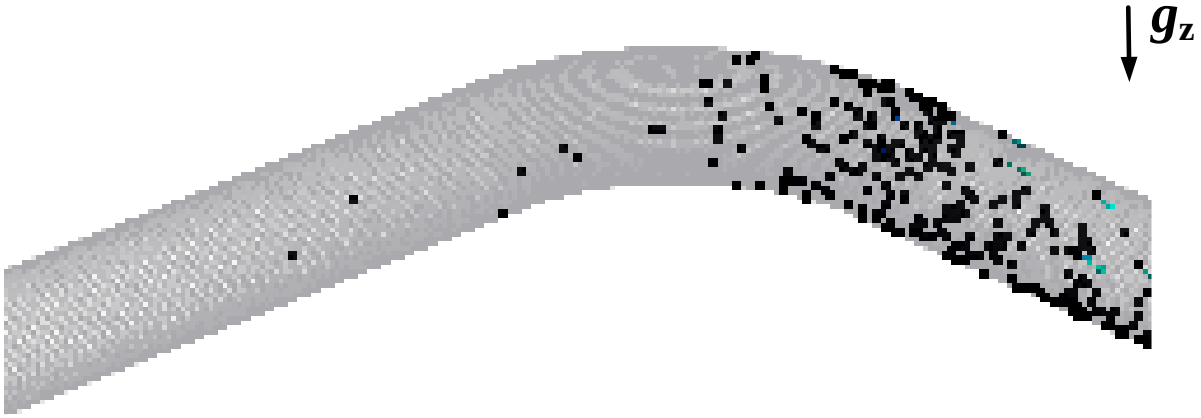
Figure 5.61 – Particle transport and deposition for the 10 μm particles in the circular duct for the following times: (a) injection, (b) 0.14 s and (c) 0.28 s after the injection.



(a) t = 0.00s.



(b) t = 0.14s.

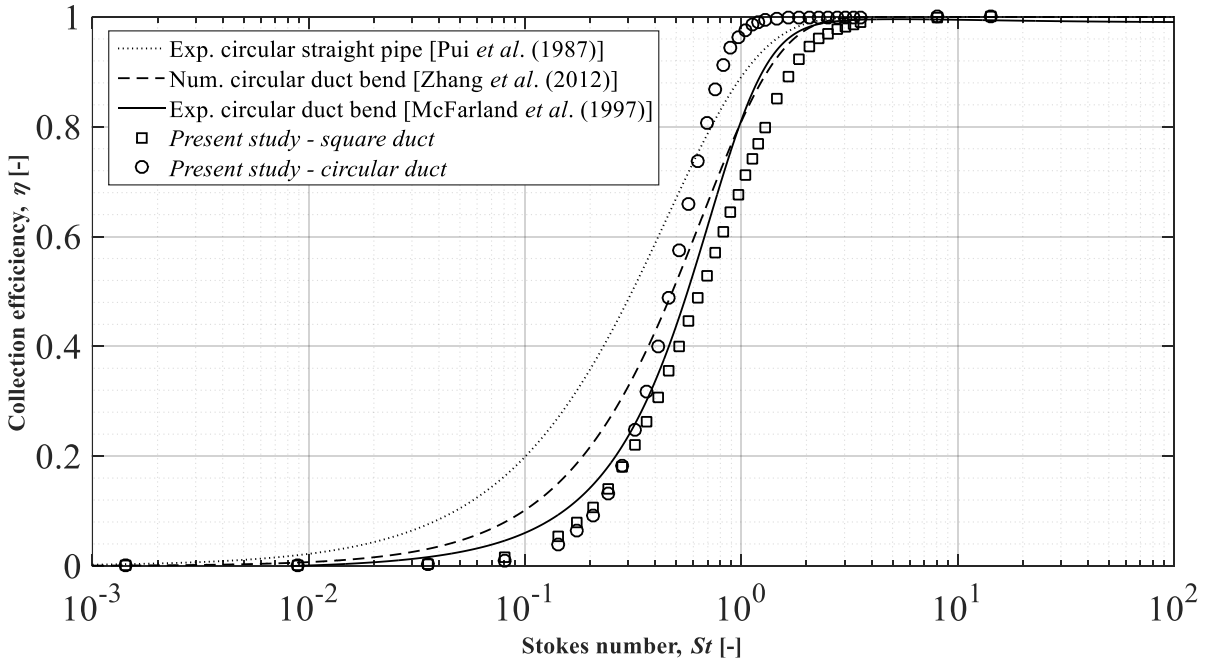


(c) t = 0.28 s.

Particle-laden flows with important deposition rates as in this case can be analyzed with the collection efficiency, η, given by Equation (2.36) in order to quantify the previous results (Figure 5.59–Figure 5.61), i.e., the particle deposition on the bend wall. The deposition efficiency regarding the square-sectioned and circular-sectioned elbows is presented in Figure 5.62 for particle diameters ranging from 0.1 μm to 100 μm, i.e., the Stokes number (Equation (2.34)) ranging from 2×10^{-3} to 14, respectively. The results were validated with the correlations deduced from experiments by PUI *et al.* (1987) (Equation (3.11)) and (MCFARLAND *et al.*, 1997) (Equation (3.12)), the former refers to a circular straight pipe and the latter refers to a circular duct bend. In addition, the findings of this study are also verified against the numerical data of ZHANG *et al.* (2012) (Equation (3.13)) who applied the Eulerian-Lagrangian method to study particle dispersion and deposition in a circular duct bend (more information about each of these studies were given in Table 3.1 and Table 3.3).

For both square and circular ducts, the lighter particles ($St \leq 0.1$) were almost all transported by the turbulent airflow through the elbow, therefore the collection efficiency is almost 0% because there was almost no particle deposition (Figure 5.62). On the other hand, almost all heavier particles ($St \geq 2$) were deposited on the elbow wall (mostly on the outer bend wall), independently of the duct cross-section. The behavior of both curves for the square and the circular duct bends agreed with the experimental data.

Figure 5.62 – Elbow deposition efficiency.

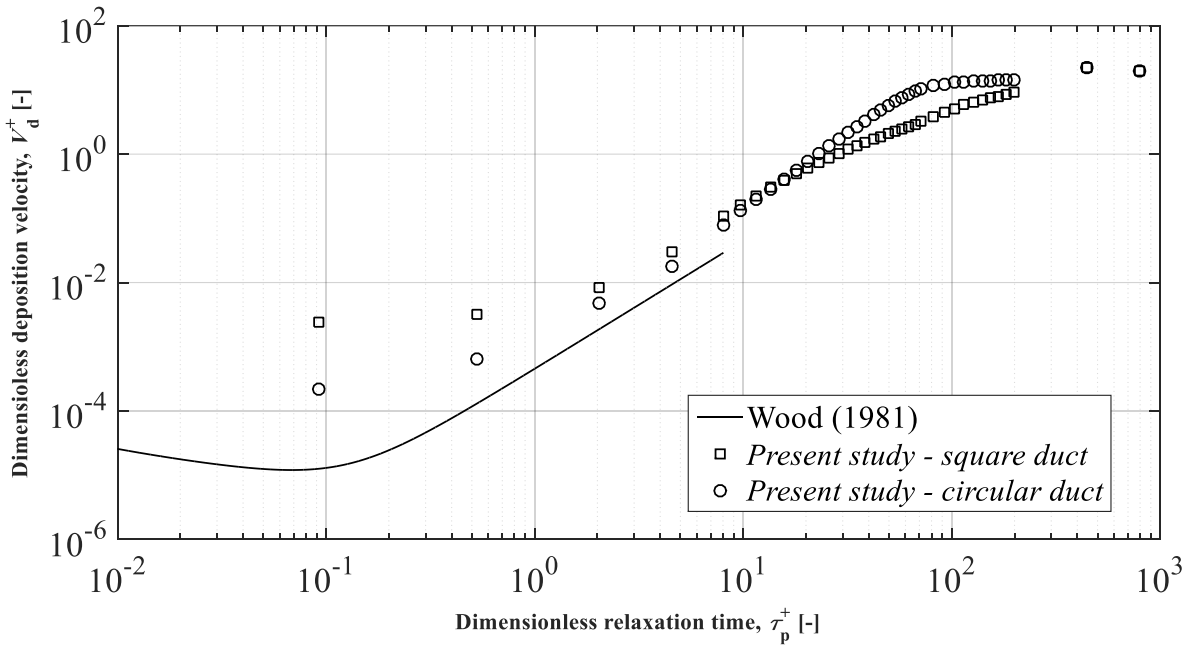


The elbow deposition efficiency for both duct cross-sections was almost the same for Stokes number up to ~0.4 and for Stokes number higher than ~3 (Figure 5.62). However, the elbow collection efficiency ranging from Stokes number of ~0.4 to ~3 was higher for the circular duct than for the square duct, i.e., the deposition rates were higher in the circular duct. Besides, when comparing with the experimental data, the collection efficiency of the circular elbow is over-predicted, while it is under-predicted for the square duct.

The dimensionless deposition velocity, V_d^+, given by Equation (2.28), is also a good parameter to analyze the deposition rates. Figure 5.63 shows the elbow dimensionless deposition velocity for both square and circular ducts compared with Wood's correlation (WOOD, 1981) given by Equation (3.1), not valid for the particle inertia moderate regime (Figure 3.1). The dimensionless relaxation time, τ_p^+, given by Equation (2.27), ranging from 9×10^{-2} to 8×10^{2}, indicates the particle diameter ranges from 0.1 μm to 100 μm, respectively.

Both square-sectioned and circular-sectioned cases clearly yielded the three characteristics regimes of this curve, i.e.: the turbulent diffusion regime ($\tau_p^+ < 2$), the eddy-diffusion impaction regime ($2 \leq \tau_p^+ \leq 50$) and the particle inertia moderate regime ($\tau_p^+ > 50$). In addition, in the second zone ($2 \leq \tau_p^+ \leq 50$) the deposition velocity of the circular duct increased with a higher growth rate than the square duct.

Figure 5.63 – Elbow dimensionless deposition velocity.



6 CONCLUSION AND FUTURE PERSPECTIVES

The focus of this research was to accurately predict by numerical simulations the diluted gas-solid turbulent flow through duct bends of ventilation and air-conditioning systems and to analyze the influence of the duct shape on particle dispersion in curved ducts. The airflow is turbulent in these ducts and the particles were modeled as monodisperse rigid spheres lower than 100 μm size. Those predictions helped to better understand the physical phenomena involved in this kind of fittings and to determinate the particle dispersion and deposition on them.

Three numerical studies were performed using an Eulerian-Lagrangian approach in order to evaluate the capability of Reynolds Averaged Navier-Stokes turbulence models to predict the airflow within curved ducts, the performance of Particle-Wall Interactions models with a Lagrangian Tracking algorithm to solve the particle dispersion in the turbulent flows, and the influence of the duct shape (square and circular ducts) on deposition rates inside the HVAC elbows.

Regarding the first study, the eight RANS models tested yielded an accurate pressure field. The eddy viscosity models achieved a better computing cost with an accurate prediction of the mean velocity despite the curvature of the streamlines. However, the linear eddy viscosity models underestimated the turbulence intensities, whereas the non-linear eddy viscosity models and the Reynolds stress models were more accurate. Summarizing, the following RANS turbulence models were selected: a low Reynolds number k-ε model (the k-ε model of Launder and Sharma), the SST k-ω model, the v^2f model, and the RSM SSG.

The second study proposed the combination of the selected RANS turbulence models from the first study with four particle-wall interaction models to run 2D and 3D simulations of the turbulent flows leading to a total of 32 combinations.

Concerning the 2D simulations, the best agreement between the numerical model and experimental data results was obtained with Brauer's PWI and Matsumoto & Saito's PWI models. However, a 2D simulation does not consider the influence of the front and back faces of the duct.

As expected, the 3D simulations yielded a better agreement of the simulated mean fluid velocity profile with the experimental one. In addition, Brauer's PWI and Brach & Dunn's PWI models provided good results. Since Brach & Dunn's PWI model is based on the impulse

equations and includes the influence of the adhesion force, it is more reliable than Brauer's model and it can be applied to a wide range of cases.

In the investigation of HVAC duct shape influence on particle deposition in elbows, the turbulence model chosen was the SST k-ω due to the results of the previous studies and the PWI was simplified with a null restitution coefficient to force the particle deposition on duct walls. No significant variation was observed regarding the deposition rates between the square and circular duct, and the models agreed with the current literature of collection efficiency and the deposition velocity.

The principal contributions of this thesis were an insightful systematic methodology for V&V using an Eulerian-Lagrangian approach of the fluid-phase and the particulate-phase findings with the proposed RANS-LPT/PWI combinations by adding features in OpenFOAM® Lagrangian library, the radar chart to select an appropriate model and the study of the duct shape influence on particle deposition rates in HVAC elbows.

Finally, further suggestions for future studies are:

- to improve the fluid-phase numerical solution by using turbulence models that provide instantaneous velocities of the large-scale structures of the flow;
- to implement a resuspension model for the particulate-phase with the LPT approach in order to enhance the deposition model;
- to add the influence of the wall on drag and shear-lift forces in the LPT particle-fluid interactions;
- to propose correlations of particle collection efficiency varying with the aspect ratio of rectangular and round ventilation ducts;
- to investigate the influence of the HVAC curvature ratio and duct orientation on the dispersion, deposition, and resuspension of particle-laden turbulent flows.

REFERENCES

ABE, K.; KONDOH, T.; NAGANO, Y. A new turbulence model for predicting fluid flow and heat transfer in separating and reattaching flows-I. Flow field calculations. **International Journal of Heat and Mass Transfer**, v. 37, n. 1, p. 139–151, 1994.

ALMOHAMMED, N.; BREUER, M. Modeling and simulation of particle–wall adhesion of aerosol particles in particle-laden turbulent flows. **International Journal of Multiphase Flow**, v. 85, p. 142–156, 2016.

ANSELMET, F.; TERNAT, F.; AMIELH, M.; BOIRON, O.; BOYER, P.; PIETRI, L. Axial development of the mean flow in the entrance region of turbulent pipe and duct flows. **Comptes Rendus - Mecanique**, v. 337, n. 8, p. 573–584, 2009.

ARCEN, B.; TANIÈRE, A.; OESTERLÉ, B. On the influence of near-wall forces in particle-laden channel flows. **International Journal of Multiphase Flow**, v. 32, n. 12, p. 1326–1339, 2006.

ASHRAE. **ASHRAE HANDBOOK: Fundamentals, SI Edition**. SI ed. United States of America: American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc., 2017.

BASSE, N. T. **Turbulence intensity and the friction factor for smooth- and rough-wall pipe flow**. Denmark, 2017.

BEAM, R. M.; WARMING, R. F. An implicit finite-difference for hyperbolic sytems in conservation-low form. **Journal of Computational Physics**, v. 22, p. 87–110, 1976.

BERGER, S. A.; TALBOT, L. Flow in curved pipes. **Annual Review of Fluid Mechanics**, n. 15, p. 461–512, 1983.

BERROUK, A. S.; LAURENCE, D. Stochastic modelling of aerosol deposition for LES of 90° bend turbulent flow. **International Journal of Heat and Fluid Flow**, v. 29, p. 1010–1028, 2008.

BIRD, R. B.; STEWART, W. E.; LIGHTFOOT, E. N. **Transport Phenomena**. 2nd. ed. United States of America: John Wiley & Sons, Inc., 2002.

BRACH, R. M.; DUNN, P. F. A mathematical model of the impact and adhesion of microsphers. **Aerosol Science and Technology**, v. 16, n. 1, p. 51–64, 1992.

BRACH, R. M.; DUNN, P. F. Models of rebound and capture for oblique microparticle impacts. **Aerosol Science and Technology**, v. 29, n. 5, p. 379–388, 1998.

BRAUER, H. Report on investigations on particle movement in straight horizontal tubes, particle-wall collision and erosion of tubes and tube bends. **Journal of powder & bulk solids technology**, v. 4, n. 2–3, p. 3–12, 1980.

BREUER, M.; BAYTEKIN, H. T.; MATIDA, E. A. Prediction of aerosol deposition in 90° bends using LES and an efficient Lagrangian tracking method. **Journal of Aerosol Science**, v. 37, n. 11, p. 1407–1428, 2006.

ÇENGEL, Y. A.; CIMBALA, J. M. **Fluid Mechanics: Fundamentals and Applications**. 1st.

ed. New York: McGraw-Hill, 2006.

CHENG, Y. S.; WANG, C. S. Motion of particles in bends of circular pipes. **Atmospheric Environment**, v. 15, n. 3, p. 301–306, 1981.

CLIFT, R.; GRACE, J. R.; WEBER, M. E. **Bubbles, Drops, and Particles**. United States of America: Academic Press, 1978.

COLBECK, I.; LAZARIDIS, M. **Aerosol Science: Technology and Applications**. United Kingdom: Wiley, 2014.

CROWE, C. T.; SCHWARZKOPF, J. D.; SOMMERFELD, M.; TSUJI, Y. **Multiphase Flows with Droplets and Particles**. 2nd. ed. Boca Raton, USA: Taylor & Francis Group, 2012.

DAHNEKE, B. The capture of aerosol particles by surfaces. **Journal of Colloid And Interface Science**, v. 37, n. 2, p. 342–353, 1971.

DAHNEKE, B. The influence of flattening on the adhesion of particles. **Journal of Colloid And Interface Science**, v. 40, n. 1, p. 1–13, 1972.

DAVIDSON, L.; NIELSEN, P.; SVENINGSSON, A. Modifications of the v2-f model for computing the flow in a 3D wall jet. **Turbulence, Heat and Mass Transfer**, v. 4, p. 577–584, 2003.

DEAN, W. R. LXXII. The stream-line motion of fluid in a curved pipe (Second paper). **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 5, n. 30, 1928.

DeANGELIS, B. C.; LOTH, E.; LANKFORD, D.; BARTLETT, C.S. Computations of turbulent droplet dispersion for wind tunnel tests. **AIAA Journal of Aircraft**, v. 34, n. 2, p. 213–219, 1997.

DENNIS, S. C. R.; NG, M. Dual solutions for steady laminar flow through a curved tube. **Quarterly Journal of Mechanics and Applied Mathematics**, v. 35, n. 3, p. 305–324, 1982.

DURBIN, P. A. Near-wall turbulence closure modeling without “damping functions”. **Theoretical and Computational Fluid Dynamics**, v. 3, n. 1, p. 1–13, 1991.

DURBIN, P. A. Separated flow computations with the k-ε-v2 model. **AIAA Journal of Aircraft**, v. 33, n. 4, p. 659–664, 1995.

EL-SHOBOKSHY, M. S. Experimental measurements of aerosol deposition to smooth and rough surfaces. **Atmospheric Environment**, v. 17, p. 639–644, 1983.

EL-SHOBOKSHY, M. S.; ISMAIL, I. A. Deposition of aerosol particles from turbulnet flow onto rough pipe wall. **Atmospheric Environment**, v. 14, p. 297–304, 1980.

ELGHOBASHI, S. On predicting particle-laden turbulent flows. **Applied Scientific Research**, v. 52, n. 4, p. 309–329, 1994.

FAN, F. G.; AHMADI, G. A sublayer model for turbulent deposition of particles in vertical ducts with smooth and rough surfaces. **Journal of Aerosol Science**, v. 24, n. 1, p. 45–64, 1993.

FAN, L.S.; ZHU, C. **Principles of Gas-Solid Flows**. United Kingdom: Cambridge University Press, 2005.

FAXÉN, H. Die Bewegung einer starren Kugel längs der Achse eines mit zäher Flüssigkeit gefüllten Rohres. **Arkiv Mat Astron Fys**, v. 17, n. 27, p. 1–28, 1923.

FERZIGER, J. H.; PERIC, M. **Computational Methods for Fluid Dynamics**. 3rd. ed. New York: Springer-Verlag Berlin Heidelberg, 2002.

FOX, R. W.; PRITCHARD, P. J.; MCDONALD, A. T. **Introduction to Fluid Mechanics**. 8th. ed. United States of America: John Wiley & Sons, Inc., 2011.

FRIEDLANDER, S. K. **Smoke, Dust, and Haze: Fundamentals of Aerosol Dynamics**. 2nd. ed. New York: Oxford University Press, Inc., 2000.

GAO, R.; LI, A. Modeling deposition of particles in vertical square ventilation duct flows. **Building and Environment**, v. 46, n. 1, p. 245–252, 2011.

GAO, R.; LI, A. Dust deposition in ventilation and air-conditioning duct bend flows. **Energy Conversion and Management**, v. 55, p. 49–59, 2012.

GIBSON, M. M.; JONES, W. P.; YOUNIS, B. A. Calculation of turbulent boundary layers on curved surfaces. **Physics of Fluids**, v. 24, n. 3, p. 386–395, 1981.

GORHAM, D. A.; KHARAZ, A. H. The measurement of particle rebound characteristics. **Powder Technology**, v. 112, p. 193–202, 2000.

GOSMAN, A. D.; IONNIDES, E. Aspects of computer simulation of liquid-fueled combustors. **Journal of Energy**, v. 7, n. 6, p. 482–490, 1983.

GRANT, G.; TABAKOFF, W. Erosion prediction in turbomachinery resulting from environmental solid particles. **Journal of Aircraft**, v. 12, n. 5, p. 471–478, 1975.

GREENSHIELDS, C. J. **OpenFOAM User Guide**. England: OpenFOAM Foundation Ltd., 2018.

GUHA, A. Transport and Deposition of Particles in Turbulent and Laminar Flow. **Annual Review of Fluid Mechanics**, v. 40, n. 1, p. 311–341, 2008.

HENRY, C.; MINIER, J. P. Progress in particle resuspension from rough surfaces by turbulent flows. **Progress in Energy and Combustion Science**, v. 45, n. C, p. 1–53, 2014.

HOLZINGER, G. **OpenFOAM: a little user-manual**. Austria (Linz): CD-Laboratory-Particulate Flow Modelling - Johannes Kepler University, 2019.

HOLZMANN, T. **Mathematics, Numerics, Derivations and OpenFOAM®**. 4th. ed. Germany: http://www.holzmann-cfd.de, 2018.

HONG, W.; WANG, X.; ZHENG, J. Numerical study on particle deposition in rough channels with different structure parameters of rough elements. **Advanced Powder Technology**, v. 29, n. 11, p. 2895–2903, 2018.

HONGTAO, L.; LI, Z. Prediction of particle deposition characteristic in 90° square bend : square bend particle deposition characteristic. **Applied Thermal Engineering**, v. 31, n. 16, p. 3402–3409, 2011.

HUSSEIN, T.; KUBINCOVA, L.; DZUMBOVÁ, L.; HRUSKA, A.; DOHANYOSOVA, P.; HEMERKA, J.; SMOLIK, J. Deposition of aerosol particles on rough surfaces inside a test

chamber. **Building and Environment**, v. 44, n. 10, p. 2056–2063, 2009.

HUSSEIN, T.; SMOLIK, J.; KERMINEN; V.M.; KULMALA, M. Modeling dry deposition of aerosol particles onto rough surfaces. **Aerosol Science and Technology**, v. 46, n. 1, p. 44–59, 2012.

INTERNATIONAL ENERGY AGENCY. **Energy and Air Pollution**. World Energy Outlook - Special Report, p. 266, 2016.

ISRAELACHVILI, J. N. **Intermolecular and surface forces**. United States of America: Elsevier, 2011.

ISSA, R. I. Solution of the implicitly discretised fluid flow equations by operator-splitting. **Journal of Computational Physics**, v. 62, n. 1, p. 40–65, 1986.

ITŌ, H. Pressure losses in smooth pipe bends. **Journal of Basic Engineering**, v. 82, n. 1, p. 131, 1960.

JENNINGS, S. The mean free path in air. **Journal of Aerosol Science**, v. 19, n. 2, p. 159–166, 1988.

JIN, C.; POTTS, I.; REEKS, M. W. The effects of near wall corrections to hydrodynamic forces on particle deposition and transport in vertical turbulent boundary layers. **International Journal of Multiphase Flow**, v. 79, p. 62–73, 2016.

KLEBANOFF, P. S. Characteristics of turbulence in a boundary layer with zero pressure gradient. **TN 3178**, 1954.

KLIAFAS, Y. **Two phase turbulent flow measurements in a ninety degree bend using laser doopler anemometry**. University of California, 1984.

KLIAFAS, Y.; HOLT, M. LDV measurements of a turbulent air-solid two-phase flow in a 90º bend. **Experiments in Fluids**, v. 5, n. 2, p. 73–85, 1987.

KUAN, B.; YANG, W.; SCHWARZ, M. Dilute gas–solid two-phase flows in a curved 90° duct bend: CFD simulation with experimental validation. **Chemical Engineering Science**, v. 62, n. 7, p. 2068–2088, 2007.

KUEHN, T. H.; RAMSEY, J. W.; THRELKELD, J. L. **Thermal Environmental Engineering**. 3rd. ed. New Jersey: Prentice Hall, 1998.

KVASNAK, W.; AHMADI, G.; BAYER, R.; GAYNES, M. Experimental investigation of dust particle deposition in a turbulent channel flow. **Journal of Aerosol Science**, v. 24, n. 6, p. 795–815, 1993.

LAI, A. C. K.; NAZAROFF, W. W. Modelling indoor particle deposition from trubulent flow onto smooth sufaces. **Journal of Aerosol Science**, v. 31, n. 4, p. 463–476, 2000.

LAI, A. C. K.; NAZAROFF, W. W. Supermicron particle deposition from turbulent chamber flow onto smooth and rough vertical surfaces. **Atmospheric Environment**, v. 39, n. 27, p. 4893–4900, 2005.

LAI, M.; KREMPL, E.; RUBEN, D. **Introduction to Continuum Mechanics**. United Kingdom: Elsevier Inc., 2010.

LAUNDER, B. E.; REECE, G. J.; RODI, W. Progress in the development of a Reynolds-stress turbulence closure. **Journal of Fluid Mechanics**, v. 68, n. 03, p. 537, 1975.

LAUNDER, B. E.; SHARMA, B. I. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. **Letters in Heat and Mass Transfer**, v. 1, n. 2, p. 131–137, 1974.

LEVY, A.; MASON, D. J. The effect of a bend on the particle cross-section concentration and segregation in pneumatic conveying systems. **Powder Technology**, v. 98, n. 2, p. 95–103, 1998.

LI, A.; AHMADI, G. Dispersion and Deposition of Spherical Particles from Point Sources in a Turbulent Channel Flow. **Aerosol Science and Technology**, v. 16, n. 4, p. 209–226, 1992.

LIEN, F.-S.; KALITZIN, G. Computations of transonic flow with the v2–f turbulence model. **International Journal of Heat and Fluid Flow**, v. 22, n. 1, p. 53–61, 2001.

LIEN, F.-S.; LESCHZINER, M. A. A pressure-velocity solution strategy for compressible flow and its application to shock/boundary-layer interaction using second-moment turbulence closure. **Journal of Fluids Engineering**, v. 115, n. 4, p. 717, 1993.

LIEN, F. S.; CHEN, W. L.; LESCHZINER, M. A. Low-reynolds-number eddy-viscosity modelling based on non-linear stress-strain/vorticity relations. **Engineering Turbulence Modelling and Experiments**, p. 91–100, 1996.

LIN, C. S.; MOULTON, R. W.; PUTNAM, G. L. Mass Transfer between Solid Wall and Fluid Streams. Mechanism and Eddy Distribution Relationships in Turbulent Flow. **Industrial & Engineering Chemistry**, v. 45, n. 3, p. 636–640, 1953.

LIU, B. Y.; AGARWAL, J. K. Experimental observation of aerosol deposition in turbulent flow. **Journal of Aerosol Science**, v. 5, n. 2, p. 145–155, 1974.

LOTH, E. Numerical approaches for motion of dispersed particles, droplets and bubbles. **Progress in Energy and Combustion Science**, v. 26, n. 3, p. 161–223, 2000.

LOYSEAU, X. F.; VERDIN, P. G. Statistical model of transient particle dispersion and deposition in vertical pipes. **Journal of Aerosol Science**, v. 101, p. 43–64, 2016.

LU, H.; LU, L. Numerical investigation on particle deposition enhancement in duct air flow by ribbed wall. **Building and Environment**, v. 85, p. 61–72, 2015.

LU, H.; LU, L.; JIANG, Y. Numerical study of monodispersed particle deposition rates in variable-section ducts with different expanding or contracting ratios. **Applied Thermal Engineering**, v. 110, p. 150–161, 2017.

LUO, J.; LAKSHMINAMYANA, B. Prediction of strongly curved turbulent duct flows with reynolds stress model. **AIAA**, v. 35, n. 1, p. 91–98, 1997.

MACPHERSON, G. B.; NORDIN, N.; WELLER, H. G. Particle tracking in unstructured , arbitrary polyhedral meshes for use in CFD and molecular dynamics. **Communications in Numerical Methods in Engineering**, n. 25, p. 263–273, 2009.

MAJLESARA, M.; SALMANZADEH, M.; AHMADI, G. A model for particles deposition in turbulent inclined channels. **Journal of Aerosol Science**, v. 64, p. 37–47, 2013.

MARCHIOLI, C.; PICCIOTTO, M.; SOLDATI, A. Influence of gravity and lift on particle velocity statistics and transfer rates in turbulent vertical channel flow. **International Journal of Multiphase Flow**, v. 33, n. 3, p. 227–251, 2007.

MARIC, T.; HÖPKEN, J.; MOONEY, K. **The OpenFOAM® Technology Primer**. Germany: SourceFlux, 2014.

MATIDA, E. A.; NISHINO, K.; TORII, K. Statistical simulation of particle deposition on the wall from turbulent dispersed pipe flow. **International Journal of Heat and Fluid Flow**, v. 21, n. 4, p. 389–402, 2000.

MATSUMOTO, S.; SAITO, S. Monte Carlo simulation of horizontal pneumatic conveying based on the rough wall model. **Journal of Chemical Engineering of Japan**, v. 3, n. 2, p. 223–230, 1970.

MAUDE, A. The movement of a sphere in front of a plane at low reynolds number. **Brit. J. Appl. Phys.**, v. 14, p. 894–898, 1963.

MAXEY, M. R.; RILEY, J. J. Equation of motion of a small rigid sphere in a non uniform flow. **Physics of Fluids**, v. 26, p. 883–889, 1983.

MCFARLAND, A. R.; GONG, H.; MUYSHONDT, A.; WENTE, W.B.; ANAND, N.K. Aerosol deposition in bends with turbulent flow. **Environmental Science and Technology**, v. 31, n. 12, p. 3371–3377, 1997.

MCLAUGHLIN, J. B. Inertial migration of a small sphere in linear shear flows. **Journal of Fluid Mechanics**, v. 224, p. 261–274, 1991.

MEI, R. An approximate expression for the shear lift force on a spherical particle at finite reynolds number. **International Journal of Multiphase Flow**, v. 18, n. 1, p. 145–147, 1992.

MENTER, F.; EGOROV, Y. A Scale Adaptive Simulation Model using Two-Equation Models. **43rd AIAA Aerospace Sciences Meeting and Exhibit**, n. January, 2005.

MENTER, F.; KUNTZ, M.; LANGTRY, R. Ten years of industrial experience with the SST turbulence model. **Turbulence, heat and mass transfer**, p. 8, 2003.

MENTER, F. R. Separated Adverse Pressure Gradient Flows. **AIAA Journal**, v. 30, n. 8, p. 2066–2072, 1992.

MIGUEL, A. F.; REIS, A. H.; AYDIN, M. Aerosol particle deposition and distribution in bifurcating ventilation ducts. **Journal of Hazardous Materials**, v. 116, n. 3, p. 249–255, 2004.

MINIER, J.-P.; POZORSKI, J. **Particles in Wall-Bounded Turbulent Flows: Deposition, Re-Suspension and Agglomeration**. CISM International Centre for Mechanical Sciences: Springer International Publishing, 2017.

MOHANARANGAM, K.; TIAN, Z. F.; TU, J. Y. Numerical simulation of turbulent gas-particle flow in a 90^{o} bend: Eulerian-Eulerian approach. **Computers and Chemical Engineering**, v. 32, n. 3, p. 561–571, 2008.

NAIK, S.; BRYDEN, I. G. Prediction of turbulent gas-solids flow in curved ducts using Eulerian-Lagrangian method. **International Journal for Numerical Methods in Fluids**, v. 31, p. 579–600, 1999.

NJOBUENWU, D. O.; FAIRWEATHER, M.; YAO, J. Coupled RANS-LPT modelling of dilute, particle-laden flow in a duct with a 90° bend. **International Journal of Multiphase Flow**, v. 50, p. 71–88, 2013.

OESTERLÉ, B. **Ecoulements multiphasiques: des fondements aux méthodes d'ingénierie**. Paris: Lavoisier, 2006.

PAPAVERGOS, P. G.; HEDLEY, A. B. Particle deposition behaviour from turbulent flows. **Chemical Engineering Research and Design**, v. 62, n. 5, p. 275–295, 1984.

PATANKAR, S. V. **Numerical heat transfer and fluid flow**. New York: McGraw-Hill, 1980.

PATANKAR, S. V.; SPALDING, D. B. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. **International Journal of Heat and Mass Transfer**, v. 15, n. 10, p. 1787–1806, 1972.

PENNEY, D. Guidelines for indoor air quality. **WHO Guidelines**, v. 9, p. 454, 2010.

PETERS, T. M.; LEITH, D. Measurement of particle deposition in industrial ducts. **Journal of Aerosol Science**, v. 35, n. 4, p. 529–540, 2004a.

PETERS, T. M.; LEITH, D. Particle deposition in industrial duct bends. **Annals of Occupational Hygiene**, v. 48, n. 5, p. 483–490, 2004b.

PHUONG, N. L.; ITO, K. Experimental and numerical study of airflow pattern and particle dispersion in a vertical ventilation duct. **Building and Environment**, v. 59, p. 466–481, 2013.

PISKUNOV, V. N. Parameterization of aerosol dry deposition velocities onto smooth and rough surfaces. **Journal of Aerosol Science**, v. 40, n. 8, p. 664–679, 2009.

POPE, S. B. **Turbulent Flows**. United Kingdom: Cambridge University Press, 2000.

PUI, D. Y. H.; ROMAY-NOVAS, F.; LIU, B. Y. H. Experimental Study of Particle Deposition in Bends of Circular Cross Section. **Aerosol Science and Technology**, v. 7, n. 3, p. 301–315, 1987.

RIZK, M. A.; ELGHOBASHI, S. E. The motion of a spherical particle suspended in a turbulent flow near a plane wall. **Physics of Fluids**, v. 28, n. 3, p. 806, 1985.

ROACHE, P. J. Perspective: A Method for Uniform Reporting of Grid Refinement Studies. **Journal of Fluids Engineering**, v. 116, n. 3, p. 405–413, 1994.

RÖHRIG, R.; JAKIRLIĆ, S.; TROPEA, C. Comparative computational study of turbulent flow in a 90° pipe elbow. **International Journal of Heat and Fluid Flow**, v. 55, p. 120–131, 2015.

SAFFMAN, P. G. The lift on a small sphere in a slow shear flow. **Journal of Fluid Mechanics**, v. 22, n. 02, p. 385–400, 1965.

SAFFMAN, P. G. Corrigendum to "The lift on a small sphere in a slow shear flow". **Journal of Fluid Mechanics**, v. 31, p. 624, 1968.

SARIC, W. S. Görtler Vortices. **Annual Review of Fluid Mechanics**, v. 26, n. 1, p. 379–409, 1994.

SCHLICHTING, H.; GERSTEN, K. **Boundary-Layer Theory**. 9th. ed. Berlin: Springer-

Verlag Berlin Heidelberg, 2017.

SEHMEL, G. A. Particle eddy diffusivities and deposition velocities for isothermal flow and smooth surfaces. **Journal of Aerosol Science**, v. 4, n. 2, p. 125–138, 1973.

SIPPOLA, M.; NAZAROFF, W. Particle deposition in ventilation ducts: connectors, bends and developing turbulent flow. **Aerosol Science & Technology**, v. 39, n. 2, p. 139–150, 2005.

SIPPOLA, M. R. **Particle Deposition in Ventilation Ducts**. University of California, Berkeley, 2002.

SOLDATI, A.; MARCHIOLI, C. Physics and modelling of turbulent particle deposition and entrainment: Review of a systematic study. **International Journal of Multiphase Flow**, v. 35, n. 9, p. 827–839, 2009.

SOMMERFELD, M.; HUBER, N. Experimental analysis of modelling of particle-wall collisions. **International Journal of Multiphase Flow**, v. 25, n. 6–7, p. 1457–1489, 1999.

SPALART, P. R.; JOU, W.H.; STRELETS, M.; ALLMARAS, S.R. Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. **Advances in DNS/LES**, p. 4–8, 1997.

SPEDDING, P. L.; BENARD, E.; MCNALLY, G. M. Fluid flow through 90 degree bends. **Dev. Chem. Eng. Mineral Process**, v. 12, p. 107–128, 2004.

SPEZIALE, C. G.; SARKAR, S.; GATSKI, T. B. Modelling the pressure–strain correlation of turbulence: an invariant dynamical systems approach. **Journal of Fluid Mechanics**, v. 227, n. 1, p. 245, 1991.

SUDO, K.; SUMIDA, M.; HIBARA, H. Experimental investigation on turbulent flow in a circular-sectioned 90-degree bend. **Experiments in Fluids**, v. 25, n. 1, p. 42–49, 1998.

SUDO, K.; SUMIDA, M.; HIBARA, H. Experimental investigation on turbulent flow in a square-sectioned 90-degree bend. **Experiments in Fluids**, v. 30, p. 246–252, 2001.

SUN, K.; LU, L.; JIANG, H.; JIN, H. Experimental study of solid particle deposition in 90° ventilated bends of rectangular cross section with turbulent flow. **Aerosol Science and Technology**, v. 47, n. 2, p. 115–124, 2013.

SUN, K.; LU, L. Particle flow behavior of distribution and deposition throughout 90° bends: Analysis of influencing factors. **Journal of Aerosol Science**, v. 65, p. 26–41, 2013.

SUN, K.; LU, L.; JIANG, H. A computational investigation of particle distribution and deposition in a 90° bend incorporating a particle-wall model. **Building and Environment**, v. 46, n. 6, p. 1251–1262, 2011.

SUN, K.; LU, L.; JIANG, H. A numerical study of bend-induced particle deposition in and behind duct bends. **Building and Environment**, v. 52, p. 77–87, 2012.

SVENINGSSON, A. **Analysis of the performance of different v2f turbulence models in a stator vane passage flow**. Chalmers University of Tecnology, 2003.

SWEARINGEN, J. D.; BLACKWELDER, R. F. The growth and breakdown of streamwise vortices in the presence of a wall. **Journal of Fluid Mechanics**, v. 182, p. 255–290, 1987.

TAKAMURA, H.; EBARA, S.; HASHIZUME, H.; AIZAWA, K.; YAMANO, H. Flow Visualization and Frequency Characteristics of Velocity Fluctuations of Complex Turbulent Flow in a Short Elbow Piping Under High Reynolds Number Condition. **Journal of Fluids Engineering**, v. 134, n. 10, p. 101201, 2012.

TANAKA, M.; OHSHIMA, H.; MONJI, H. Numerical Investigation of Flow Structure in Pipe Elbow With Large Eddy Simulation Approach. **ASME 2009 - PVP2009-77598**, p. 1–10, 2009.

TCHEN, C. M. **Mean value and correlation problems connected with the motion of small particles suspended in a turbulent fluid**. TU Delft, 1947.

THOMSON, J. On the origin of windings of rivers in alluvial plains, with remarks on the flow of water round bends in pipes. **Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 25, p. 5–8, 1876.

TIAN, L.; AHMADI, G. Particle deposition in turbulent duct flows-comparisons of different model predictions. **Journal of Aerosol Science**, v. 38, n. 4, p. 377–397, 2007.

TIAN, Z. F.; INTHAVONG, K.; TU, J.Y.; YEOH, G.H. Numerical investigation into the effects of wall roughness on a gas-particle flow in a 90° bend. **International Journal of Heat and Mass Transfer**, v. 51, n. 5–6, p. 1238–1250, 2008.

TSAI, C.; PUI, D. Y. H. Numerical study of particle deposition in bends of a circular cross-section-laminar flow regime. **Aerosol Science and Technology**, v. 12, p. 813–331, 1990.

TU, J.; INTHAVONG, K.; AHMADI, G. **Computational Fluid and Particle Dynamics in the Human Respiratory System**. New York: Springer International Publishing, 2013.

TU, J. Y.; FLETCHER, C. A. J. Numerical computation of turbulent gas-solid particle flow in a 90° bend. **A.I.Ch.E. Journal**, v. 41, n. 10, p. 2187–2197, 1995.

UIJTTEWAAL, W. S. J.; OLIEMANS, R. V. A. Particle dispersion and deposition in direct numerical and large eddy simulations of vertical pipe flows. **Physics of Fluids**, v. 8, n. 10, p. 2590–2604, 1996.

VAN DOORMAAL, J. P.; RAITHBY, G. D. Enhancements of the simple method for predicting incompressible fluid flows. **Numerical Heat Transfer: An International Journal of Computation and Methodology**, v. 7, n. 2, p. 147–163, 1984.

VERSTEEG, H. K.; MALALASEKERA, W. **An Introduction to Computational Fluid Dynamics: The Finite Volume Method**. England: Pearson Education Limited, 2007.

WALLIN, S.; JOHANSSON, A. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. **Journal of Fluid Mechanics**, p. 89–132, 2000.

WANG, Q.; SQUIRES, K.D.; CHEN, M.; MCLAUGHLIN, J.B. On the role of the lift force in turbulence simulations of particle deposition. **International Journal of Multiphase Flow**, v. 23, n. 4, p. 749–763, 1997.

WHITE, F. **Fluid Mechanics**. New York: McGraw-Hill, 2010.

WILCOX, D. C. Reassessment of the scale-determining equation for advanced turbulence models. **AIAA Journal**, v. 26, n. 11, p. 1299–1310, 1988.

WOLFSHTEIN, M. The velocity and temperature distribution in one-dimensional flow with turbulence augmentation and pressure gradient. **International Journal of Heat and Mass Transfer**, v. 12, p. 301–318, 1969.

WOOD, N. B. A simple method for the calculation of turbulent deposition to smooth and rough surfaces. **Journal of Aerosol Science**, v. 12, n. 3, p. 275–290, 1981.

WU, J.; ZHAO, B. Effect of ventilation duct as a particle filter. **Building and Environment**, v. 42, n. 7, p. 2523–2529, 2007.

YAKHOT, V.; ORSZAG, S.A.; THANGAM, S.; GATSKI, T.B.; SPEZIALE, C.G. Development of turbulence models for shear flows by a double expansion technique. **Physics of Fluids**, v. 4, n. 7, p. 1510–1520, 1992.

YANG, W.; KUAN, B. Experimental investigation of dilute turbulent particulate flow inside a curved 90° bend. **Chemical Engineering Science**, v. 61, n. 11, p. 3593–3601, 2006.

YIN, Z. Q.; LIN, J. Z.; LOU, M. Penetration efficiency of nanoparticles in a bend of circular cross-section. **Journal of Hydrodynamics**, v. 27, n. 1, p. 93–98, 2015.

ZAICHIK, L. I.; DROBYSHEVSKY, N.I.; FILIPPOV, A.S.; MUKIN, R.V.; STRIZHOV, V.F. A diffusion-inertia model for predicting dispersion and deposition of low-inertia particles in turbulent flows. **International Journal of Heat and Mass Transfer**, v. 53, n. 1–3, p. 154–162, 2010.

ZAMANI, M.; SEDDIGHI, S.; NAZIF, H. R. Erosion of natural gas elbows due to rotating particles in turbulent gas-solid flow. **Journal of Natural Gas Science and Engineering**, v. 40, p. 91–113, 2017.

ZENG, L.; NAJJAR, F.; BALACHANDAR, S.; FISCHER, P. Forces on a finite-sized particle located close to a wall in a linear shear flow. **Physics of Fluids**, v. 21, n. 3, p. 033302, mar. 2009.

ZHANG, J.; LI, A.; LI, D. Modeling deposition of particles in typical horizontal ventilation duct flows. **Energy Conversion and Management**, v. 49, p. 3672–3683, 2008.

ZHANG, P.; ROBERTS, R. M.; BÉNARD, A. Computational guidelines and an empirical model for particle deposition in curved pipes using an Eulerian-Lagrangian approach. **Journal of Aerosol Science**, v. 53, p. 1–20, 2012.

ZHANG, Z.; CHEN, Q. Prediction of particle deposition onto indoor surfaces by CFD with a modified Lagrangian method. **Atmospheric Environment**, v. 43, n. 2, p. 319–328, 2009.

ZHAO, B.; CHEN, J. Numerical analysis of particle deposition in ventilation duct. **Building and Environment**, v. 41, n. 6, p. 710–718, 2006.

ZHAO, B.; WU, J. Modeling particle deposition from fully developed turbulent flow in ventilation duct. **Atmospheric Environment**, v. 40, n. 3, p. 457–466, 2006.

APPENDICES

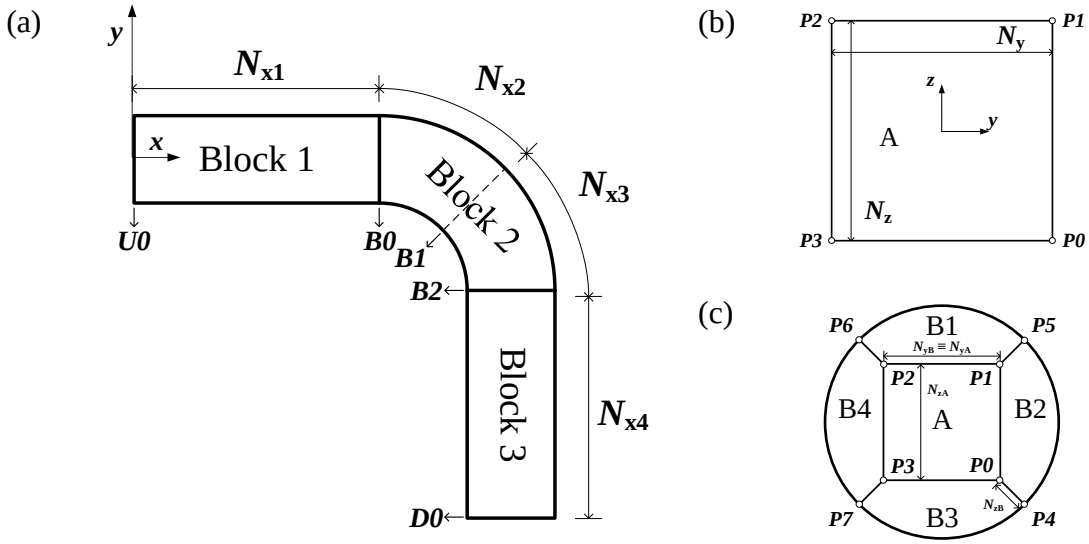
APPENDIX A OpenFOAM® guideline for the grid generation

The grid generation with ***blockMesh*** of the OpenFOAM® v6.

All meshes of this study were generated with the *blockMesh* (GREENSHIELDS, 2018), the OpenFOAM® standard mesh generator. The third study case of this manuscript was presented in Section 5.3 and *Grid 2* will be the example to explain how this grid generator works (see Section 5.3.1 for geometry information). The other studies of Chapter 5 followed the same methodology of grid generation[28].

First, the block grid strategy must be defined according to the geometry and boundary conditions such as illustrated in Figure A.1. Block 1 and 3 represent the upstream and downstream straight ducts, respectively. Block 2 (the elbow) must be split into two parts because of the limitations of *blockMesh* to represent better the curved geometry. Therefore, we have four blocks instead of three.

Figure A.1 – Block meshing strategy: (a) top view, (b) square duct and (c) circular duct.



To generate the Block 1 we need to inform the points of the faces *U0* and *B0* (Figure A.1a) as well as the points of the faces *B2* and *D0* to generate the Block 3. To generate the first part of Block 2 we need the points of the faces *B0* and *B1*, while the second part needs the points of the faces *B1* and *B2*. Each face (*U0* to *D0*) depends on the cross-section. The points of the square cross-section (Figure A.1b) are *P0* to *P3*, while the circular cross-section (Figure A.1c) is *P0* to *P7* because of the O-grid approach.

The file *blockMeshDict.C* of the square and the circular duct are presented in Figure A.2 and Figure A.3, respectively, with all information given in Figure A.1. With the *blockMesh* command line, we can generate the grid (to display the grid we need to open the mesh with the Paraview®). Once the mesh is generated, the mesh quality must be measured with the *checkMesh* command line.

Two important tips are: to determine the distance from the wall to the first node we find online tools (e.g., https://www.cfd-online.com/Tools/yplus.php - access 13/11/2019), and to calculate the grid wall-normal growth rates we also find them (e.g., https://openfoamwiki.net/index.php/Scripts/blockMesh_grading_calculation - access 13/11/2019).

[28] We need only to change the vertices values of *blockMeshDict.C* file for the respective study case.

Figure A.2 – Square duct: *blocMeshDict.C*.

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

/*----------------------*- SQURE DUCT (FIG. A1b) -*----------------------*\

convertToMeters 0.001;

vertices
(
// Plane U0 (domain inlet)
    (0. 40. -40.)                       // 0  (P0)  r*=1 (outer wall)
    (0. 40. 40.)                        // 1  (P1)  r*=1
    (0. -40. 40.)                       // 2  (P2)  r*=0 (inner wall)
    (0. -40. -40.)                      // 3  (P3)  r*=0
// Plane B0 (bend inlet - 0 deg.)
    (4000. 40. -40.)                    // 4  (P0)
    (4000. 40. 40.)                     // 5  (P1)
    (4000. -40. 40.)                    // 6  (P2)
    (4000. -40. -40.)                   // 7  (P3)
// Plane B1 - (45 deg.)
    (4141.4214 -18.5786 -40.)           // 8  (P0)
    (4141.4214 -18.5786 40.)            // 9  (P1)
    (4084.8528 -75.1472 40.)            // 10 (P2)
    (4084.8528 -75.1472 -40.)           // 11 (P3)
// Plane B2 (bend outlet - 90 deg.)
    (4200. -160. -40.)                  // 12 (P0)
    (4200. -160. 40.)                   // 13 (P2)
    (4120. -160. 40.)                   // 14 (P3)
    (4120. -160. -40.)                  // 15 (P4)
// Plane D0 (domain outlet)
    (4200. -4160. -40.)                 // 16 (P0)
    (4200. -4160. 40.)                  // 17 (P1)
    (4120. -4160. 40.)                  // 18 (P2)
    (4120. -4160. -40.)                 // 19 (P3)
);

blocks
(
      // Block 1 (U0 to B0 - the upstream duct)
      hex (0 1 2 3 4 5 6 7) upstream (30 30 200) // (Nz Ny Nx1)
      simpleGrading
      (
            ((.5 .5 500.)
            (.5 .5 .002))     // z  (spanwise dir.)
            ((.5 .5 500.)
```

```
            (.5 .5 .002))     // y  (radial dir.)
            .05               // x  (streamwise dir.)
      )
      // Block 2 (B0 to B1 - 1st part)
      hex (4 5 6 7 8 9 10 11) elbow (30 30 40) // (Nz Ny Nx2)
      simpleGrading
      (
            ((.5 .5 500.)
            (.5 .5 .002))     // z  (spanwise dir.)
            ((.5 .5 500.)
            (.5 .5 .002))     // y  (radial dir.)
            1.                // x  (streamwise dir.)
      )
      // Block 2 (B1 to B2 - 2nd part)
      hex (8 9 10 11 12 13 14 15) elbow (30 30 40) // (Nz Ny Nx3)
      (
            ((.5 .5 500.)
            (.5 .5 .002))     // z  (spanwise dir.)
            ((.5 .5 500.)
            (.5 .5 .002))     // y  (radial dir.)
            1.                // x  (streamwise dir.)
      )
      // Block 3 (B2 to D0 - the downstream duct)
      hex (12 13 14 15 16 17 18 19) downstream (30 30 200) // (Nz Ny Nx4)
      simpleGrading
      (
            ((.5 .5 500.)
            (.5 .5 .002))     // z  (spanwise dir.)
            ((.5 .5 500.)
            (.5 .5 .002))     // y  (radial dir.)
            20.               // x  (streamwise dir.)
      )
);

edges
(
      // Curved lines from B0 to B1 (interpolation point at 22.5 deg.)
      arc 4 8 (4076.5367 24.7759 -40.)          // r*=1 (bend outer wall)
      arc 5 9 (4076.5367 24.7759 40.)           // r*=1
      arc 6 10 (4045.9220 -49.134456 40.)       // r*=0 (bend inner wall)
      arc 7 11 (4045.9220 -49.134456 -40.)      // r*=0
      // Curved lines from B1 to B2 (interpolation point at 67.5 deg.)
      arc 8 12 (4184.7759 -83.4633 -40.)        // r*=1
      arc 9 13 (4184.7759 -83.4633 40.)         // r*=1
      arc 10 14 (4110.8655 -114.0780 40.)       // r*=0
      arc 11 15 (4110.8655 -114.0790 -40.)      // r*=0
);

boundary
(
    wall
    {
        type wall;
        faces
        (
        // Block 1 (upstream duct)
        (0 1 5 4)
        (1 2 6 5)
        (2 3 7 6)
        (3 0 4 7)
        // Block 2 (elbow - 1st part)
```

```
120             (4 5 9 8)
121             (5 6 10 9)
122             (6 7 11 10)
123              (7 4 8 11)
124             // Block 2 (elbow - 2nd part)
125             (8 9 13 12)
126             (9 10 14 13)
127             (10 11 15 14)
128             (11 8 12 15)
129             // Block 3 (downstream duct)
130             (12 13 17 16)
131             (13 14 18 17)
132             (14 15 19 18)
133             (15 12 16 19)
134             );
135         }
136         inlet
137         {
138             type patch;
139             faces
140             ((0 1 2 3));
141         }
142         outlet
143         {
144             type patch;
145             faces
146             ((16 17 18 19));
147         }
148     );
149
150     //************************************************************************//
```

Figure A.3 – Circular duct: *blockMeshDict.C.*

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------
--*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

/*--------------------*- CIRCULAR DUCT (FIG. A1c) -*---------------------*\

convertToMeters 0.001;

vertices
(
// Plane U0 (domain inlet)
    (0. 20. -20.)                       // 0  (P0)  r*=3/4
    (0. 20. 20.)                        // 1  (P1)  r*=3/4
    (0. -20. 20.)                       // 2  (P2)  r*=1/4
    (0. -20. -20.)                      // 3  (P3)  r*=1/4
    (0. 28.2843 -28.2843)               // 4  (P4)  r*=6/7 (outer wall)
    (0. 28.2843 28.2843)                // 5  (P5)  r*=6/7
    (0. -28.2843 28.2843)               // 6  (P6)  r*=1/7 (inner wall)
    (0. -28.2843 -28.2843)              // 7  (P7)  r*=1/7
// Plane B0 (bend inlet – 0 deg.)
    (4000. 20. -20.)                    // 8  (P0)
    (4000. 20. 20.)                     // 9  (P1)
    (4000. -20. 20.)                    // 10 (P2)
    (4000. -20. -20.)                   // 11 (P3)
    (4000. 28.2843 -28.2843)            // 12 (P4)
    (4000. 28.2843 28.2843)             // 13 (P5)
    (4000. -28.2843 28.2843)            // 14 (P6)
    (4000. -28.2843 -28.2843)           // 15 (P7)
// Plane B1 – (45 deg.)
    (4127.2792 -32.7208 -20.)           // 16 (P0)
    (4127.2792 -32.7208 20.)            // 17 (P1)
    (4098.995 -61.005 20.)              // 18 (P2)
    (4098.995 -61.005 -20.)             // 19 (P3)
    (4133.1371 -26.8629 -28.2843)       // 20 (P4)
    (4133.1371 -26.8629 28.2843)        // 21 (P5)
    (4093.1371 -66.8629 28.28427)       // 22 (P6)
    (4093.1371 -66.8629 -28.28427)      // 23 (P7)
// Plane B2 (bend outlet – 90 deg.)
    (4180. -160. -20.)                  // 24 (P0)
    (4180. -160 20.)                    // 25 (P1)
    (4140. -160. 20.)                   // 26 (P2)
    (4140. -160. -20.)                  // 27 (P3)
    (4188.2843 -160. -28.2843)          // 28 (P4)
    (4188.2843 -160. 28.2843)           // 29 (P5)
    (4131.7157 -160. 28.2843)           // 30 (P6)
    (4131.7157 -160. -28.2843)          // 31 (P7)
```

```
// Plane D0 (domain outlet)
    (4180. -4160. -20.)                  // 32 (P0)
    (4180. -4160 20.)                    // 33 (P1)
    (4140. -4160. 20.)                   // 34 (P2)
    (4140. -4160. -20.)                  // 35 (P3)
    (4188.2843 -4160. -28.2843)          // 36 (P4)
    (4188.2843 -4160. 28.2843)           // 37 (P5)
    (4131.7157 -4160. 28.2843)           // 38 (P6)
    (4131.7157 -4160. -28.2843)          // 39 (P7)
);

blocks
(
// Block 1 (U0 to B0 – the upstream duct)
// A (O-grid center) - (NzA NyA Nx1)
hex (0 1 2 3 8 9 10 11) upstream (13 13 200) simpleGrading (1. 1. .05)
// B1 (O-grid north) - (NzB NyB Nx1)
hex (6 2 1 5 14 10 9 13) upstream (16 13 200) simpleGrading (90. 1. .05)
// B2 (O-grid east) - (NzB NyB Nx1)
hex (5 1 0 4 13 9 8 12) upstream (16 13 200) simpleGrading (90. 1. .05)
// B3 (O-grid south) - (NzB NyB Nx1)
hex (4 0 3 7 12 8 11 15) upstream (16 13 200) simpleGrading (90. 1. .05)
// B4 (O-grid west) - (NzB NyB Nx1)
hex (7 3 2 6 15 11 10 14) upstream (16 13 200) simpleGrading (90. 1. .05)

// Block 2 (B0 to B1 – 1st part)
// A (O-grid center) - (NzA NyA Nx2)
hex (8 9 10 11 16 17 18 19) elbow (13 13 40) simpleGrading (1. 1. 1.)
// B1 (O-grid north) - (NzB NyB Nx2)
hex (14 10 9 13 22 18 17 21) elbow (16 13 40) simpleGrading (90. 1. 1.)
// B2 (O-grid east) - (NzB NyB Nx2)
hex (13 9 8 12 21 17 16 20) elbow (16 13 40) simpleGrading (90. 1. 1.)
// B3 (O-grid south) - (NzB NyB Nx2)
hex (12 8 11 15 20 16 19 23) elbow (16 13 40) simpleGrading (90. 1. 1.)
// B4 (O-grid west) - (NzB NyB Nx2)
hex (15 11 10 14 23 19 18 22) elbow (16 13 40) simpleGrading (90. 1. 1.)

// Block 2 (B1 to B2 – 2nd part)
// A (O-grid center) - (NzA NyA Nx3)
hex (16 17 18 19 24 25 26 27) elbow (13 13 40) simpleGrading (1. 1. 1.)
// B1 (O-grid north) - (NzB NyB Nx3)
hex (22 18 17 21 30 26 25 29) elbow (16 13 40) simpleGrading (90. 1. 1.)
// B2 (O-grid east) - (NzB NyB Nx3)
hex (21 17 16 20 29 25 24 28) elbow (16 13 40) simpleGrading (90. 1. 1.)
// B3 (O-grid south) - (NzB NyB Nx3)
hex (20 16 19 23 28 24 27 31) elbow (16 13 40) simpleGrading (90. 1. 1.)
// B4 (O-grid west) - (NzB NyB Nx3)
hex (23 19 18 22 31 27 26 30) elbow (16 13 40) simpleGrading (90. 1. 1.)

// Block 3 (B2 to D0 – the downstream duct)
// A (O-grid center) - (NzA NyA Nx4)
hex (24 25 26 27 32 33 34 35) downstream (13 13 200) simpleGrading (1. 1.
20.)
// B1 (O-grid north) - (NzB NyB Nx4)
hex (30 26 25 29 38 34 33 37) downstream (16 13 200) simpleGrading (90. 1.
20.)
// B2 (O-grid east) - (NzB NyB Nx4)
hex (29 25 24 28 37 33 32 36) downstream (16 13 200) simpleGrading (90. 1.
20.)
// B3 (O-grid south) - (NzB NyB Nx4)
```

```
hex (28 24 27 31 36 32 35 39) downstream (16 13 200) simpleGrading (90. 1.
20.)
// B4 (O-grid west) - (NzB NyB Nx4)
hex (31 27 26 30 39 35 34 38) downstream (16 13 200) simpleGrading (90. 1.
20.)
);

edges
(
// Curvature of the cross section
// Plane U0
arc 0 1 (0. 22. 0.)                    // P0 to P1 - r*=3/4 + 10%/4
arc 1 2 (0. 0. 22.)                    // P1 to P2 - r*=1/2 + 10%/4
arc 2 3 (0. -22. 0.)                   // P2 to P3 - r*=1/4 - 10%/4
arc 3 0 (0. 0. -22.)                   // P3 to P0 - r*=1/2 - 10%/4
arc 4 5 (0. 40. 0.)                    // P4 to P5 - r*=1
arc 5 6 (0. 0. 40.)                    // P5 to P6 - r*=1/2
arc 6 7 (0. -40. 0.)                   // P6 to P7 - r*=0
arc 7 4 (0. 0. -40.)                   // P7 to P4 - r*=1/2
// Plane B0
arc 8 9 (4000. 22. 0.)                 // P0 to P1
arc 9 10 (4000. 0. 22.)                // P1 to P2
arc 10 11 (4000. -22. 0.)              // P2 to P3
arc 11 8 (4000. 0. -22.)               // P3 to P0
arc 12 13 (4000. 40. .0)               // P4 to P5
arc 13 14 (4000. 0. 40.)               // P5 to P6
arc 14 15 (4000. -40. 0.)              // P6 to P7
arc 15 12 (4000. 0. -40.)              // P7 to P4
// Plane B1
arc 16 17 (4128.6934 -31.3066 0.)      // P0 to P1
arc 17 18 (4113.1371 -46.8629 22.)     // P1 to P2
arc 18 19 (4097.5807 -62.4193 0.)      // P2 to P3
arc 19 16 (4113.1371 -46.8629 -22.)    // P3 to P0
arc 20 21 (4141.4214 -18.5786 0.)      // P4 to P5
arc 21 22 (4113.1371 -46.8629 40.)     // P5 to P6
arc 22 23 (4084.8528 -75.1472 0.)      // P6 to P7
arc 23 20 (4113.1371 -46.8629 -40.)    // P7 to P4
// Plano B2
arc 24 25 (4182. -160. 0.)             // P0 to P1
arc 25 26 (4160. -160. 22.)            // P1 to P2
arc 26 27 (4138. -160. 0.)             // P2 to P3
arc 27 24 (4160. -160. -22.)           // P3 to P0
arc 28 29 (4200. -160. 0.)             // P4 to P5
arc 29 30 (4160. -160. 40.)            // P5 to P6
arc 30 31 (4120. -160. 0.)             // P6 to P7
arc 31 28 (4160. -160. -40.)           // P7 to P4
// Plane D0
arc 32 33 (4182. -4160. 0.)            // P0 to P1
arc 33 34 (4160. -4160. 22.)           // P1 to P2
arc 34 35 (4138. -4160. 0.)            // P2 to P3
arc 35 32 (4160. -4160. -22.)          // P3 to P0
arc 36 37 (4200. -4160. 0.)            // P4 to P5
arc 37 38 (4160. -4160. 40.)           // P5 to P6
arc 38 39 (4120. -4160. 0.)            // P6 to P7
arc 39 36 (4160. -4160. -40.)          // P7 to P4

// Curvature of the elbow
// Curved lines from B0 to B1 (interpolation point at 22.5 deg.)
arc 8 16 (4068.8830 6.2983 -20.)       // r*=3/4
arc 9 17 (4068.8830 6.2983 20.)        // r*=3/4
arc 10 18 (4053.5756 -30.6569 20.)     // r*=1/4
```

```
arc 11 19 (4053.5756 -30.6569 -20.)       // r*=1/4
arc 12 20 (4072.0533 13.9520 -28.2843)    // r*=6/7
arc 13 21 (4072.0533 13.9520 28.2843)     // r*=6/7
arc 14 22 (4050.4054 -38.3105 28.2843)    // r*=1/7
arc 15 23 (4050.4054 -38.3105 -28.2843)   // r*=1/7
// Curved lines from B1 to B2 (interpolation point at 67.5 deg.)
arc 16 24 (4166.2983 -91.1170 -20.)       // r*=3/4
arc 17 25 (4166.2983 -91.1170 20.)        // r*=3/4
arc 18 26 (4129.3431 -106.4243 20.)       // r*=1/4
arc 19 27 (4129.3431 -106.4243 -20.)      // r*=1/4
arc 20 28 (4173.9520 -87.9467 -28.2843)   // r*=6/7
arc 21 29 (4173.9520 -87.9467 28.2843)    // r*=6/7
arc 22 30 (4121.6895 -109.5946 28.2843)   // r*=1/7
arc 23 31 (4121.6895 -109.5946 -28.28427) // r*=1/7
);

boundary
(
    wall
    {
        type wall;
        faces
        (
        // Block 1 (upstream duct)
            (4 5 13 12)
            (5 6 14 13)
            (6 7 15 14)
            (7 4 12 15)
        // Block 2 (elbow - 1st part)
            (12 13 21 20)
            (13 14 22 21)
            (14 15 23 22)
            (15 12 20 23)
        // Block 2 (elbow - 2nd part)
            (20 21 29 28)
            (21 22 30 29)
            (22 23 31 30)
            (23 20 28 31)
        // Block 3 (downstream duct)
            (28 29 37 36)
            (29 30 38 37)
            (30 31 39 38)
            (31 28 36 39)
        );
    }

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

    outlet
    {
        type patch;
        faces
```

```
242             ((32 33 34 35)
243              (38 34 33 37)
244              (37 33 32 36)
245              (36 32 35 39)
246              (39 35 34 38));
247         }
248 );
249
250 //*********************************************************************//
```

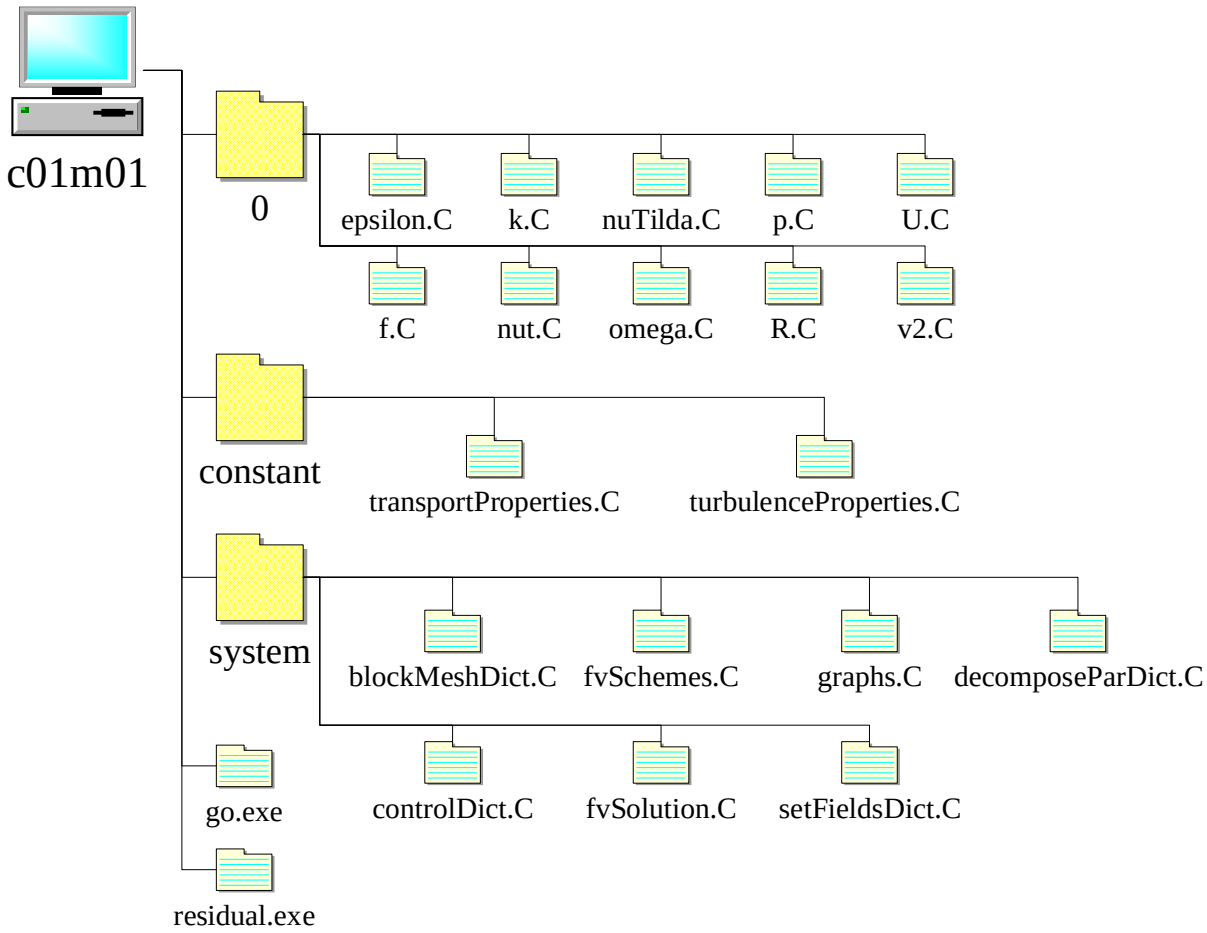
APPENDIX B OpenFOAM® guideline for the Eulerian solver

The ***pimpleFoam*** solver of the OpenFOAM® v6.

The files of a typical incompressible flow run with the OpenFOAM® code, e.g., for the case directory *c01m01* regarding the case "c01" solved with the mesh "m01", is schematized in Figure A.4. The folder *0* must contain the boundary (indicated in the *blockMeshDict.C* as shown in Figure A.2 and Figure A.3) and the initial conditions (Section 5.1.1 presented how to estimate the turbulent properties for this study) such as the velocity and the pressure[29] in files *U.C* and *p.C*, respectively. No further information about those files will be presented because no significant changing was done in this study, however, if we need a complex boundary and/or initial conditions a good tip is to code with *codeStream* (GREENSHIELDS, 2018), e.g., to set an inlet turbulent profile.

The folder *constant* (Figure A.4) contains the *transportProperties.C* file in which, in this study, only the kinematic viscosity was informed. The respective turbulence model (see Section 4.1.1 to consult the turbulence models applied in this study) was selected in the *turbulenceProperties.C* file. Once again, no further information about these files will be given, to choose the appropriate turbulence models see GREENSHIELDS (2018) and the installation files. In addition, the folder *constant* will contain the folder *polyMesh* (not presented) after the grid generation with all mesh data.

Figure A.4 – Fluid-phase: typical run with the *pimpleFoam* solver.



[29] For incompressible solvers in OpenFOAM® the pressure is modified as: $p^* = p/\rho$.

The folder *system* (Figure A.4) contains important numerical information such as the *blockMeshDict.C* (see Figure A.2 and Figure A.3) and the *controlDict.C* to set, respectively, the grid and the run solver control (such as the time step and convergence criterion). For parallel running, we need to set the *decomposeParDict.C* to select the domain decomposition method and to determine the number of subdomains (we can apply the *setFieldsDict.C* to control the domain decomposition with the *manual* method).

The finite volume discretization schemes in the numerical solution are defined in the *fvSchemes.C* file of the *system* folder (Figure A.4), whereas the linear equations solver, the tolerances and others algorithm controls are made in the *fvSolution.C* file. The correct choice of the discretization and algorithm controls play an important role in the convergence. In this study, we started with a first-order scheme (the *upwind* scheme) and then change it for a second-order scheme (the *linearUpwindScheme* of BEAM and WARMING (1976)). Regarding the linear equation solver, we applied the *GAMG* solver for the pressure and *PBiCGStab* solver for other variables.

In addition, we suggest launching the run from the file *go.exe* (Figure A.4) with the command line *./go.exe &*. This file can simply contain the following:

```
mpirun -np 4 pimpleFoam_new -parallel > log 2> error
```

,i.e., it will launch the *pimpleFoam_new* solver using parallel computing with *4* processors and the convergence history will be safe at the file *log* and any eventual error that can stop the computation in the file *error*[30].

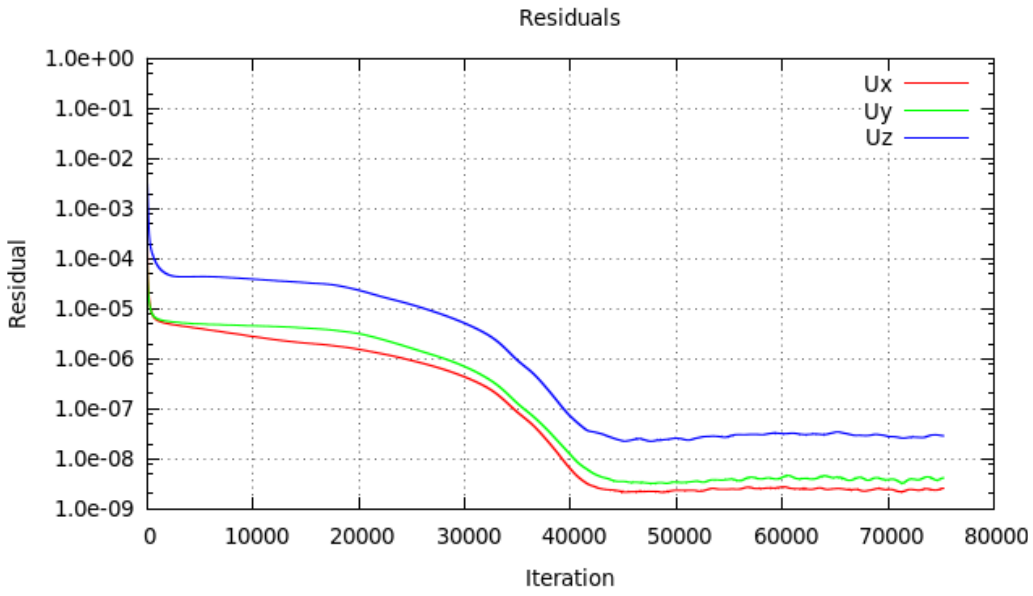
Once the computations were launched with the *go.exe* file, we can see the convergence using Gnuplot®. In this case, we must set the file *residual.exe* with, e.g., Figure A.5. The file can be read with the command line: *gnuplot residual.exe*. This yields a convergence curve such as shown in Figure A.6. The convergence regarding the other variables can be obtained by changing the variables of velocity components indicated in Figure A.5 for the respective variable, e.g., *p* and *k* for the convergence of the pressure and turbulent kinetic energy, respectively.

Figure A.5 – The *residual.exe* file.

```
set logscale y
set title "Residuals"
set format y "%.1e"; set ylabel 'Residual'
set xlabel 'Iteration'
plot "< cat log | grep 'Solving for Ux' | cut -d' ' -f9 | tr -
d ','" title 'Ux' with lines,\
     "< cat log | grep 'Solving for Uy' | cut -d' ' -f9 | tr -
d ','" title 'Uy' with lines,\
     "< cat log | grep 'Solving for Uz' | cut -d' ' -f9 | tr -
d ','" title 'Uz' with lines
pause 1
set grid
reread
```

[30] Before launching the *go.exe* file using parallel computing we must to create the *processor** folders with the command line: *decomposePar*.

Figure A.6 – Example of a convergence history plotted with the *residual.exe* file.



Another good convergence indicator can be the total kinetic energy, U_energy, and the "total pressure energy", p_energy, defined as:

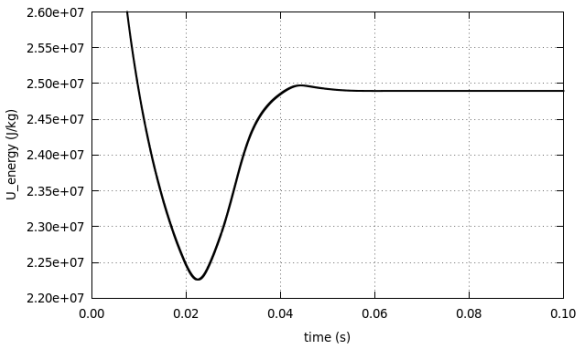
$$U\_energy = \sum_{i=1}^{N} \left(\frac{{U_i}^2}{2}\right), \quad \text{(A-1)}$$

$$p\_energy = \sum_{i=1}^{N} \left(\frac{{p_i}^2}{2}\right), \quad \text{(A-2)}$$

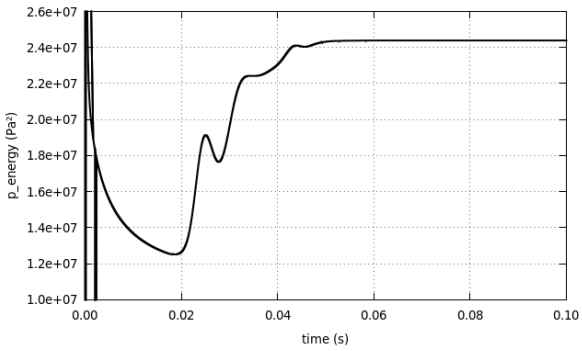
where N is the total grid cell number, U_i is the velocity and p_i is the pressure of the volume i.

Figure A.7 illustrated an example of the convergence observed with U_energy and p_energy. From 0.05s there is no significant variation of the U_energy (Figure A.7a) as well as the p_energy (Figure A.7b). Therefore, we concluded that from this given time the numerical solution achieved the convergence, i.e., no matter if the residuals of Figure A.6 tend to zero the solution must be the same and we have a converged solution.

Figure A.7 – Example of a convergence history with: (a) *U_energy* and (b) *p_energy*.



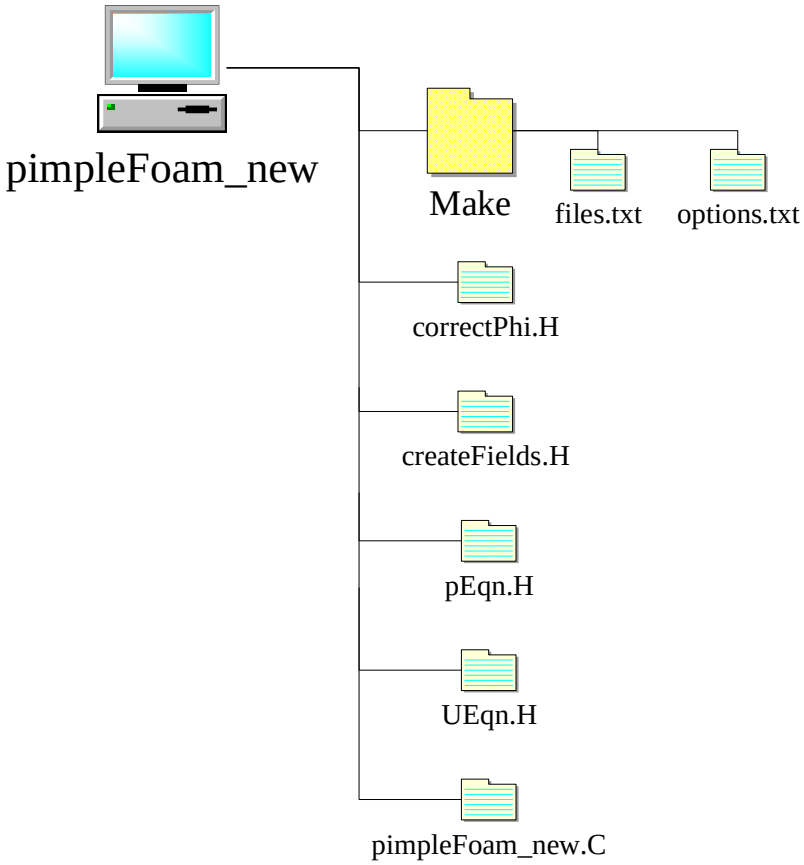
(a) *U_energy*.



(b) *p_energy*.

Nevertheless, to verify the convergence with *U_energy* and *p_energy* we need first to change the original solver. Figure A.8 shown the files of the *pimpleFoam_new* solver, i.e., the *pimpleFoam* solver with new convergence indicators.

Figure A.8 – Folder of the *pimpleFoam_new* solver.



The file *pimpleFoam_new.C* is presented in Figure A.9 in which the lines 67-69 and 132-149 were adding in the original *pimpleFoam* solver. In order to create the *pimpleFoam_new* we must be compiling[31] it with the *wmake* command line (GREENSHIELDS, 2018). Besides, before the compilation, only the *files.txt* file of the *Make* folder must be modified according to the following[32]:

```
pimpleFoam_new.C

EXE = $(FOAM_USER_APPBIN)/pimpleFoam_new
```

Notice the momentum equation is implemented in the file *UEqn.H* and the *pEqn.H* corrects the pressure and the velocities (Figure A.8). The *createFields.H* file declares the pressure and velocity fields, and the *correctPhi.H* files declare the flux corrections functions to ensure continuity. These files have not been modified, therefore they will not be present here.

[31] Never compile a new solver directly from the installation directory. Instead, copy and paste the source code to a local directory, e.g., *$FOAM_RUN/applications/solver/incompressible/pimpleFoam_new*, and then change and compile it.

[32] The file *options.txt* of the *Make* folder will be not change. However, this file is important because it calls all the libraries necessary to compile and run the solver.

Figure A.9 – The *pimpleFoam_new.C* file.

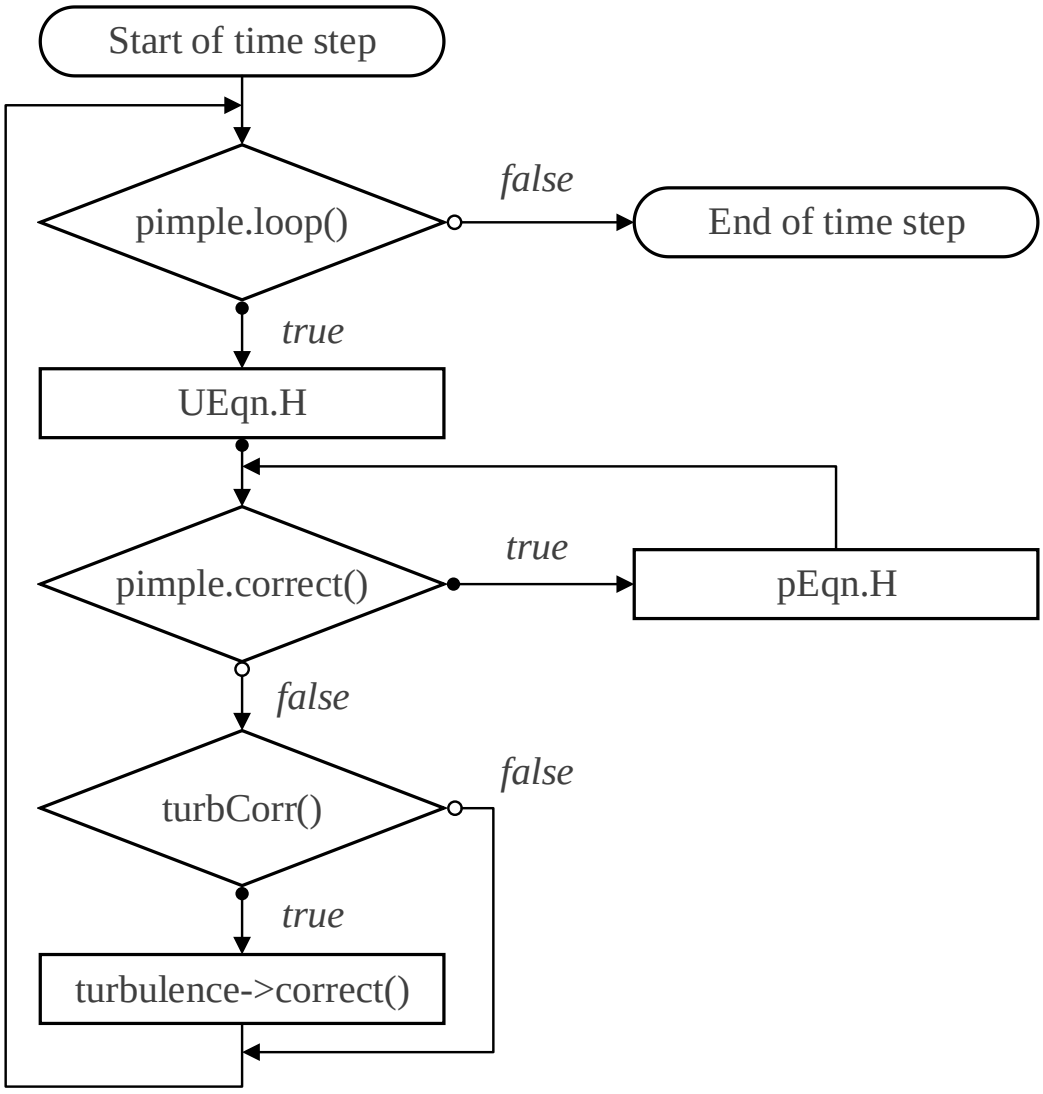
```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2011-2018 OpenFOAM 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/>.

Application
    pimpleDyMFoam.C

Description
    Transient solver for incompressible, turbulent flow of Newtonian
fluids, with optional mesh motion and mesh topology changes.

    Turbulence modelling is generic, i.e. laminar, RAS or LES may be
selected.

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

#include "fvCFD.H"
#include "dynamicFvMesh.H"
#include "singlePhaseTransportModel.H"
#include "turbulentTransportModel.H"
#include "pimpleControl.H"
#include "CorrectPhi.H"
#include "fvOptions.H"

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

int main(int argc, char *argv[])
{
    #include "postProcess.H"

    #include "setRootCaseLists.H"
    #include "createTime.H"
    #include "createDynamicFvMesh.H"
    #include "initContinuityErrs.H"
    #include "createDyMControls.H"
    #include "createFields.H"
    #include "createUfIfPresent.H"
    #include "CourantNo.H"
```

```
59      #include "setInitialDeltaT.H"
60
61      turbulence->validate();
62
63      // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
64
65      Info<< "\nStarting time loop\n" << endl;
66
67      // To create the file energy.txt
68      fileName outputFile("energy.txt");
69      OFstream os(outputFile);
70
71      while (runTime.run())
72      {
73          #include "readDyMControls.H"
74          #include "CourantNo.H"
75          #include "setDeltaT.H"
76
77          runTime++;
78
79          Info<< "Time = " << runTime.timeName() << nl << endl;
80
81          // --- Pressure-velocity PIMPLE corrector loop
82          while (pimple.loop())
83          {
84              if (pimple.firstIter() || moveMeshOuterCorrectors)
85              {
86                  mesh.update();
87
88                  if (mesh.changing())
89                  {
90                      MRF.update();
91
92                      if (correctPhi)
93                      {
94                          // Calculate absolute flux
95                          // from the mapped surface velocity
96                          phi = mesh.Sf() & Uf();
97
98                          #include "correctPhi.H"
99
100                         // Make the flux relative to the mesh motion
101                         fvc::makeRelative(phi, U);
102                     }
103
104                     if (checkMeshCourantNo)
105                     {
106                         #include "meshCourantNo.H"
107                     }
108                 }
109             }
110
111             #include "UEqn.H"
112
113             // --- Pressure corrector loop
114             while (pimple.correct())
115             {
116                 #include "pEqn.H"
117             }
118
119             if (pimple.turbCorr())
```

```
            {
                laminarTransport.correct();
                turbulence->correct();
            }
        }

        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;

      // To create the U_energy and P_energy
      int nbMaille=mesh.cells().size();
      int celli;
      double U_energy, p_energy;
            U_energy=0.;
            p_energy=0.;
            for (celli=0; celli<nbMaille; celli++)
                  {
                        U_energy += 0.5*(mag(U[celli])*mag(U[celli]));
                        p_energy += 0.5*(p[celli])*(p[celli]);

                  }
      Info<< "U_energy = " << U_energy << endl;
      Info<< "p_energy = " << p_energy << endl;

      // To save U_energy and p_energy in energy.txt
      os << runTime.timeName() << " " << U_energy << " " << p_energy ;
      os << "\n";
    }

    Info<< "End\n" << endl;

    return 0;
}


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

The *pimpleFoam* is a transient incompressible solver using the PIMPLE algorithm, which is a combination of the SIMPLE and the PISO algorithms. Figure A.10 shows the flow chart of the PIMPLE algorithm. The main loop of the *pimpleFoam* solver was presented in the lines 82-124 of Figure A.9. In this loop first the momentum equation is solver (line 111), then the pressure correction loop is entered (line 116). At the end of the main loop the turbulent equations are solved (line 122). The algorithm is executed every time step.

Figure A.10 – Flow chart of the PIMPLE algorithm.



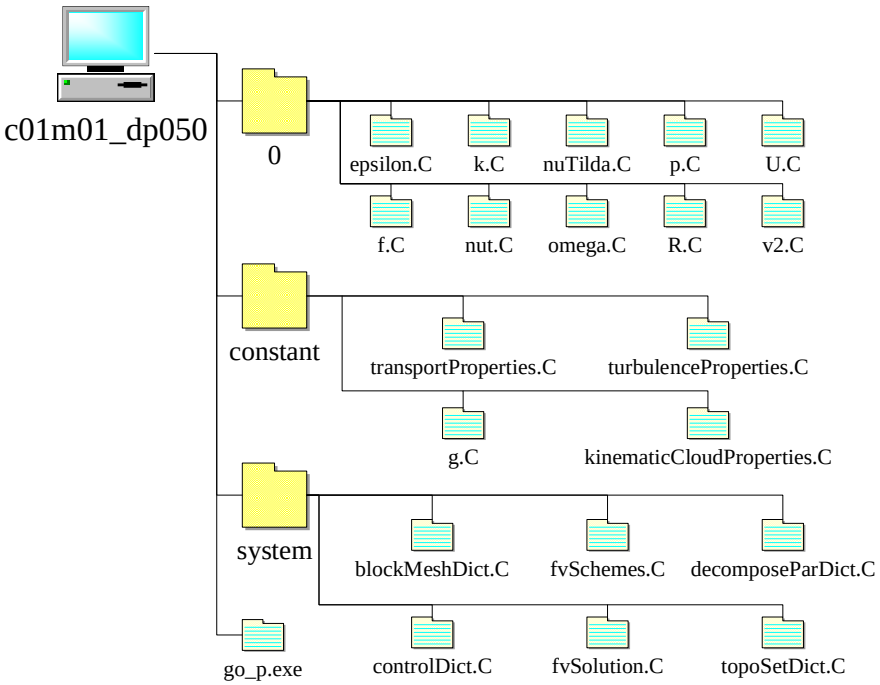
Source: adapted from HOLZINGER (2019).

APPENDIX C OpenFOAM® guideline for the Lagrangian solver

The ***icoUncoupledKinematicParcelFoam*** solver of the OpenFOAM® v6.

Section 5.2 we studied the application of several PWI models. Therefore, the LPT solver must be changed for this purpose and it will be presented here. In this study first, we solved the fluid-phase with the PIMPLE algorithm as in APPENDIX B. Then, we injected and tracked the particulate-phase in the airflow using one-way coupling. The files of a typical run with the LPT solver of OpenFOAM®, the *icoUncoupledKinematicParcelFoam* solver, are presented in Figure A.11 for the case *c01m01* and particles with 50 μm, i.e., the case *c01m01_dp050*.

Figure A.11 – Particulate-phase: typical run with the *icoUncoupledKinematicParcelFoam* solver.



Firstly, we copy and paste the flow solution of the *pimpleFoam* solver in folder *0* of Figure A.11. Second, we build the mesh with the *blockMeshDict.C* in the *system* folder with the same grid of the flow solution (the *fvSchemes.C* and the *fvSolution.C* are the same as the *pimpleFoam*). Next, we set the time step (based on the particle relaxation time and minimum grid size) and the end time (duration of the particles to cross the elbow) in the *controlDict.C* file. Then, we build the *cellZone* named *injection*[33] with the *topoSet* command line as shown in *topoSetDict.C* file of Figure A.12. Lastly, we input the gravity in the *g.C* file of the *constant* folder[34] and we set the *kinematicCloudProperties.C* file as shown in Figure A.13.

In addition, we suggest launching the run from the file *go_p.exe* (Figure A.11) with the command line *./go_p.exe &*. This file can simply contain the following:

```
mpirun -np 4 icoUncoupledKinematicParcelFoam_new_3Dsym -parallel > log 2> error
```

,i.e., it will launch the *icoUncoupledKinematicParcelFoam_new_3Dsym* solver using parallel computing with *4* processors and the convergence history will be safe at the file *log* and any eventual error that can stop the computation in the file *error*.

[33] This region is a box located close to the bend inlet.
[34] The turbulence model must be the same as the flow solution (*turbulenceProperties.C*) and the fluid density must be specified in the *transportProperties.C* file as well as the fluid viscosity.

Figure A.12 – The *topoSetDict.C* file.

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      topoSetDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

actions
(
   {
        name    injectionRegion;
        type    cellSet;
        action  new;
        source  boxToCell;
        sourceInfo
        {
            // box (MINX MINY MINZ) (MAXX MAXY MAXZ);
            box (0.50 -0.04 0.01) (0.60 0.04 0.04);
        }
    }
    {
        name    injection;
        type    cellZoneSet;
        action  new;
        source  setToCellZone;
        sourceInfo
        {
            set injectionRegion;
        }
    }
);

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

Figure A.13 – The *kinematicCloudProperties.C* file.

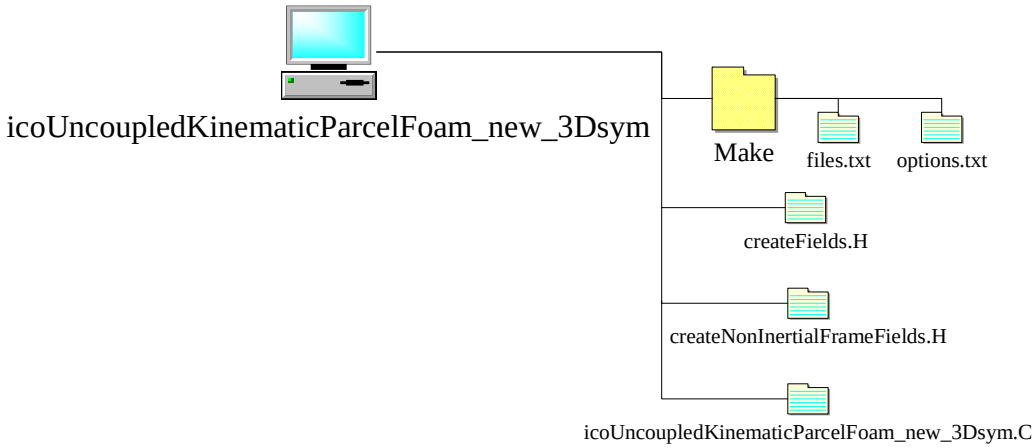
```
/*--------------------------------*- C++ -*----------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      particleProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//

solution
{
    active          true;
    coupled         false;
    transient       yes;
    cellValueSourceCorrection off;
    maxCo           0.3;

    interpolationSchemes
    {
        rho             cell;
        U               cellPoint;
        mu              cell;
        curlUcDt        cellPoint;
    }

    integrationSchemes
    {
        U               analytical; // Euler;
    }
}

constantProperties
{
    rho0            2990.; // particle density
}

subModels
{
    particleForces
    {
      gravity;
        sphereDrag;
        SaffmanMeiLiftForce
        {
            U     U;
            };
    }

    /*
    BrownianMotion
```

```
    {
            lambda      68.03E-9; // air mean free path
            turbulence  true;
    }
    */


    injectionModels
    {
        model1
        {
          type cellZoneInjection;
            cellZone injection;
            SOI 0.; // Start Of Injection
            parcelBasisType fixed;
            nParticle 1.;
        // numberDensity = total number of particles/volume of the cellZone
            numberDensity 1.2E9;
            duration 1.;

            U0 (52.19 0. 0.); // particle initial velocity
             massTotal  1.0e-7;

            /*
          type            patchInjection;
            parcelBasisType fixed;
            patchName       inlet;
            nParticle       1.0;

            U0 (52.19 0. 0.);
            flowRateProfile constant 1;
             massTotal 1.0e-7;

            SOI 0.;
            duration 0.01;
            parcelsPerSecond 3.0e7;
             */

         sizeDistribution
            {
                type        fixedValue;
                fixedValueDistribution
                {
                    value   50.0e-6; // particle diameter
                }
            }
      }
    }

    dispersionModel stochasticDispersionRAS;

    patchInteractionModel localInteraction;

    localInteractionCoeffs
    {
        patches
        (
            wall
            {
                type reboundA; // reboundB; reboundC; reboundD;
            }
```

```
            inlet
            {
            type escape;
            }
            outlet
            {
                  type escape;
            }
            symmetry
            {
                  type rebound;
            }
        );
    }

    surfaceFilmModel none;

    stochasticCollisionModel none;

    collisionModel none;

    }
}

cloudFunctions
{}

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

As presented in Section 5.2, to obtain the particle-phase mean streamwise velocity at a given duct position line and to include the PWI models we need to change the original LPT. Figure A.14 shows the files of the *icoUncoupledKinematicParcelFoam_new_3Dsym* solver, i.e., the *icoUncoupledKinematicParcelFoam* with new PWI models.

Figure A.14 – Folder of the *icoUncoupledKinematicParcelFoam_new_3Dsym* solver.



In order to compile the current LPT solver, the *files.txt* file of the *Make* folder (Figure A.14) must to be modified according to the following:

```
icoUncoupledKinematicParcelFoam_new_3Dsym.C

EXE = $(FOAM_USER_APPBIN)/icoUncoupledKinematicParcelFoam_new_3Dsym
```

And to add the new library features for the PWI models we must to change the *options.txt* file according to the following[35]:

```
EXE_INC = \
  ...
  -I$(WM_PROJECT_USER_DIR)/src/lagrangian/intermediate_new_3Dsym/lnInclude\
  ...

EXE_LIBS = \
  ...
  -L$(FOAM_USER_LIBBIN) \
  -llagrangianIntermediate_new_3Dsym \
  ...
```

The new lines included in the *icoUncoupledKinematicParcelFoam_new_3Dsym.C* (Figure A.15) are lines 66-78 and lines 92-165[36] (to create the lines and save the data into the created files for each iteration). Similarly, the *pimpleFoam_new* solver we must compile the new LPT solver with the *wmake* command line[37].

Notice the resulting files (see lines 66-78 of Figure A.15) will save the instantaneous particle velocity at each time step. However, these data need a statistical treatment such as illustrated in Figure A.16.

[35] Only the intermediate library must be changed, the other libraries remined the same.
[36] The lines 107-110 were saved to indicate the change of each time step.
[37] Before compile the LPT solver, the library with the PWI models must be compiled.

Figure A.15 – The *icoUncoupledKinematicParcelFoam_new_3Dsym* solver.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2011-2018 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/>.

Application
    icoUncoupledKinematicParcelFoam

Description
    Transient solver for the passive transport of a single kinematic
    particle cloud.

    Uses a pre-calculated velocity field to evolve the cloud.

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

#include "fvCFD.H"
#include "singlePhaseTransportModel.H"
#include "turbulentTransportModel.H"
#include "basicKinematicCollidingCloud.H"

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

//typedef typename basicKinematicCollidingCloud::particleType parcelType;
typedef basicKinematicCollidingCloud::particleType parcelType;

int main(int argc, char *argv[])
{
    argList::addOption
    (
        "cloudName",
        "name",
        "specify alternative cloud name. default is 'kinematicCloud'"
    );

    #include "postProcess.H"

    #include "setRootCaseLists.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControl.H"
```

```
    #include "createFields.H"

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

    Info<< "\nStarting time loop\n" << endl;

    #include "OFstream.H"

    fileName name1 = ("b00.txt");
    OFstream os1(name1);

    fileName name2 = ("b15.txt");
    OFstream os2(name2);

    fileName name3 = ("b30.txt");
    OFstream os3(name3);

    fileName name4 = ("b45.txt");
    OFstream os4(name4);

    while (runTime.loop())
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;

        Info<< "Evolving " << kinematicCloud.name() << endl;

        laminarTransport.correct();

        mu = laminarTransport.nu()*rhoInfValue;

        kinematicCloud.evolve();

      float xmin = 0.99995;         // x=(L1-1dp) (dp - particle diameter)
      float xmax = 1.;              // x=L1 (L1 - upstream duct length)

      float aB15Min = 0.260926723;  // bend angle 14.95deg
      float aB15Max = 0.262672052;  // bend angle 15.05deg

      float aB30Min = 0.522726111;  // bend angle 29.95deg
      float aB30Max = 0.524471440;  // bend angle 30.05deg

      float aB45Min = 0.784525498;  // bend angle 44.95deg
      float aB45Max = 0.786270828;  // bend angel 45.05deg

      float Rb = 0.176;             // bend radius
      float L1 = 1.;                // length of the upstream duct

        os1 << "1000000 1000000 1000000 1000000 1000000 1000000" << endl;
        os2 << "1000000 1000000 1000000 1000000 1000000 1000000" << endl;
        os3 << "1000000 1000000 1000000 1000000 1000000 1000000" << endl;
        os4 << "1000000 1000000 1000000 1000000 1000000 1000000" << endl;

      forAllConstIter(basicKinematicCollidingCloud, kinematicCloud, iter)
      {
            const parcelType& p = iter();

          // ab = 0deg (bend inlet)
          if((p.active()==1) && (p.position().component(0) >= xmin) &&
(p.position().component(0) <= xmax))
          {
```

```
            os1<<p.position().component(0)<<"
"<<p.position().component(1)<<" "<<p.position().component(2)<<"
"<<p.U().component(0)<<" "<<p.U().component(1)<<"
"<<p.U().component(2)<<endl;
          }

          // change of basis- Cartesian with bend center
          float x = p.position().component(0)-L1;
          float y = p.position().component(1)+Rb;

          if((p.active()==1) && (x >= 0) && (y >= 0))
          {
            // change of basis - polar coordinates
            float rr = pow(x,2.)+pow(y,2.);
            float r = pow(rr,0.5);
            float angle = y/r;
            float aB = acos(angle);

            // ab = 15deg
            if((aB >= aB15Min) && (aB <= aB15Max))
            {
                  os2<<p.position().component(0)<<"
"<<p.position().component(1)<<" "<<p.position().component(2)<<"
"<<p.U().component(0)<<" "<<p.U().component(1)<<"
"<<p.U().component(2)<<endl;
            }

            // ab = 30deg
            if((aB >= aB30Min) && (aB <= aB30Max))
            {
                  os3<<p.position().component(0)<<"
"<<p.position().component(1)<<" "<<p.position().component(2)<<"
"<<p.U().component(0)<<" "<<p.U().component(1)<<"
"<<p.U().component(2)<<endl;
            }

            // ab = 45deg
            if((aB >= aB45Min) && (aB <= aB45Max))
            {
                  os4<<p.position().component(0)<<"
"<<p.position().component(1)<<" "<<p.position().component(2)<<"
"<<p.U().component(0)<<" "<<p.U().component(1)<<"
"<<p.U().component(2)<<endl;
            }
          }
      }

        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }

    Info<< "End\n" << endl;

    return 0;
}
// ************************************************************************* //
```

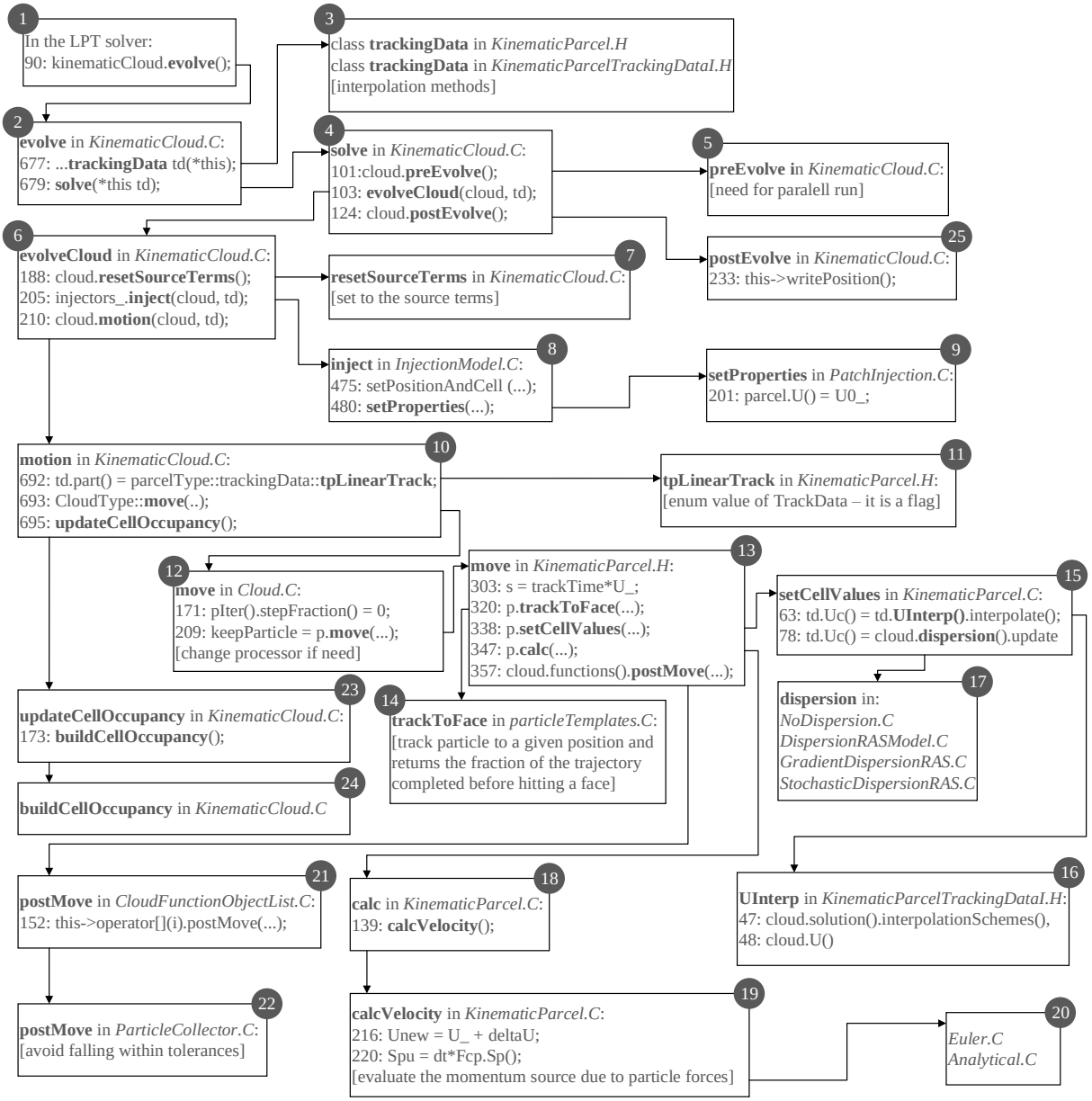
Figure A.16 – Python® script to calculate the particle-phase mean velocity at line α_b = 45° (the file *b45.txt*).

```
import numpy as np

Npart_temps=50000             # number of injected particles
nbin_y=500                    # number of slices between Ri and Ro
ybin_min=0.126                # ybin_min=Ri - bend inner radius
ybin_max=0.226                # ybin_max=Ro - bend outer radius
Rb = 0.176                    # bend radius
L1 = 1.0                      # upstream straight duct length
ab = 45.*np.pi/180.           # bend angle - 45degree

# normal vector to the 45degres cross section plane
normal = [np.cos(ab),-np.sin(ab), 0.]

# to create vectors with the values of position and velocity of b45.txt
Xtemp,Ytemp,Ztemp,Utemp,Vtemp,Wtemp=np.loadtxt('b45.txt',delimiter="
",unpack=True)

N = len(Xtemp)                # vector length

# coordinates of the y line for the statistical treatment

ybin=np.zeros((nbin_y+1))
yy=np.zeros((nbin_y))
ybin[0]=ybin_min
deltay=(ybin_max-ybin_min)/nbin_y

for i in range(nbin_y):
      ybin[i+1]=ybin[i]+deltay

# td is a table with 1 when the line has 1000000 (the time step)

td=np.zeros((N))
nbtemps=0
for i in range(N):
      if (Ytemp[i]==1000000):
            td[i]=1
            nbtemps=nbtemps+1

# instantaneous particle velocity

upartInst=np.zeros((nbtemps,nbin_y+1))
nbpartInst=np.zeros((nbtemps,nbin_y+1))

# mean particle velocity

upartMoy=np.zeros((nbin_y+1))
ntempsMoy=np.zeros((nbin_y+1))

# tables for position Y[j,k] and velocity U[j,k]
# j - particle number, k - time step

Y=np.zeros((Npart_temps,nbtemps))
U=np.zeros((Npart_temps,nbtemps))
nbpart=np.zeros(nbtemps)

# table for the total number of particle nbpart[k]

k=-1
```

```
for i in range(N):
      if td[i]==1:
            j=0
            k=k+1
      if td[i]==0:
        # radius r
            Y[j,k]=np.sqrt((Xtemp[i]-L1)**2+(Ytemp[i]+Rb)**2)
        # normal velocity - V.n = V1n1+V2n2+V3n3
            U[j,k]=Utemp[i]*normal[0]+Vtemp[i]*normal[1]
            j=j+1
            nbpart[k]=j

# instantaneous particle velocity upartInst[k,ii]
# k - time step, ii - part of the y line for the statistical treatment

for k in range(nbtemps):
      for ii in range(nbin_y):
            for j in range(int(nbpart[k])):
                  if Y[j,k]>ybin[ii] and Y[j,k]<=ybin[ii+1]:
                        upartInst[k,ii]=upartInst[k,ii]+U[j,k]
                        nbpartInst[k,ii]=nbpartInst[k,ii]+1
            if nbpartInst[k,ii]!=0:
                  upartInst[k,ii]=upartInst[k,ii]/nbpartInst[k,ii]

# mean particle velocity upartMoy
# ii - part of the y line for the statistical treatment

for ii in range(nbin_y):
      for k in range(nbtemps):
            if nbpartInst[k,ii]!=0:
                  upartMoy[ii]=upartMoy[ii]+upartInst[k,ii]
                  ntempsMoy[ii]=ntempsMoy[ii]+1
      if ntempsMoy[ii]!=0:
            upartMoy[ii]=upartMoy[ii]/ntempsMoy[ii]

# yy - radius position

for ii in range(nbin_y):
      yy[ii]=(ybin[ii]+ybin[ii+1])/2.

# write the results data in the file result_b45.dat
# yy - duct radius position
# upartMoy - mean streamwise particle velocity

with open("result_b45.dat","w") as f:
      f.write("mean particle velcoity\n")
      for ii in range(nbin_y):
            valeurs = "%12.6f %12.6f\n" % (yy[ii],upartMoy[ii])
            f.write(valeurs)
```

The main function of the time loop is `kinematicCloud.evolve()` (line 90 of Figure A.15). Figure A.17 represents the time sequence of the method *evolve* of the current LPT solver. The box 19 shows where in the code the new particle velocity is evaluated and the box 14 where the PWI model will change the particle trajectory.

Figure A.17 – Time sequence for the method *evolve* of the class *basicKinematicCloud*.



Source: Adapted from the "Weekly Report" of Marco Atzori (June 20, 2016).

The OpenFOAM® library with the PWI models is inside the *intermediate* folder of the Lagrangian library with the following path:

```
$WM_PROJECT_DIR/src/lagrangian/intermediate
```

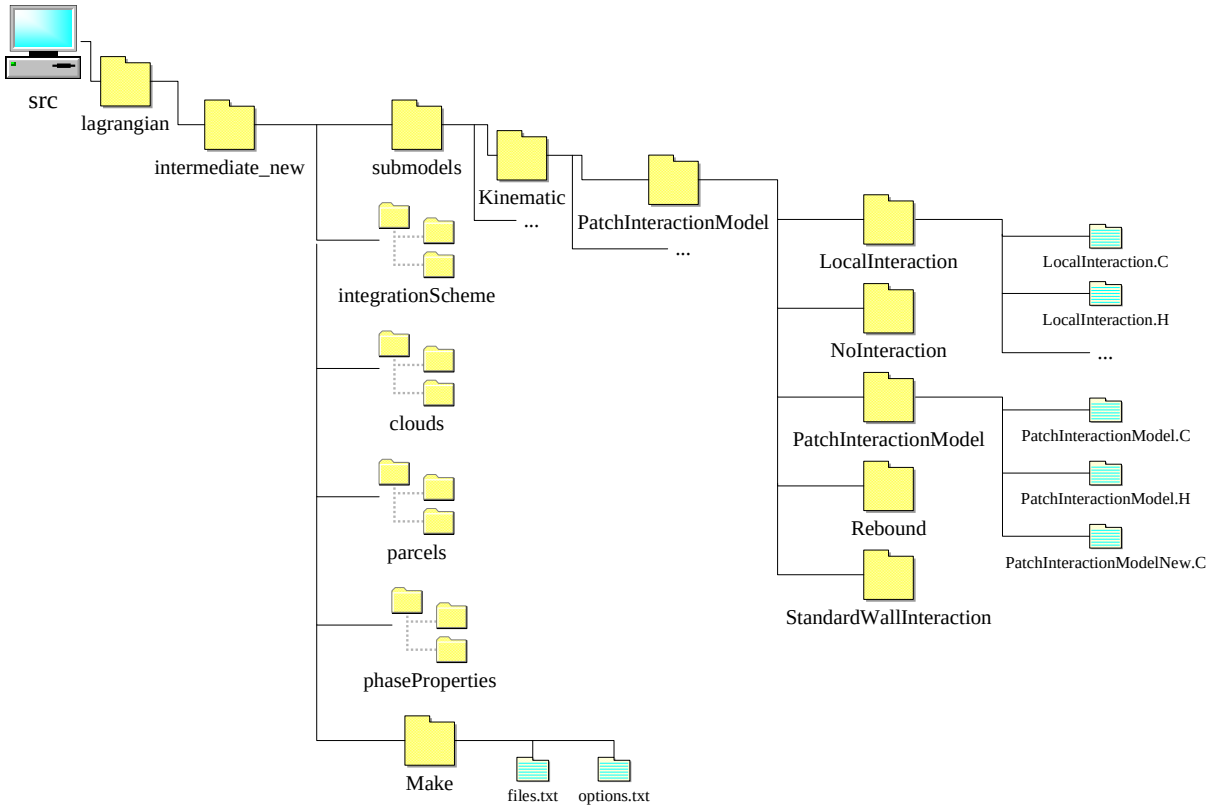
In order to compile the *intermediate* Lagrangian library we need to copy and paste it to the user path directory and change the library name such as the following:

```
$WM_PROJECT_USER_DIR/src/lagrangian/intermediate_new
```

The *intermediate_new* Lagrangian directory is shown in Figure A.18. Some files and folders were hidden for the sake of brevity. We must change the bottom line of the *files.txt* file as following:

```
LIB = $(FOAM_USER_LIBBIN)/liblagrangianIntermediate_new_3Dsym
```

Figure A.18 – The *intermediate_new* folder.



The *PatchInteractionModel.C* file is presented in Figure A.19 in which the lines 66-85 and 120-135 were added to create the new names of the PWI models, i.e., *reboundA* to *reboundD*. The lines 70-73 of the *PatchInteractionModel.H* shown in Figure A.20 must be also included to create these boundary conditions names.

Finally, the PWI models of Section 4.2.3 were applied in the lines 230-497 of the file *LocalInteraction.C* presented in Figure A.21. After changing these files we need to compile the library with the *wmake libso* command line.

Figure A.19 – The *PatchInteractionModel.C* file.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2011-2018 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 "PatchInteractionModel.H"
#include "fvMesh.H"
#include "Time.H"
#include "volFields.H"

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

template<class CloudType>
Foam::wordList
Foam::PatchInteractionModel<CloudType>::interactionTypeNames_
(
    IStringStream
    (
        "(none rebound reboundA reboundB reboundC reboundD stick escape)"
    )()
);

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

template<class CloudType>
Foam::word Foam::PatchInteractionModel<CloudType>::interactionTypeToWord
(
    const interactionType& itEnum
)
{
    word it = "other";

    switch (itEnum)
    {
        case itNone:
        {
            it = "none";
            break;
```

```
60          }
61          case itRebound:
62          {
63              it = "rebound";
64              break;
65          }
66          case itReboundA:
67          {
68              it = "reboundA";
69              break;
70          }
71          case itReboundB:
72          {
73              it = "reboundB";
74              break;
75          }
76          case itReboundC:
77          {
78              it = "reboundC";
79              break;
80          }
81          case itReboundD:
82          {
83              it = "reboundD";
84              break;
85          }
86          case itStick:
87          {
88              it = "stick";
89              break;
90          }
91          case itEscape:
92          {
93              it = "escape";
94              break;
95          }
96          default:
97          {
98          }
99      }
100
101     return it;
102 }
103
104
105 template<class CloudType>
106 typename Foam::PatchInteractionModel<CloudType>::interactionType
107 Foam::PatchInteractionModel<CloudType>::wordToInteractionType
108 (
109     const word& itWord
110 )
111 {
112     if (itWord == "none")
113     {
114         return itNone;
115     }
116     if (itWord == "rebound")
117     {
118         return itRebound;
119     }
120     else if (itWord == "reboundA")
```

```
121         {
122             return itReboundA;
123         }
124         else if (itWord == "reboundB")
125         {
126             return itReboundB;
127         }
128         else if (itWord == "reboundC")
129         {
130             return itReboundC;
131         }
132         else if (itWord == "reboundD")
133         {
134             return itReboundD;
135         }
136         else if (itWord == "stick")
137         {
138             return itStick;
139         }
140         else if (itWord == "escape")
141         {
142             return itEscape;
143         }
144         else
145         {
146             return itOther;
147         }
148     }
149
150
151     // * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * //
152
153     template<class CloudType>
154     Foam::PatchInteractionModel<CloudType>::PatchInteractionModel
155     (
156         CloudType& owner
157     )
158     :
159         CloudSubModelBase<CloudType>(owner),
160         UName_("unknown_U")
161     {}
162
163
164     template<class CloudType>
165     Foam::PatchInteractionModel<CloudType>::PatchInteractionModel
166     (
167         const dictionary& dict,
168         CloudType& owner,
169         const word& type
170     )
171     :
172         CloudSubModelBase<CloudType>(owner, dict, typeName, type),
173         UName_(this->coeffDict().lookupOrDefault("U", word("U")))
174     {}
175
176
177     template<class CloudType>
178     Foam::PatchInteractionModel<CloudType>::PatchInteractionModel
179     (
180         const PatchInteractionModel<CloudType>& pim
181     )
```

```
182 :
183     CloudSubModelBase<CloudType>(pim),
184     UName_(pim.UName_)
185 {}
186
187
188 // * * * * * * * * * * * * * * * * * * Destructor  * * * * * * * * * * * * * * *
189 * //
190
191 template<class CloudType>
192 Foam::PatchInteractionModel<CloudType>::~PatchInteractionModel()
193 {}
194
195
196 // * * * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * *
197 * //
198
199 template<class CloudType>
200 const Foam::word& Foam::PatchInteractionModel<CloudType>::UName() const
201 {
202     return UName_;
203 }
204
205
206 template<class CloudType>
207 void Foam::PatchInteractionModel<CloudType>::info(Ostream& os)
208 {}
209
210
211 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
212 * //
213
214 #include "PatchInteractionModelNew.C"
215
216 //
217 ************************************************************************
218 //
```

Figure A.20 – The *PatchInteractionModel.H* file.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2011-2018 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/>.

Class
    Foam::PatchInteractionModel

Description
    Templated patch interaction model class

SourceFiles
    PatchInteractionModel.C
    PatchInteractionModelNew.C

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

#ifndef PatchInteractionModel_H
#define PatchInteractionModel_H

#include "IOdictionary.H"
#include "autoPtr.H"
#include "runTimeSelectionTables.H"
#include "polyPatch.H"
#include "wallPolyPatch.H"
#include "tetIndices.H"
#include "CloudSubModelBase.H"

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

namespace Foam
{

/*---------------------------------------------------------------------------*\
                   Class PatchInteractionModel Declaration
\*---------------------------------------------------------------------------*/

template<class CloudType>
class PatchInteractionModel
:
```

```
59      public CloudSubModelBase<CloudType>
60  {
61  public:
62
63      // Public enumerations
64
65          // Interaction types
66          enum interactionType
67          {
68              itNone,
69              itRebound,
70              itReboundA,
71              itReboundB,
72              itReboundC,
73              itReboundD,
74              itStick,
75              itEscape,
76              itOther
77          };
78
79          static wordList interactionTypeNames_;
80
81
82  private:
83
84      // Private data
85
86          //- Name of velocity field - default = "U"
87          const word UName_;
88
89
90  public:
91
92      //- Runtime type information
93      TypeName("patchInteractionModel");
94
95      //- Declare runtime constructor selection table
96      declareRunTimeSelectionTable
97      (
98          autoPtr,
99          PatchInteractionModel,
100         dictionary,
101         (
102             const dictionary& dict,
103             CloudType& owner
104         ),
105         (dict, owner)
106     );
107
108
109     // Constructors
110
111         //- Construct null from owner
112         PatchInteractionModel(CloudType& owner);
113
114         //- Construct from components
115         PatchInteractionModel
116         (
117             const dictionary& dict,
118             CloudType& owner,
119             const word& type
```

```
120         );
121
122         //- Construct copy
123         PatchInteractionModel(const PatchInteractionModel<CloudType>& pim);
124
125         //- Construct and return a clone
126         virtual autoPtr<PatchInteractionModel<CloudType>> clone() const =
127 0;
128
129
130     //- Destructor
131     virtual ~PatchInteractionModel();
132
133
134     //- Selector
135     static autoPtr<PatchInteractionModel<CloudType>> New
136     (
137         const dictionary& dict,
138         CloudType& owner
139     );
140
141
142     // Access
143
144         //- Return name of velocity field
145         const word& UName() const;
146
147
148     // Member Functions
149
150         //- Convert interaction result to word
151         static word interactionTypeToWord(const interactionType& itEnum);
152
153         //- Convert word to interaction result
154         static interactionType wordToInteractionType(const word& itWord);
155
156         //- Apply velocity correction
157         //  Returns true if particle remains in same cell
158         virtual bool correct
159         (
160             typename CloudType::parcelType& p,
161             const polyPatch& pp,
162             bool& keepParticle
163         ) = 0;
164
165
166         // I-O
167
168             //- Write patch interaction info to stream
169             virtual void info(Ostream& os);
170 };
171
172
173 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
174
175 } // End namespace Foam
176
177 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
178
179 #define makePatchInteractionModel(CloudType)
180 \
```

```
181
182 \
183     typedef Foam::CloudType::kinematicCloudType;            \
184     defineNamedTemplateTypeNameAndDebug
185 \
186     (
187 \
188         Foam::PatchInteractionModel<kinematicCloudType>,
189 \
190         0
191 \
192     );
193 \
194
195 \
196     namespace Foam
197 \
198     {
199 \
200         defineTemplateRunTimeSelectionTable
201 \
202         (
203 \
204             PatchInteractionModel<kinematicCloudType>,
205 \
206             dictionary
207 \
208         );
209 \
210     }
211
212 #define makePatchInteractionModelType(SS, CloudType)
213 \
214
215 \
216     typedef Foam::CloudType::kinematicCloudType;            \
217     defineNamedTemplateTypeNameAndDebug(Foam::SS<kinematicCloudType>, 0);
218 \
219
220 \
221     Foam::PatchInteractionModel<kinematicCloudType>::
222 \
223         adddictionaryConstructorToTable<Foam::SS<kinematicCloudType>>
224 \
225             add##SS##CloudType##kinematicCloudType##ConstructorToTable_;
226
227 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
228
229 #ifdef NoRepository
230     #include "PatchInteractionModel.C"
231 #endif
232
233 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
234
235 #endif
236
237 //
238 ***************************************************************************
239 //
```

Figure A.21 – The *LocalInteraction.C* file.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2011-2018 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 "LocalInteraction.H"

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

template<class CloudType>
Foam::LocalInteraction<CloudType>::LocalInteraction
(
    const dictionary& dict,
    CloudType& cloud
)
:
    PatchInteractionModel<CloudType>(dict, cloud, typeName),
    patchData_(cloud.mesh(), this->coeffDict()),
    nEscape_(patchData_.size(), 0),
    massEscape_(patchData_.size(), 0.0),
    nStick_(patchData_.size(), 0),
    massStick_(patchData_.size(), 0.0),
    writeFields_(this->coeffDict().lookupOrDefault("writeFields",
false)),
    massEscapePtr_(nullptr),
    massStickPtr_(nullptr)
{
    if (writeFields_)
    {
        word massEscapeName(this->owner().name() + ":massEscape");
        word massStickName(this->owner().name() + ":massStick");
        Info<< "    Interaction fields will be written to " <<
massEscapeName
```

```
55              << " and " << massStickName << endl;
56
57          (void)massEscape();
58          (void)massStick();
59      }
60      else
61      {
62          Info<< "    Interaction fields will not be written" << endl;
63      }
64
65      // check that interactions are valid/specified
66      forAll(patchData_, patchi)
67      {
68          const word& interactionTypeName =
69              patchData_[patchi].interactionTypeName();
70          const typename PatchInteractionModel<CloudType>::interactionType&
71  it =
72              this->wordToInteractionType(interactionTypeName);
73
74          if (it == PatchInteractionModel<CloudType>::itOther)
75          {
76              const word& patchName = patchData_[patchi].patchName();
77              FatalErrorInFunction
78                  << "Unknown patch interaction type "
79                  << interactionTypeName << " for patch " << patchName
80                  << ". Valid selections are:"
81                  << this-
82  >PatchInteractionModel<CloudType>::interactionTypeNames_
83                  << nl << exit(FatalError);
84          }
85      }
86  }
87
88  template<class CloudType>
89  Foam::LocalInteraction<CloudType>::LocalInteraction
90  (
91      const LocalInteraction<CloudType>& pim
92  )
93  :
94      PatchInteractionModel<CloudType>(pim),
95      patchData_(pim.patchData_),
96      nEscape_(pim.nEscape_),
97      massEscape_(pim.massEscape_),
98      nStick_(pim.nStick_),
99      massStick_(pim.massStick_),
100     writeFields_(pim.writeFields_),
101     massEscapePtr_(nullptr),
102     massStickPtr_(nullptr)
103 {}
104
105 // * * * * * * * * * * * * * * * * * * Destructor  * * * * * * * * * * * * * //
106
107 template<class CloudType>
108 Foam::LocalInteraction<CloudType>::~LocalInteraction()
109 {}
110
111 // * * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * //
```

```
112
113 template<class CloudType>
114 Foam::volScalarField& Foam::LocalInteraction<CloudType>::massEscape()
115 {
116     if (!massEscapePtr_.valid())
117     {
118         const fvMesh& mesh = this->owner().mesh();
119
120         massEscapePtr_.reset
121         (
122             new volScalarField
123             (
124                 IOobject
125                 (
126                     this->owner().name() + ":massEscape",
127                     mesh.time().timeName(),
128                     mesh,
129                     IOobject::READ_IF_PRESENT,
130                     IOobject::AUTO_WRITE
131                 ),
132                 mesh,
133                 dimensionedScalar("zero", dimMass, 0.0)
134             )
135         );
136     }
137
138     return massEscapePtr_();
139 }
140
141 template<class CloudType>
142 Foam::volScalarField& Foam::LocalInteraction<CloudType>::massStick()
143 {
144     if (!massStickPtr_.valid())
145     {
146         const fvMesh& mesh = this->owner().mesh();
147
148         massStickPtr_.reset
149         (
150             new volScalarField
151             (
152                 IOobject
153                 (
154                     this->owner().name() + ":massStick",
155                     mesh.time().timeName(),
156                     mesh,
157                     IOobject::READ_IF_PRESENT,
158                     IOobject::AUTO_WRITE
159                 ),
160                 mesh,
161                 dimensionedScalar("zero", dimMass, 0.0)
162             )
163         );
164     }
165
166     return massStickPtr_();
167 }
168
```

```
template<class CloudType>
bool Foam::LocalInteraction<CloudType>::correct
(
    typename CloudType::parcelType& p,
    const polyPatch& pp,
    bool& keepParticle
)
{
    label patchi = patchData_.applyToPatch(pp.index());

    if (patchi >= 0)
    {
        vector& U = p.U();
        bool& active = p.active();

        typename PatchInteractionModel<CloudType>::interactionType it =
            this->wordToInteractionType
            (
                patchData_[patchi].interactionTypeName()
            );

        switch (it)
        {
            case PatchInteractionModel<CloudType>::itNone:
            {
                return false;
            }
            case PatchInteractionModel<CloudType>::itEscape:
            {
                scalar dm = p.mass()*p.nParticle();

                keepParticle = false;
                active = false;
                U = Zero;
                nEscape_[patchi]++;
                massEscape_[patchi] += dm;
                if (writeFields_)
                {
                    label pI = pp.index();
                    label fI = pp.whichFace(p.face());
                    massEscape().boundaryFieldRef()[pI][fI] += dm;
                }
                break;
            }
            case PatchInteractionModel<CloudType>::itStick:
            {
                scalar dm = p.mass()*p.nParticle();

                keepParticle = true;
                active = false;
                U = Zero;
                nStick_[patchi]++;
                massStick_[patchi] += dm;
                if (writeFields_)
                {
                    label pI = pp.index();
                    label fI = pp.whichFace(p.face());
```

```
                massStick().boundaryFieldRef()[pI][fI] += dm;
            }
            break;
        }
    // Perfect Rebound e = 1
        case PatchInteractionModel<CloudType>::itRebound:
        {
            keepParticle = true;
            active = true;

            vector nw;
            vector Up;

            this->owner().patchData(p, pp, nw, Up);

            // Calculate motion relative to patch velocity
            U -= Up;

            scalar Un = U & nw;

            if (Un > 0)
            {
             U -= 2.0*Un*nw;
            }

            // Return velocity to global space
            U += Up;

            break;
        }
    // Model A - Matsumoto and Saito (1970)
        case PatchInteractionModel<CloudType>::itReboundA:
        {
              keepParticle = true;
            active = true;

            vector nw;
            vector Up;

        scalar e;         // normal restitution coefficient
        scalar et;        // tangential restitution coefficient
        scalar fd;        // dynamic friction coefficient
        scalar Uc;        // critical velocity
        scalar alpha;           // incident angle

            this->owner().patchData(p, pp, nw, Up);

            // Calculate motion relative to patch velocity
            U -= Up;

            scalar Un = U & nw;
            vector Ut = U - Un*nw;

        alpha = acos(U&Ut/(mag(U)*mag(Ut)));

        // Virtual wall model of Sommerfeld and Huber (1999)
```

```
282             scalar deltaGamma = 3.8*(constant::mathematical::pi)/180.0;
283       // angle for a Plexiglass plate gamma = 3.8 degree
284
285             // Random normal distribution - Listing 353 (pdf 248) of
286   Holzinger (2018) + StochasticDispersionRAS.H
287
288             #include "Random.H"
289             #include "clock.H"
290
291             scalar qsi;             // standard normal distribution of a
292   random number
293             scalar gamma = -10000.0;      // initialization of a negative
294   roughness angle to do the next while loop
295
296             while ((gamma < 0.) && (abs(gamma) > alpha))
297             {
298                   Random rndGen(clock::getTime()+pid());
299                   qsi = rndGen.scalarNormal();
300                   gamma = deltaGamma*qsi;
301             }
302
303             alpha += gamma;
304
305             scalar en = 1.0-0.015*(alpha);
306             scalar enmin = 0.73;
307
308             e = max(enmin,en);
309
310             scalar fdd = 0.4-0.00926*(alpha);
311             scalar fdmin = 0.15;
312
313             fd = max(fdmin,fdd);
314
315             Uc = 3.5*(fd*(1.0+e)*abs(Un));
316
317             if (abs(Un) < abs(Uc))
318             {
319                   et = 5.0/7.0;
320             }
321             else
322             {
323                   et = 1.0-fd*(1.0+e)*abs(Un/mag(Ut));
324             }
325
326                 if (Un > 0)
327             {
328                 Un *= -e;
329             }
330
331             Ut *= et;
332
333             U = Un*nw+Ut;
334
335                 // Return velocity to global space
336                 U += Up;
337
338                 break;
```

```
            }
          // Model B - Brach and Dunn (1998)
            case PatchInteractionModel<CloudType>::itReboundB:
            {
                  keepParticle = true;
                active = true;

            // To change the velocity components according to Sun et al.
(2011)
            scalar L1=1.0;
            scalar Rb=0.176;
            scalar xp=p.position().component(0)-L1;
            scalar yp=p.position().component(1)+Rb;
            scalar costheta=xp/sqrt(xp*xp+yp*yp);
            scalar sintheta=yp/sqrt(xp*xp+yp*yp);
            scalar Vpx=p.U().component(0);
            scalar Vpy=p.U().component(1);

                vector nw;
                vector Up;

            scalar e;
            scalar et;

                this->owner().patchData(p, pp, nw, Up);

                // Calculate motion relative to patch velocity
                U -= Up;

                scalar Un = U & nw;

            scalar Vpt=sintheta*Vpx-costheta*Vpy;

            // constants for Ag-coated glass particles and stainless
steel wall

            scalar Rpw = 272.0/(272.0+abs(Un));
            scalar phopw =
pow(1.74,0.5)/(pow(1.74,0.5)+pow(abs(Un+0.4),0.5));

            e = Rpw*(1.0-phopw);
            et = 5./7.;

                if (Un > 0)
            {
                Un *= -e;
            }

            Vpt *= et;
            vector AA(sintheta,-costheta,0.);

            U = Un*nw+Vpt*AA;

                // Return velocity to global space
                U += Up;

                break;
```

```
396             }
397           // Model C - Grant and Tabakoff (1975)
398             case PatchInteractionModel<CloudType>::itReboundC:
399             {
400                   keepParticle = true;
401                 active = true;
402
403                 vector nw;
404                 vector Up;
405
406             scalar e;
407             scalar et;
408             scalar alpha;
409
410                 this->owner().patchData(p, pp, nw, Up);
411
412                 // Calculate motion relative to patch velocity
413                 U -= Up;
414
415                 scalar Un = U & nw;
416                 vector Ut = U - Un*nw;
417
418             alpha = acos(U&Ut/(mag(U)*mag(Ut)));
419
420             // Virtual wall model of Sommerfeld and Huber (1999)
421             scalar deltaGamma = 3.8*(constant::mathematical::pi)/180.0;
422       // angle for a Plexiglass plate gamma = 3.8 degree
423
424             // Random normal distribution - Listing 353 (pdf 248) of
425 Holzinger (2018) + StochasticDispersionRAS.H
426
427             #include "Random.H"
428             #include "clock.H"
429
430             scalar qsi;             // standard normal distribution of a
431 random number
432             scalar gamma = -10000.0;      // initiazliation of a negative
433 roughness angle to do the next while loop
434
435             while ((gamma < 0.) && (abs(gamma) > alpha))
436             {
437                   Random rndGen(clock::getTime()+pid());
438                   qsi = rndGen.scalarNormal();
439                   gamma = deltaGamma*qsi;
440             }
441
442             alpha += gamma;
443
444             // Model of Grant and Tabakoff (1975)
445
446             e = 0.993-1.76*alpha+1.56*pow(alpha,2.0)-0.49*pow(alpha,3.0);
447             et = 0.988-1.66*alpha+2.11*pow(alpha,2.0)-
448 0.67*pow(alpha,3.0);
449
450                 if (Un > 0)
451             {
452                 Un *= -e;
```

```
            }

            Ut *= et;

            U = Un*nw+Ut;

                // Return velocity to global space
                U += Up;

                break;
            }
          // Model D - Brauer (1980)
            case PatchInteractionModel<CloudType>::itReboundD:
            {
                  keepParticle = true;
                active = true;

                vector nw;
                vector Up;

            scalar e = 0.9;
            scalar et = 0.8;

                this->owner().patchData(p, pp, nw, Up);

                // Calculate motion relative to patch velocity
                U -= Up;

                scalar Un = U & nw;
                vector Ut = U - Un*nw;

                if (Un > 0)
            {
                Un *= -e;
            }

                Ut *= et;

                U = Un*nw+Ut;

                // Return velocity to global space
                U += Up;

                break;
            }
            default:
            {
                FatalErrorInFunction
                    << "Unknown interaction type "
                    << patchData_[patchi].interactionTypeName()
                    << "(" << it << ") for patch "
                    << patchData_[patchi].patchName()
                    << ". Valid selections are:" << this-
>interactionTypeNames_
                    << endl << abort(FatalError);
            }
        }
```

```
510
511             return true;
512         }
513
514         return false;
515     }
516
517     template<class CloudType>
518     void Foam::LocalInteraction<CloudType>::info(Ostream& os)
519     {
520         // retrieve any stored data
521         labelList npe0(patchData_.size(), 0);
522         this->getModelProperty("nEscape", npe0);
523
524         scalarList mpe0(patchData_.size(), 0.0);
525         this->getModelProperty("massEscape", mpe0);
526
527         labelList nps0(patchData_.size(), 0);
528         this->getModelProperty("nStick", nps0);
529
530         scalarList mps0(patchData_.size(), 0.0);
531         this->getModelProperty("massStick", mps0);
532
533         // accumulate current data
534         labelList npe(nEscape_);
535         Pstream::listCombineGather(npe, plusEqOp<label>());
536         npe = npe + npe0;
537
538         scalarList mpe(massEscape_);
539         Pstream::listCombineGather(mpe, plusEqOp<scalar>());
540         mpe = mpe + mpe0;
541
542         labelList nps(nStick_);
543         Pstream::listCombineGather(nps, plusEqOp<label>());
544         nps = nps + nps0;
545
546         scalarList mps(massStick_);
547         Pstream::listCombineGather(mps, plusEqOp<scalar>());
548         mps = mps + mps0;
549
550         forAll(patchData_, i)
551         {
552             os  << "    Parcel fate (number, mass)      : patch "
553                 <<  patchData_[i].patchName() << nl
554                 << "      - escape                      = " << npe[i]
555                 << ", " << mpe[i] << nl
556                 << "      - stick                       = " << nps[i]
557                 << ", " << mps[i] << nl;
558         }
559
560         if (this->writeTime())
561         {
562             this->setModelProperty("nEscape", npe);
563             nEscape_ = 0;
564
565             this->setModelProperty("massEscape", mpe);
566             massEscape_ = 0.0;
```

```
567
568             this->setModelProperty("nStick", nps);
569             nStick_ = 0;
570
571             this->setModelProperty("massStick", mps);
572             massStick_ = 0.0;
573         }
574     }
575
576     //*********************************************************************//
```