Study and improvement of mass transfer in membranes using CFD technology

Marta Ramos Andrés

Thesis to obtain the Master of Science Degree in

Chemical Engineering

Supervisors: Prof. Vítor Manuel Geraldes Fernandes

Prof. Juan García Serna

Júri

Chairperson: Prof. Sebastião Manuel Tavares da Silva Alves

Members of the Committee: Prof. Vítor Manuel Geraldes Fernandes

Prof. Svetlozar Velizarov

Junho, 2017

Estudo e melhoria da transferência de massa em módulos de membranas utilizanto a tecnologia CFD

Marta Ramos Andrés

Dissertação para obter o grau de Mestre em

Engenharia Química

Orientadores: Prof. Vítor Manuel Geraldes Fernandes

Prof. Juan García Serna

Júri

Presidente: Prof. Sebastião Manuel Tavares da Silva Alves

Vogais: Prof. Vítor Manuel Geraldes Fernandes

Prof. Svetlozar Velizarov

Junho, 2017

Resumo

Foi realizada a simulação do processo de separação de membranas através da tecnología Computational Fluid Dynamics (CFD), visando minimizar o declínio do fluxo de permeado causado pela polarização da concentração (PC).

Foi utilizado o OpenFOAM como software onde um modelo matemático foi formulado em código C++. Foram aplicadas condições periódicas e não periódicas. Após validação, a PC foi estimada por meio da simulação de um dominio periódico impermeável com um valor constante e arbitrário de fração de massa de soluto na parede. Foram aplicados dois métodos, uma correlação para corrigir os coeficientes de transferencia de massa, e a bem conhecida Teoria do Filme (TF). Ambos os métodos foram comparados. A redução do declínio do fluxo de permeado foi estudada comparando ducto tubular e placas paralelas. Também foi estudado o efeito dos espaçadores ea influencia da pressão transmembrana (PTM).

O fator de correção de transferencia de massa obtido foi apropriado para taxas de permeado baixo e alto, enquato que a TP foi aplicada apenas em baixa taxa de permeação. O fator de correção deu um erro relativo máximo de 0.92% eo TP de 1.35%, quado esses métodos foram comparados com os resultados da simulação com permeação. Comparando com a configuração tubular, o declínio do fluxo de permeado é reduzido em 28.6% com placas paralelas e em 49.7-50% com o uso de espaçadores. Com configuraçõ tubular, um aumento de PTM (de 1bar a 3bar) dá um aumento indesejável de 14.6% em soluto no permeado, com placas paralelas a 7.6% e com espaçadores a 5.6-5.4%.

Palavras-chave: polarização de concentração, CFD, OpenFOAM, periódico, fator de correção de transferencia de massa

Abstract

It was carried out the simulation of membrane separation process through Computational Fluid Dynamics (CFD) technology in order to minimize the permeate flux decline caused by the concentration polarization (CP).

It was used OpenFOAM as software where a mathematical model was formulated in C++ code. Periodic and non periodic boundary conditions were applied. After validation, CP was estimated by means the simulation of an impermeable periodic domain with a constant and arbitrary solute mass fraction value at the wall. Two methods were applied, a correlation to correct the mass transfer coefficients, and the well-known Film Theory (FT). Both methods were compared, showing the limitations of FT. Reduction of permeate flux decline was studied comparing tubular and parallel plates ducts. Regarding parallel plates, it was also studied the effect of spacers and the influence of transmembrane pressure (TMP) according to geometry and spacers.

The mass transfer correction factor obtained was appropriate for both low and high permeate rates, while the FT was only applied to low permeation rate. The correction factor gave a maximum relative error of 0.92%, and the FT 1.35%, when these methods were used to estimate the CP comparing their results with results from the simulation with permeation. Comparing with tubular configuration, permeate flux decline is reduced in a 28.6% with parallel plates, and a 49.7-50% with the use of spacers. With the tubular configuration, an increase in TMP (from 1 bar to 3 bar) gives an undesirable increase of 14.6% in solute at the permeate, with parallel plates a 7.6% and with spacers a 5.6-5.4%.

Keywords: concentration polarization, CFD, OpenFOAM, periodic, mass transfer correction factor

Index

Introduction **1**

1. Membrane technology 1

1.1. Membrane filtration 1

1.2. Membrane flux decline 1

1.2.1. Concentration polarization 2

1.2.2. Membrane fouling 2

1.2.3. Methods to reduce flux decline 3

2. Computational Fluid Dynamics 4

2.1. CFD technology 4

2.2. Governing equations 4

3. CFD applied to membranes 6

3.1. Bases 6

3.2. Boundary conditions 7

3.2.1. Concentration polarization at the membrane surface 8

3.2.2. Periodic/wrapped boundary conditions 10

3.3. Literature review 13

3.3.1. First period (1995-2000) 13

3.3.2. Second period (2000-2005) 14

3.3.3. Third period (2005-2010) 14

3.3.4. Last period (2010-2015) 15

3.4. Model 16

3.4.1. Conservation laws 16

3.4.2. Conservation laws with periodic boundary conditions 18

3.4.3. Mass transfer parameters 19

3.4.4. The Graetz problem 22

Objectives **25**

Methodology 27

1. Softwares used 27

1.1. OpenFOAM versión 3.0.1. 27

1.2. Paraview versión 4.4.0. 28

1.3. SolidWorks 2016 28

1.4. Swak4foam 28

1.5. cfMesh versión 1.1.1. 28

2. Simulation 29

2.1. Pre-processing 29

2.1.1. Case geometry 29

2.1.2. Mesh generation 31

2.1.3. Boundary conditions 34

2.1.4. Fluid and membrane properties 36

2.1.5. Flux regime modeling 37

2.1.6. Time step and data output control 37

2.1.7. Numerical schemes 38

2.1.8. Solution and algorithm control 39

2.2. Solving 40

2.2.1. Previous steps 40

2.2.2. Solvers 41

2.3. Post-processing 43

Results and discussion 45

1. Mesh independency 45

1.1. Geometry 1 45

1.2. Geometry 5 46

2. Validation 49

2.1. Geometry 1 and long Geometry 1 49

2.2. Geometry 2 and long Geometry 2 52

2.3. Geometry 3 and long Geometry 3 55

2.4. Periodicity analysis 57

3. Estimation of concentration polarization 59

3.1. Semi-permeable duct 59

3.2. Impermeable periodic domain 64

4. Comparison between correction factor Ξ method and Film Theory method 66
5. Effect of geometry of the duct in concentration polarization 70
6. Effect of spacers in concentration polarization 71
6.1. Rectangular spacers 71
6.2. Semi-cylindrical spacers 74
6.3. Comparison between spacers 75
7. Effect of transmembrane pressure in concentration polarization 76
7.1. Effect according to the geometry 76
7.2. Effect according to the spacers 76

Conclusions 79

References 81

Appendix

Appendix I. BlockMeshDict

Appendix II. checkMesh

Appendix III. Meshes

Appendix IV. MeshDict

Appendix V. Boundary files

Appendix VI. controlDict

Appendix VII. fvSchemes

Appendix VIII. fvSolution

Appendix IX. createPatchDict

List of Figures

Figure 1. Membrane separation principle (Schretter, 2016) 1
Figure 2. Typical flux decline during filtration run (Keir, 2012). 2
Figure 3. Concentration polarization profile under steady-state conditions (Keir, 2012). 2
Figure 4. Schematic of a 3D flow domain depicting the locations of the different boundary conditions (Fimbres-Weihs & Wiley, 2010) 7
Figure 5. Schematic of a permeable wall boundary for modeling of membrane channels (Fimbres-Weihs & Wiley, 2010) 8
Figure 6. Schematic of an impermeable-dissolving wall boundary for modelling of membrane channels (Fimbres-Weihs & Wiley, 2010) 8
Figure 7. A schematic diagram of cell arrangement in a spacer-filled membrane channel (Fimbres-Weihs & Wiley, 2010). 10
Figure 8. Control volume of the duct 16
Figure 9. Variation in the local Nusselt number for laminar flow in tubes (Welty et al., 2008) 23
Figure 10. a) An overview of the OpenFOAM structure. b) Example of a case directory. Cursive letter in the figure indicates type/class of folders or files, not file name (Aslak, 2014). 27
Figure 11. Illustration of the difference between a) a continuous domain and b) a discrete domain for a 1D case (Aslak, 2014). 29
Figure 12. Geometry 1 30
Figure 13. Geometry 2 30
Figure 14. Geometry 3 30
Figure 15. Geometry 4 31
Figure 16. Geometry 5 31
Figure 17. Patches in Geometry 1, Geometry 2, Geometry 3, long Geometry 1, long Geometry 2 and long Geometry 3 34
Figure 18. Patches in Geometry 4 and Geometry 5 35
Figure 19. PIMPLE algorithm (Koch, 2015) 42
Figure 20. Paraview interface 43
Figure 21. Sherwood number vs x-direction with three different meshes in Geometry 1 46
Figure 22. Velocity field with three different meshes in Geometry 5 47
Figure 23. Sherwood number vs x-direction with three different meshes in Geometry 5 48
Figure 24. Sherwood number vs Pe(d/X) in Geometry 1 and long Geometry 1 51
Figure 25. Sherwood number vs Pe(d/X) in Geometry 2 and long Geometry 2 54
Figure 26. Sherwood number vs Pe(d/X) in Geometry 3 and long Geometry 3 57
Figure 27. wAm/wAb vs x-direction in Geometry 1 59
Figure 28. Real semi-permeable duct divided in impermeable periodic domains 60
Figure 29. Velocity in the bulk vs x-direction in a semi-permeable tubular duct 63
Figure 30. Solute mass fraction in the bulk vs x-direction in a semi-permeable tubular duct. 63
Figure 31. Solute mass fraction at the membrane wall vs x-direction in a real semi-permeable tubular duct 66
Figure 32. Concentration polarization index vs ϕ for low permeate rates with mass transfer correction factor (Factor), Film Theory (FT) and CFD 67
Figure 33. Concentration polarization index vs ϕ with mass transfer correction factor (Factor), Film Theory (FT) and CFD 68
Figure 34. Permeate rate Jp vs x-direction in a real semi-permeable tubular duct 69
Figure 35. Solute mass fraction at the membrane wall vs x-direction in Geometry 1, Geometry 2 and Geometry 4 73
Figure 36. Permeate rates vs x-direction in Geometry 1, Geometry 2 and Geometry 4 73

Figure 37. Solute mass fraction at the membrane wall vs x-direction in Geometry 1, Geometry 2 and Geometry 5 74
Figure 38. Permeate rates vs x-direction in Geometry 1, Geometry 2 and Geometry 5 75
Figure 39. Solute mass fraction at the membrane wall vs x-direction in Geometry 4 and Geometry 5 75
Figure 40. Permeate rates vs x-direction in Geometry 4 and Geometry 5 76
Figure 41. Average solute mass fraction at the permeate vs transmembrane pressure in Geometry 1, Geometry 2, Geometry 4 and Geometry 5 77

List of Tables

Table 1. Typical boundary conditions for channel inlet, outlet, impermeable walls and symmetry planes (Fimbres-Weihs & Wiley, 2010) 7
Table 2.Mesh parameters in Geometry 1, Geometry 2 and Geometry 3 32
Table 3. Mesh parameters in long Geometry 1, long Geometry 2 and long Geometry 3 33
Table 4. Mesh parameters in Geometry 4 and Geometry 5 34
Table 5. Type of patches in Geometry 1, Geometry 2, Geometry 3, Geometry 4 and Geometry 5 35
Table 6. Type of patches in long Geometry 1, long Geometry 2 and long Geometry 3 35
Table 7. Fluid and membrane properties 37
Table 8. Different meshes for Geometry 1 45
Table 9. Different refinements for Geometry 1 45
Table 10. Different meshes for Geometry 5 46
Table 11. Different refinements for Geometry 5 48
Table 12. Size of Geometry 1 for the validation 49
Table 13.Size of long Geometry 1 for the validation 49
Table 14. Initial conditions of Geometry 1 for the validation 50
Table 15. Boundary conditions of Geometry 1 for the validation 50
Table 16. Initial conditions of long Geometry 1 for the validation 50
Table 17.Boundary conditions of long Geometry 1 for the validation 50
Table 18. Size of Geometry 2 for the validation 52
Table 19. Size of long Geometry 2 for the validation 52
Table 20. Initial conditions of Geometry 2 for the validation 52
Table 21. Boundary conditions of Geometry 2 for validation 53
Table 22. Initial conditions of long Geometry 2 for the validation 53
Table 23. Boundary conditions of long Geometry 2 for validation 53
Table 24. Size of Geometry 3 for the validation 55
Table 25. Size of long Geometry 3 for the validation 55
Table 26. Initial conditions of Geometry 3 for the validation 55
Table 27. Boundary conditions of Geometry 3 for the validation 56
Table 28. Initial conditions of long Geometry 3 for the validation 56
Table 29. Boundary conditions of long Geometry 3 for the validation 56
Table 30. Size of the Geometry 1 for the periodicity analysis 58
Table 31. Initial conditions of Geometry 1 for the periodicity analysis 58
Table 32. Boundary conditions of Geometry 1 for the periodicity analysis 58
Table 33. Size of the Geometry 1 for the determination of mass transfer correction factor Ξ 60
Table 34. Initial conditions of the Geometry 1 for the determination of mass transfer correction factor Ξ 61
Table 35. Boundary conditions of the Geometry 1 for the determination of mass transfer correction factor Ξ 61
Table 36. Velocity and solute mass fraction in the bulk along the real semi-permeable duct 62
Table 37. Size of the Geometry 1 for the simulation of an impermeable periodic domain 64
Table 38. Initial conditions of the Geometry 1 for the simulation of an impermeable periodic domain 64
Table 39. Boundary conditions of the Geometry 1 for the simulation of an impermeable periodic domain 64
Table 40. Parameters of the mass transfer correction factor Ξ for low permeation rates 65
Table 41. Relative error of the solute mass fraction at the membrane wall with mass transfer correction factor Ξ and Film Theory 67
Table 42. Parameters of the mass transfer correction factor Ξ 68

Table 43. Relative error in the calculation of permeate rates. 69
Table 44. Size of Geometry 2 for the study of the effect of geometry 70
Table 45. Initial conditions of Geometry 2 for the study of the effect of geometry 70
Table 46. Boundary conditions of Geometry 2 for the study of the effect of geometry..... 71
Table 47. Size of Geometry 4 for the study of the effect of spacers 71
Table 48. Initial conditions of Geometry 4 for the study of the effect of spacers 72
Table 49. Boundary conditions of Geometry 4 for the study of the effect of spacers 72
Table 50. Size of Geometry 5 for the study of the effect of spacers 74

List of Abbreviations

Abbreviation	Description
FT	Film Theory
CP	concentration polarization
CFD	Computational Fluid Dynamics
CV	control volume
A	area
1D	one dimension
2D	two dimensions
3D	three dimensions
NF	nanofiltration
RO	reverse osmosis
SWM	spiral-wound membrane
UF	ultrafiltration
RED	reverse electrodialysis
CAD	computer-aided design
CAE	computer-aided engineering
SMP	shared memory parallelization
MPI	message passing interface
TVD	total variation diminishing
NVD	normalized variable diagram
Geom	geometry
GAMG	Generalized Algebraic Multi-Grid
PBiCG	Preconditioned Bi-Conjugate Gradient
DILU	Diagonal Incomplete-LU
PISO	Pressure Implicit Splitting of Operators
SIMPLE	Semi-Implicit Method for Pressure Linked Equations

List of Symbols

Variable	Description	Units
k_c	Mass transfer coefficient	m/s
D_{AB}	Diffusion coefficient	m^2/s
δ	Film thickness	m
Sh	Sherwood number	-
d_h	Hydraulic diameter	m
a	Sh correlation parameter	-
b	Sh correlation parameter	-
c	Sh correlation parameter	-
L	Length of the channel	m
Re	Reynolds number	-
Sc	Schmidt number	-
$\vec{v}$	Velocity vector	m/s
u	x-direction velocity	m/s
$\hat{u}$	x-direction unit vector	-
v	y-direction velocity	m/s
w	z-direction velocity	m/s
υ	Kinematic viscosity	m^2/s
ρ	Density	kg/m^3
∇, grad	Gradient	m^{-1}
p	Pressure	Pa
∇^2	Laplacian	m^{-2}
f	External body forces per mass	m/s^2
t	Time	s
μ	Viscosity	Ns/m^2
ψ	Cauchy stress tensor	N/m^2
F	Body force	N
φ	General dependent variable	--
Γ_ω	General transfer coefficient	--
S_φ	General source-sink term	--
Δt	Time step	s
J_p	Permeate rate	m/s
L_p	Hydraulic permeability	m^2s/kg
Δp	Transmembrane pressure	Pa
σ	Osmotic coefficient	kg/ms^2
$\Delta\pi$	Osmotic pressure	Pa
χ	Osmotic pressure coefficient	Pa
w	Solute mass fraction	–
R_{real}	Real rejection	–
θ	Scaled solute mass fraction	–
$\overrightarrow{N}$	Mass flow per unit area	kg/m^2
ft	Upstream over downstream	–
Pe	Peclet number	–
Sh^0	Sh withouth permeation	–
H	Height	M
p^*	Periodic part of pressure	Pa
β	Average pressure gradient	Pa/m
x	x-coordinate position	–
y	y-coordinate position	–
z	z-coordinate position	–
$k_{c,p}$	k_c with permeation	m/s
S	Mass source term	s^{-1}

Variable	Description	Units
Ξ	Mass transfer correction factor	-
M	Concentration polarization index	-
ξ	Function to obtain Ξ	-
c_1	Correction factor parameter	-
c_2	Correction factor parameter	-
c_3	Correction factor parameter	-
Co	Courant number	-

Subscripts

Variable	Description
m	membrane
p	permeate
i	*initial*
b	bulk
us	upstream
ds	downstream
s	solute
slv	*solvent*
loc	*local*
0	*inlet*

Introduction

1. Membrane technology

1.1. Membrane filtration

Membrane filtration is a widely applied technique that is used for removal of solvent molecules (often water) from a solution containing one or more solute species (such as salt). The membrane acts as a selective thin layer (typically less than 1 mm-thick) which is a physical barrier that completely rejects or reduces the flux of a given chemical species or particles through the membrane so it can be separated from the main stream.

During the operation, the water (Feed) is pumped across the membrane, which allows certain constituents to pass through it and other constituents still remain in the stream, which is known as waste stream, concentrate or retentate. The stream that contains permeable components is known as product stream or permeate, which is relatively free of constituents.

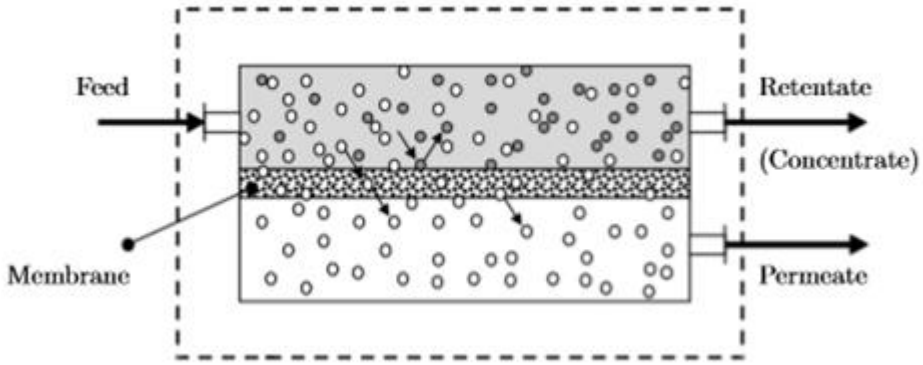


Figure 1. Membrane separation principle (Schretter, 2016)

Membrane filtration plays an important role in modern life, and has grown in a matter of years from a specialized laboratory tool into a widespread industrial process with significant industrial, economic and environmental importance. Membranes are used today for a variety of large scale applications, including (Keir, 2012):

- Treatment of drinking water to remove contaminants and pathogens
- Production of potable water from seawater
- Treatment of domestic and industrial wastewater
- Recovery of valuable constituents from industrial effluent
- Various medical applications, such as removal of urea and toxins from blood streams by dialysis

1.2. Membrane flux decline

The primary technical issue which has limited the application of membrane filtration is that of flux decline – during a membrane separation process, the flux through the membrane will inevitably decrease with time. The flux decline can be substantial in some processes: flux is often less than 5% of the pure water flux. Eliminating or reducing the extent of flux decline is thus one of the primary aims of membrane research (Keir, 2012).

This flux decline can be attributed to two main factors: polarization phenomena (mainly concentration polarization), and membrane fouling. A typical pattern of flux decline over the duration of a membrane filtration run illustrating the influence of these two factors is shown in the following figure.

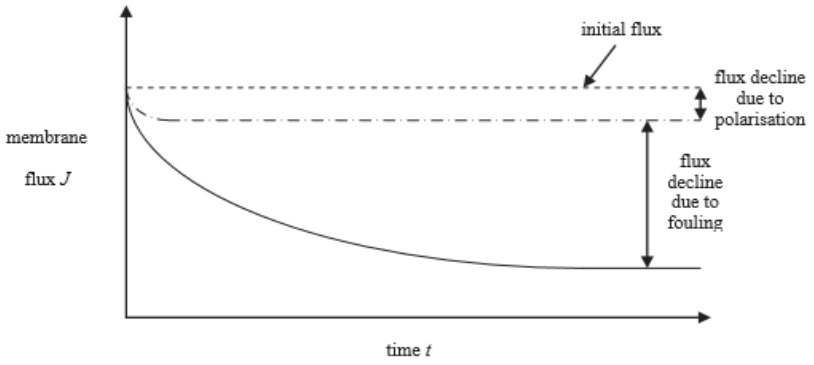


Figure 2. Typical flux decline during filtration run (Keir, 2012).

Flux decline due to polarization occurs relatively rapidly, but is a reversible process which represents only a small amount of the total flux decline. The reminder of the flux decline is due to membrane fouling. However, fouling is not completely independent of polarization phenomena, and can be minimized by restricting the extent of polarization. It is useful to distinguish between concentration polarization and membrane fouling.

1.1.1. Concentration polarization

Concentration polarization occurs when rejected solutes accumulate next to the membrane wall, forming a boundary layer. This accumulation of solute causes a diffusive flow back to the feed, which eventually reaches a steady-state condition where the solute flow to the membrane surface equals the solute flux through the membrane plus the diffusive flow from the membrane surface back to the feed. This establishes a concentration profile within the boundary layer as shown in the following figure.

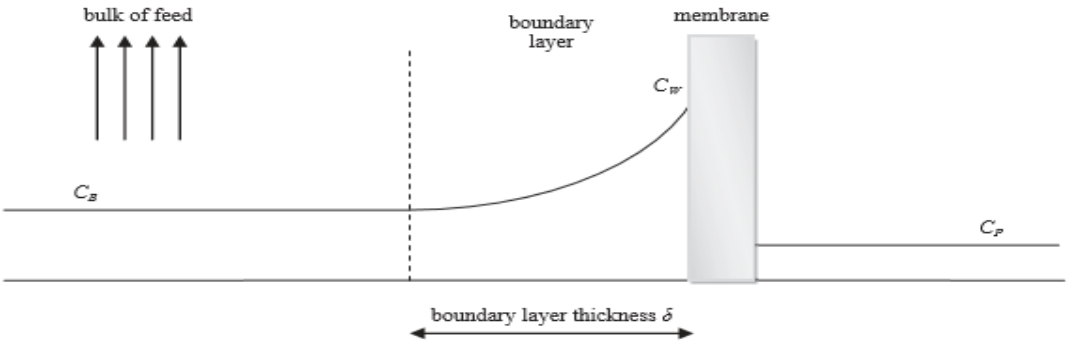


Figure 3. Concentration polarization profile under steady-state conditions (Keir, 2012).

The primary consequence of concentration polarization is the reduction in permeate flux, by either increasing the osmotic pressure on the feed side of the membrane or forming a gel layer at the membrane surface.

1.1.2. Membrane fouling

Distinct from concentration polarization, membrane fouling is defined as the irreversible deposition of retained particles, colloids, emulsions, suspensions, macromolecules, salts or other substances on or within the membrane. Membrane fouling depends on an array of physical and chemical parameters (such as concentration, temperature, pH, ionic strength and others) as well as other specific interactions (e.g. Van der Walls forces, hydrogen bonding, dipole interactions). As such, the phenomenon of fouling is extremely complex and not fully understood (Keir, 2012).

1.1.3. Methods to reduce flux decline

A multitude of methods have been developed to attempt to reduce the flux decline phenomenon, though the complex nature of fouling makes a general approach difficult. The strategies employed fall largely into four groups (Keir, 2012):

- Pre-treatment of the membrane feed solution
- Modification of the membrane properties (surface and materials)
- Control of membrane operating parameters
- Membrane cleaning

In this work the third group has been studied because it is not an experimental work, but a CFD work.

Minimizing the effect of concentration polarization can be achieved by either reducing membrane flux (which is usually undesirable), or by increasing the mass transfer coefficient k_c of the system, defined as:

$$k_c = \frac{D_{AB}}{\delta} \qquad (1)$$

The mass transfer coefficient characterizes the hydrodynamics of the membrane filtration system, i.e. the value of k_c is governed by the fluid properties (viscosity, density, diffusivity), the fluid velocity, and the membrane geometry. The mass transfer coefficient can be related to some well-known dimensionless numbers as follows:

$$Sh = \frac{k_c d_h}{D_{AB}} = aRe^b Sc^c \left(\frac{d_h}{L}\right)^d \qquad (2)$$

$$Re = \frac{d_h u}{\upsilon} \qquad (3)$$

$$Sc = \frac{\upsilon}{D_{AB}} \qquad (4)$$

Where Sh is the Sherwood number, Re is the Reynolds number, Sc is the Schmidt number, a, b, c and d are constants, d_h is the hydraulic diameter, L is the length of the channel, u is the flow velocity, and υ is the kinematic viscosity. Generally, an increase in Reynolds number leads to an increase in the mass transfer coefficient, due to increased local shear stress on the membrane surface which reduces concentration polarization and cake formation. The majority of methods to reduce concentration polarization thus focus on promoting turbulent conditions to maximize shear stress and hence mass transfer (Keir, 2012).

2. Computational Fluid Dynamics

2.1. CFD technology

Computational Fluid Dynamics or CFD is the use of numerical methods and algorithms to analyze problems involving fluid dynamics, heat and mass transfer, transport process and so on. As an analysis tool, CFD has the advantages such as the reduction of lead times and costs.

CFD has been developed in recent years in both research and industry organizations. With ongoing development of computer hardware, mathematical-physical models and numerical solution algorithms, the simulation results of complex fluid-dynamic problems with CFD become more reliable. The research work in the process engineering and chemical industry, which contains a lot of work with fluid mechanics, could be speeded up with the help of CFD methods (Ding, 2012)

Mass, momentum and energy balances are the basic laws in the fluid flow and the transport process, which have been described with a series of governing laws in different situations. CFD involves the numerical solution of these governing laws of fluid dynamics. The complex sets of partial differential equations are solved in geometrical domain divided into small segments/volumes, commonly known as mesh (or grid) (Ding, 2012).

2.2. Governing equations

The fundamental governing equations of fluid dynamics, upon which all of fluid dynamics are based, are continuity, momentum and energy equations that are respectively the mathematical statement of foundational physical conservation laws for all of fluid dynamics: the Conservation of Mass, the Newton's Second Law (or the Conservation of Momentum) and the Conservation of Energy. These basic Mass Conservation law and Momentum Conservation law can be described by the following equations, respectively (Cao, 2015):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \qquad (5)$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v}\vec{v}) = -\nabla p + \mu \nabla^2 \vec{v} + \rho f \qquad (6)$$

Where ρ is the density of the fluid (kg/m^3); $\vec{v}$ is the fluid velocity (m/s); t is the time (s); p is the pressure (N/m^2); μ is the viscosity of the fluid (Ns/m^2); f is the external volume or body forces per unit volume (m/s^2).

Equation 5 is a partial differential equation (PDE) form of the continuity equation. For an incompressible fluid with constant density like water, $\partial p/\partial t = 0$, and the equation becomes $\mathrm{div}(\vec{v}) = 0$, which can also be written as $\partial v_i/\partial x_i = 0$.

Equation 6 is also known as Navier-Stokes equation, which is the representation of the flow motion for incompressible newton's fluid with consideration of the effect of viscosity of the fluid. If the flow is in-viscid, the dissipative, transport phenomena of viscosity, mass diffusion, and thermal conductivity can be neglected. Then we have the Euler equations, which is expressed as below:

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v}\vec{v}) = -\nabla p + \rho f \qquad (7)$$

Equation 7 can be also expressed in a more general form by the Cauchy momentum equation, which is Cauchy's first law of motion:

$$\rho\frac{\mathrm{D}\vec{\mathrm{v}}}{\mathrm{Dt}} = \nabla\cdot\psi + \mathrm{F} \qquad (8)$$

Where F is the body force (N), e.g. gravitational force, buoyancy force, centrifugal force and in some cases electric field force and electromagnetic force. ψ is the Cauchy stress tensor ($\mathrm{N/m^2}$), which is from the pressure distribution acting on the surface and the shear and normal stress distributions acting on the surface.

Equation 6 can be expressed in a more general form by the conservation law of all the fluid flow, as shown in the following expression (Cao, 2015):

$$\frac{\partial(\rho\varphi)}{\partial\mathrm{t}} + \mathrm{div}(\rho\vec{\mathrm{v}}\varphi) = \mathrm{div}\left(\Gamma_\varphi \mathrm{grad}\varphi\right) + \mathrm{S}_\varphi \qquad (9)$$

Where φ is the dependent variable, which can be scalar or vector. When φ is equal to 1, the conservation of mass equation can be derived. When φ is v, the equation stands for the conservation of momentum. φ can also be temperature, and concentration of the fluid, which means respectively heat transport and mass transport. Γ_φ is the appropriate coefficient for variable φ. In mass transport equation this coefficient can be calculated with $\Gamma_\varphi = \rho\mathrm{D}$, D is the diffusion coefficient. S_φ is the source-sink term.

The integral form of the conservation equation, which is the actual form solved by CFD program, is shown in the following:

$$\frac{\partial}{\partial\mathrm{t}}\int_{\mathrm{CV}}\rho\varphi\mathrm{dV} + \int_{\mathrm{A}}\mathrm{n}\cdot(\rho\vec{\mathrm{v}}\varphi)\mathrm{dA} = \int_{\mathrm{A}}\mathrm{n}\cdot\left(\Gamma_\varphi\mathrm{grad}\varphi\right)\mathrm{dA} + \int_{\mathrm{CV}}\mathrm{S}_\varphi\mathrm{dV} \qquad (10)$$

Given the time dependent problems the transport equation can be expressed as the most generate integrated form as below:

$$\int_{\Delta\mathrm{t}}\frac{\partial}{\partial\mathrm{t}}\int_{\mathrm{CV}}\rho\varphi\mathrm{dV} + \int_{\Delta\mathrm{t}}\int_{\mathrm{A}}\mathrm{n}\cdot(\rho\vec{\mathrm{v}}\varphi)\mathrm{dA} = \int_{\Delta\mathrm{t}}\int_{\mathrm{A}}\mathrm{n}\cdot\left(\Gamma_\varphi\mathrm{grad}\varphi\right)\mathrm{dA} + \int_{\Delta\mathrm{t}}\int_{\mathrm{CV}}\mathrm{S}_\varphi\mathrm{dV} \qquad (11)$$

3. CFD applied to membranes

3.1. Bases

A number of studies have focused on enhancing microfiltration, ultrafiltration, nanofiltration and reverse osmosis processes. Some of the studies include (Ghidossi, Veyret, & Moulin, 2006):

(I) testing of new membrane materials;
(II) use of different pore sizes;
(III) determination of conditions for optimal selectivity;
(IV) attempts to determine the optimum trans-membrane pressure or permeate flux to minimize fouling.

The design of a new membrane typically requires a considerable amount of process development. The effects of various parameters on permeate flux decline and the mechanisms of membrane fouling have been investigated, but little progress has been made in understanding the fundamental mechanisms of membrane fouling. The definition of a generic model with applicability to membrane systems requires rigorous and robust methods, but the complexity of these processes is reduced using numerical simulations. The CFD simulations yield a better understanding of these complex processes and help minimize the number of experiments. As a consequence, CFD has become an effective tool to achieve the goal of a better design more rapidly and cost effectively (Ghidossi et al., 2006).

Numerous improvements of the technology have allowed membrane section for a particular process to be done more easily and more quickly. One of the important aspects is the understanding of membrane fouling and subsequent permeate flux decline. Experimental studies provided only qualitative or perhaps semi-quantitative information on concentration polarization, cake formation and fouling phenomena. Several techniques in the fields of nanotechnology, microfluidics, optics, spectroscopy and sensors need to be refined in order to improve their accuracy and resolution. Some of them are light deflection techniques, magnetic resonance imaging, direct pressure measurements, direct observation through the membrane, direct visualization above the membrane, laser triangulometry and optical laser sensors. But, despite the potential of these techniques to advance the understanding of membrane filtration, the investigations available are quite limited in scope. This is the reason why computational fluid dynamics is so often used to develop and understand these processes. Simulations are more precise and easier to perform than experimental studies but require the input of suitable constitutive relations for quantities such as viscosity, diffusivity, rejection coefficient, etc., which are not always known to estimate the cake thickness, the rejection or the profile of the flow (Ghidossi et al., 2006).

Fluid transport phenomena are of great importance for membrane separation systems. The performance of a membrane module is inherently linked to the fluid movement through its volume, and thus its geometrical configuration is an important design consideration. The system will take into account the flow, the bulk region of the module, the boundary layer transport, the diffusive transport and the separation activity driven by the surface wet-ability, as well as the electrostatic, chemical and physical interactions at the membrane surface. All these processes are interdependent and non-linear, A CFD approach is suitable for combining them into a numerical simulation and see the effects of different parameters on the system performance (Ghidossi et al., 2006).

3.2. Boundary conditions

There are several ways of specifying appropriate boundary conditions applicable to flows in narrow conduits such as those encountered in membrane processes. The most common are the inlet, outlet, opening, wall boundary and symmetry conditions. As their names imply, these boundary conditions are applied at the inlet, outlet, symmetry planes and non-permeable walls such as spacer surfaces and channel walls that are not membranes. It should be noted that this terminology is applicable to both 2D and 3D modeling. The following table shows the definition of the boundary conditions for 3D modeling, and the figure depicts the typical locations of these boundaries on a 3D flow domain (Fimbres-Weihs & Wiley, 2010).

Table 1. Typical boundary conditions for channel inlet, outlet, impermeable walls and symmetry planes (Fimbres-Weihs & Wiley, 2010)

Boundary location	Boundary condition	Mathematical condition
Inlet	Specified velocity ($\vec{v}_{in}$) and concentration ($\omega_{i,in}$) profiles	$\vec{v} = \vec{v}_{in}(x,y,z) \quad \omega_i = \omega_{i,in}(x,y,z)$
Inlet	Specified pressure (p_{in}), flow direction ($\hat{v}$; inlet velocity is calculated) and concentration profile ($\omega_{i,in}$)	$p = p_{in} \quad \vec{v} = v_{calc}\hat{v} \quad \omega_i = \omega_{i,in}(x,y,z)$
Outlet	Specified uniform static pressure, constant concentration gradient	$p = p_{out} \quad \frac{\partial \omega_i}{\partial L} = 0$
Outlet	Specified average static pressure, constant concentration gradient	$\frac{1}{A_{out}}\int_{S_{out}} p\,dA = p_{out} \quad \frac{\partial \omega_i}{\partial L} = 0$
Opening	Specified average static pressure; calculated concentration for flow out of domain, specified for flow into domain	$\frac{1}{A_{out}}\int_{S_{out}} p\,dA = p_{out} \quad (\omega_i)_{inflow} = \omega_{i,in}$
Wall	No-slip wall, no mass transfer	$\vec{v} = 0 \quad \frac{\partial \omega_i}{\partial n} = 0$
Symmetry Plane	Velocity normal to plane (v_n) set to zero, velocity and concentration gradients normal to plane set to zero.	$v_n = 0, \quad \frac{\partial \vec{v}}{\partial n} = 0 \quad \frac{\partial \omega_i}{\partial n} = 0$

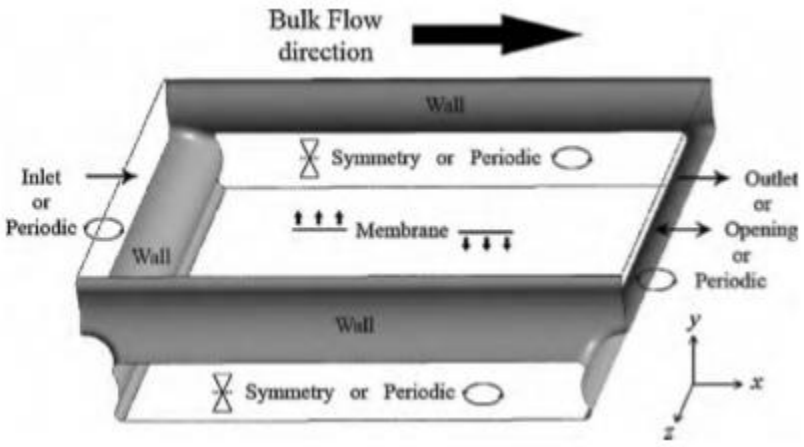


Figure 4. Schematic of a 3D flow domain depicting the locations of the different boundary conditions (Fimbres-Weihs & Wiley, 2010)

Appropriate choice of boundary conditions can help simplify the flow problem, and allow for more efficient calculation of the converged numerical solution. In particular, symmetry planes are used when it is known a-priori that the flow conditions on both sides of a plane are the exact reflection of each other, e.g. the plane located at half the channel height in an unobstructed rectangular channel under laminar flow conditions. Symmetry boundary conditions can also be used to model 2D flow when utilizing a 3D solver, in which case the side boundary conditions are specified as symmetry planes (Fimbres-Weihs & Wiley, 2010).

Although the choice of inlet and outlet boundary conditions will depend on the operating conditions of the process being simulated, the most commonly used inlet boundary condition involves specifying the velocity and concentration profiles. Inlet velocity profiles can be specified if they are known a-priori (such as a fully developed velocity profile obtained analytically) or, if no information is known, they can be defined as a flat profiles with magnitude equal to the average velocity. Although they are strictly physically unrealistic, flat

profiles are commonly used when no information is known about the geometry of the conduit upstream of the channel inlet. When using these inlet conditions, it is essential to include an entrance section in order to allow the velocity profile to develop according to the geometry of the conduit (Fimbres-Weihs & Wiley, 2010).

For the outlet boundary condition, an outflow boundary condition can be specified for steady flows, but an "opening" boundary condition is preferred for unsteady flows due to possible recirculation zones near the outlet, which can occur under a vortex shedding flow regime (Fimbres-Weihs & Wiley, 2010).

3.2.1. Concentration polarization at the membrane surface

From the numerical viewpoint, concentration polarization refers to a concentration profile that results in a higher concentration of solute at the membrane surface than in the bulk flow. Published studies that utilize CFD to analyze this phenomenon have generated this type of concentration profiles via two different boundary conditions at the membrane surface: the permeable wall case, and the impermeable or dissolving wall case:

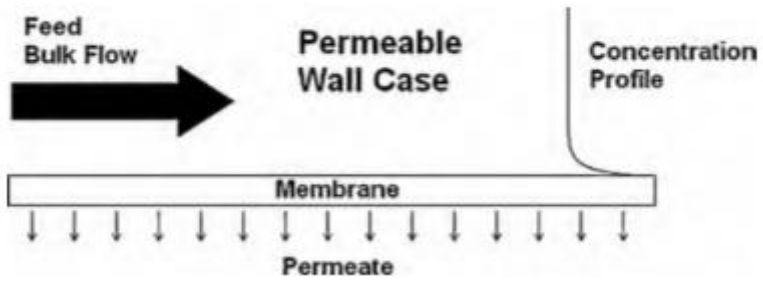


Figure 5. Schematic of a permeable wall boundary for modelling of membrane channels (Fimbres-Weihs & Wiley, 2010)

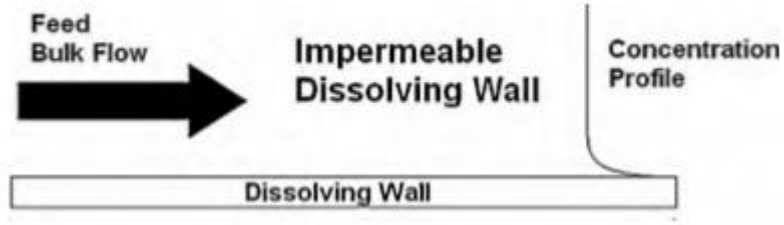


Figure 6. Schematic of an impermeable-dissolving wall boundary for modelling of membrane channels (Fimbres-Weihs & Wiley, 2010)

- Permeable wall

It is possible to set the boundary condition as permeable walls such that there is fluid extraction normal to the wall. In such cases, the fluid velocity normal to the wall must be specified, which can be obtained from the permeate flux. The local permeate flux can be calculated following the approach of Kedem and Katchalsky and Merten, which yields the following expression:

$$J_p = L_p(\Delta p - \sigma\Delta\pi) \qquad (12)$$

J_p is the permeate flux or velocity depending of L_p units, L_p the hydraulic permeability of the membrane, Δp the transmembrane pressure, σ the osmotic coefficient and $\Delta\pi$ the osmotic pressure.

One of the main sources of uncertainty when utilizing the permeate wall boundary condition is the lack of complete knowledge of the distribution of the hydraulic permeability along the membrane surface. Membranes may exhibit substantial hydraulic permeability variation due to their manufacturing process. This has been shown to have a significant effect on solute concentration at the membrane surface and, thus, on concentration polarization (Fimbres-Weihs & Wiley, 2010).

The osmotic pressure of the solute can be approximated by an expression taking the following form:

$$\pi = \chi w \qquad (13)$$

Where the value for the osmotic pressure coefficient (χ) can either be a constant (which yields a linear equation relating osmotic pressure to concentration), or an expression depending on concentration. The linear version of Equation 13 is often only applicable at low solute concentrations. Many solutes exhibit osmotic pressures significantly higher than predicted by this equation at concentrations typically encountered in the concentration boundary layer adjacent to a membrane surface (Fimbres-Weihs & Wiley, 2010).

Substituting Equation 13 into Equation 12 gives the following expression for the local fluid velocity normal to the wall in terms of the local wall solute mass fraction:

$$J_p = L_p[\Delta p - (\chi_m w_m - \chi_p w_p)] \qquad (14)$$

The local solute mass fraction and pressure at the membrane wall can be directly obtained from the CFD simulation results.

The boundary condition for the solute mass fraction at the membrane surface is obtained from the mass balance of solute at the membrane surface:

$$J_p \rho w_p = J_p \rho w_m + \rho D_{AB} \left(\frac{\partial w}{\partial y}\right)_m \qquad (15)$$

From the mass balance, the following expression in terms of the intrinsic or real rejection of the solute at the membrane can be used as boundary condition:

$$-D_{AB} \left(\frac{\partial w}{\partial y}\right)_m = J_p w_m R_{real} \qquad (16)$$

It is important to note that real rejection is a local membrane property. Apparent rejection, on the other hand, is an averaged quantity over the membrane area and can be measured experimentally.

- Impermeable dissolving wall

Despite the fact that the permeable wall model is a realistic representation of what happens in a real membrane separation system, it involves the calculation of local fluxes during simulation run time. These calculations add a further computational load to an already large numerical problem, especially in 3D computations. On the other hand, the impermeable wall model involves fewer calculations and is therefore a simpler and more computationally efficient boundary condition (Fimbres-Weihs & Wiley, 2010).

The impermeable-dissolving wall model (see Figure 6) treats the boundary as a non-slip wall ($\vec{v} = 0$), and specifies either a constant concentration at the wall ($w_i = w_{i,m}$), or a constant dissolution rate from the wall $(\partial w_i / \partial y)_m$. Given that the relative magnitude of the permeation velocity is usually a few orders of magnitude smaller than the average fluid velocity, the change of Reynolds number along a permeable channel due to fluid extraction will have a minimal effect on wall shear and mass transfer over short channel lengths (Fimbres-Weihs & Wiley, 2010).

However, it is also possible to extrapolate mass transfer results obtained using the impermeable wall boundary condition to cases with permeation. Investigations (Miranda & Campos, 2002) found that, even for complex geometries, impermeable mass transfer coefficients did not deviate more than 25% to those predicted by the film theory equation for flow with permeation. They recommend the use of mass transfer data obtained using a uniform wall concentration when estimating permeable mass transfer coefficients for cases where the ratio for inlet osmotic pressure to transmembrane pressure is higher than 0.8. Alternatively, for cases where this ratio is less than 0.8, they recommend the use of mass transfer data obtained using a uniform dissolution rate. Moreover, for channel geometries and flow conditions typically found in NF and RO, studies (Vitor Geraldes & Afonso, 2006) demonstrated that Sherwood numbers obtained for cases without permeation can be used to predict the wall conditions in cases where permeation is present. They developed a correlation for this purpose which is valid for a larger range of permeation rates than the film theory.

Therefore, the impermeable-dissolving wall model is capable of providing valuable insights into the flow field induced mass transfer phenomena taking place inside narrow channels while avoiding higher computational costs, and the results obtained can be used to estimate the flow and mass transfer conditions in membrane modules.

3.2.2. Periodic/wrapped boundary conditions

There are cases in which it is not desirable to model the entire flow domain due to computational or time restrictions. This is usually the case for three-dimensional simulations. In these cases, spatially periodic boundary conditions may be employed, which involve the periodic “wrapping” of the flow variables. Periodic momentum boundary conditions are commonly used for geometries in which a particular pattern is repeated over the length of the fluid domain (e.g. Figure 7). This repeating pattern, such as the spacer filament mesh, generates a spatially repeating velocity field beyond the developing or entry region. The use of spatially periodic boundary conditions has also been used in 2D studies to reduce computational requirements and increase spatial resolution (Fimbres-Weihs & Wiley, 2010).

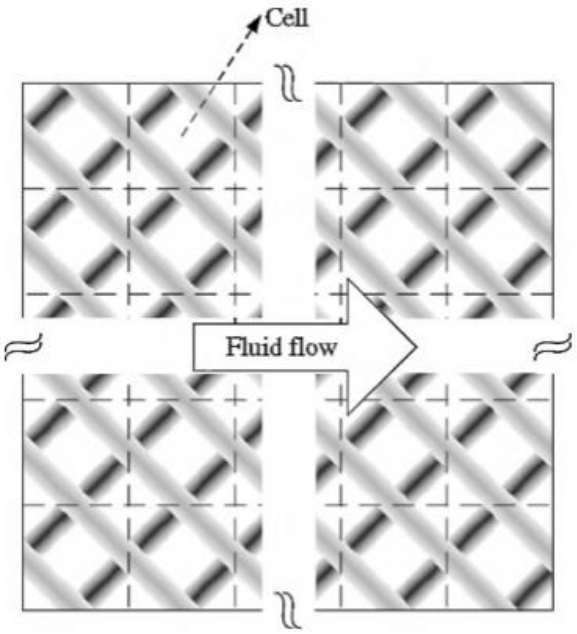


Figure 7. A schematic diagram of cell arrangement in a spacer-filled membrane channel (Fimbres-Weihs & Wiley, 2010).

An study (Yuan, Tao, & Wang, 1998) showed that for periodically disturbed duct flow with heat transfer, entrance effects are much shorter than for flows without the disturbances. Often, the heat and fluid flow become fully developed after about 5 repetitions of the periodic section. Since heat transfer units may consist of dozens or even hundreds of these repeating sections, the analysis of just one periodic section can give a clear picture of the transfer phenomena dictating the overall performance of the unit. Therefore, for this and other types of spatially repeating geometries many heat transfer studies have utilized a fully developed temperature profile, i.e. a spatially periodic heat transport boundary condition. These concepts can be extrapolated to mass transfer (Fimbres-Weihs & Wiley, 2010).

As (Darcovich, Dalcin, & Gros, 2009) point out, spatial periodicity should not be confused with temporal periodicity. The former refers to a repeating flow pattern due to a repeating geometric pattern in the spatial sense, whereas the latter refers to a repeating behavior of the flow pattern over regular time intervals. Temporal periodicity has been observed for certain spacer geometries, but is usually not imposed on the flow a-priori as there is no guarantee that the flow pattern will repeat itself exactly for more complex spacer geometries. Even for a simple 2D zig-zag spacer, temporal periodicity was not observed for the flow (Fimbres-Weihs, Wiley, & Fletcher, 2006; Schwinge, Wiley, & Fletcher, 2003). All of the current literature on 3D modeling for spacer-filled channels in membrane systems has focused on spatial periodicity.

Currently, two distinct periodic boundary conditions for mass transport with application to membrane separation have been proposed. The first is an extrapolation of the approach used for periodic temperature wrapping used in heat transfer (Fimbres-Weihs & Wiley, 2007; Koutsou, Yiantsios, & Karabelas, 2009), which is only applicable to flows with low or no permeation through the membrane wall. The second (Darcovich et al., 2009; Lau, Abu Bakar, Ahmad, & Murugesan, 2009) accounts for mass losses due to permeation through the membrane wall.

- Mass transport periodicity for flow without permeation.

In spatially repeating fluid domains, the mass fraction field does not become spatially periodic in the same way that velocity does. Instead, the bulk mass fraction along the length of a membrane module gradually increases due to the permeation and removal of low-concentration fluid. This is similar to the situation in a heat exchanger, where the geometry and velocity fields are spatially periodic, but the temperature field is not. Several authors (Carluccio, Starace, Ficarella, & Laforgia, 2005; Li, Feng, He, & Tao, 2006; Rosaguti, Fletcher, & Haynes, 2005) have modelled the temperature problem by utilizing a dimensionless scaled temperature which remains spatially periodic, using the method originally developed by Patankar et al. (Patankar, Liu, & Sparrow, 1977) or variations of this method. In particular, Rosaguti et al. (Rosaguti et al., 2005) applied a variation of the method of Patankar et al. in which only the flow variables but not their gradients are wrapped from the outlet to the inlet. Despite spatial periodicity conditions not being exactly satisfied in membrane modules due to the removal of fluid via permeation, it can still be used as an acceptable approximation if the amount of fluid removed only represents a small fraction of the bulk flow in the feed channel. In addition, this boundary condition becomes easy to implement when combined with the assumption of constant properties and a dissolving wall boundary condition at the membrane surface. Under those assumptions, volumetric flow through the periodic flow domain becomes constant, and velocity periodicity is satisfied (Fimbres-Weihs & Wiley, 2007).

Given the similarities between temperature and mass transport it is possible to specify a non-dimensional scaled mass fraction that remains spatially periodic, i.e. the shape of its profile remains the same for downstream locations separated by a distance equal to the unit cell length, even though the actual mass fraction values are not the same. There are two ways to define a scaled mass fraction with such properties: the first is to assume a constant mass fraction gradient at the membrane wall surface, and the second is by assuming a constant mass fraction value at the membrane wall. For many membrane applications the concentration at the membrane surface reaches a limiting value, such as when the channel becomes gel-polarized in the pressure independent region of the flux-pressure curve or due to a boundary layer of high fixed osmotic pressure or viscosity (Fletcher & Wiley, 2004; Gill, Wiley, Fell, & Fane, 1988; Wiley, Fell, & Fane, 1985). In addition, other works (Lau et al., 2009, Fimbres-Weihs et al., 2006) have shown that for flow rates

such as those commonly found in real-world membrane modules, flow instabilities and flow oscillations appear after just 4 or 5 spacer filaments, which in turn result in a relatively constant average mass transfer coefficient over the rest of the channel length. This makes the latter assumption of a constant wall mass fraction a suitable approximation over much of a SWM.

By analogy to the heat transfer problem, the scaled mass fraction for a constant wall mass fraction then takes the following form (Fimbres-Weihs & Wiley, 2010):

$$\theta = \frac{w_m - w}{w_m - w_b} \qquad (17)$$

Where w_m represents the wall mass fraction value and w_b is the inlet bulk mass fraction value. In membrane systems, because the value of the Sherwood number (and hence the mass transfer coefficient) usually approaches an asymptotic or limiting value as the flow becomes fully developed both in terms of velocity and mass transfer, the choice of wall mass fraction and inlet bulk mass fraction becomes irrelevant. These values still need to be specified in order to calculate the actual mass fraction values used in the calculations, but should have no impact on the mass transfer results obtained, as long as they are not approximately equal, a circumstance that would lead to numerical round-off error (Fimbres-Weihs & Wiley, 2010).

- Mass transport periodicity for flow with permeation

In a work (Darcovich et al., 2009) was proposed a methodology for simulating spatially periodic flows where permeation through the membrane is occurring. Although their model was proposed for 2D-flow fields, it can be extended to 3D cases. This approach is based on the premise that, at points far enough downstream (ds) from the channel inlet such that the velocity and concentration boundary layers have fully developed the profiles of mass flow per unit area of both the solute and the solvent become self-similar. Therefore, for the solute:

$$\vec{N}_{s,us}(y,z) = f_s \vec{N}_{s,ds}(y,z) \qquad (18)$$

Similarly, for the solvent:

$$\vec{N}_{slv,us}(y,z) = f_{slv} \vec{N}_{slv,ds}(y,z) \qquad (19)$$

The methodology of Darcovich et al. involves iteratively setting the upstream (us) boundary conditions by using the competed downstream conditions. Initially, the upstream conditions are set as flat properties (constant and normal to the boundary surface) for both the velocity and mass fraction gradients, using predefined total mass flow rates for both the solvent ($\dot{m}_{slv,us}$) and the solute ($\dot{m}_{s,us}$). In subsequent iterations, the upstream velocity and mass fraction profiles are calculated by using the following expressions:

$$\vec{v}_{us}(y,z) = \frac{\vec{N}_{slv,us}(y,z) + \vec{N}_{s,us}(y,z)}{\rho A_{us}} \qquad (20)$$

$$w_{us}(y,z) = \frac{\vec{N}_{s,us}(y,z) \cdot \hat{u}}{\left[\vec{N}_{slv,us}(y,z) + \vec{N}_{s,us}(y,z)\right] \cdot \hat{u}} \qquad (21)$$

Where the upstream mass flow profiles are calculated using Equation 18 and 19, and the upstream to downstream ratios are calculated by:

$$ft_{slv} = \frac{\dot{m}_{slv,us}}{\int_{S_{ds}} \vec{N}_{slv} \cdot d\vec{A}} \qquad (22)$$

$$ft_{s} = \frac{\dot{m}_{s,us}}{\int_{S_{ds}} \vec{N}_{s} \cdot d\vec{A}} \qquad (23)$$

Although this model is claimed to be applicable for cases with variable fluid properties, it cannot be used for cases where the permeation rate is determined by gel layer resistance. Moreover, it is unclear whether this model will give accurate results for cases where the range of concentrations as the membrane surface is quite large, and the osmotic pressure dependency with concentration is highly non-linear, a situation often encountered in membrane systems (Fimbres-Weihs & Wiley, 2010).

3.3. Literature review

In this review, have been considered investigations of the previous 20 years divided in 4 periods of 5 years. In the first period (1995-2000), CFD allowed include detailed models for the osmotic pressure and intrinsic rejection. Even different discretization schemes and grids were studied.

In the second period (2000-2005) different obstacles were simulated to improve the mass transfer coefficients and to analyze the periodicity of the mass transfer in the channels. It was also solved a system with periodic boundary conditions and permeable walls, so the downstream values of velocity and concentration had to be corrected to obtain the upstream values to the second simulation of the periodic domain, and so on.

In the third period (2005-2010) different obstacles were also investigated. It was also determined a correlation to calculate the mass transfer coefficients for high mass transfer rates, using the mass transfer coefficients from low or null mass transfer rates. This allowed for instance calculate concentration polarization simulating the system with impermeable dissolving-walls, with a considerable decrease of necessary process parameters. Regarding the periodic conditions, in some cases they were used with permeable walls, so it was necessary to recalculate the velocity and concentration, but in other cases they were used with impermeable walls, so the velocity did not need a recalculation although the mass needed a source term to avoid solute mass fraction changes due to mass transfer at the walls in the periodic domain.

In the last period (2010-2015) the entrance effects on Sherwood number were investigated with different obstacles. The periodicity in the channel depends on whether the entrance effects are important. In one study the Sherwood number was calculated with impermeable dissolving-walls and the mass transfer coefficients were recalculated with the mass transfer correction correlation, and finally results were compared with a detailed CFD simulation with permeable walls. In this work the conditions did not be periodic but the periodicity was studied. In another investigation an alternative to reverse electrodialysis was proposed, a corrugated membrane which was simulated with periodic conditions.

3.3.1. First period (1995-2000)

Sherwood-number relations were obtained for laminar flow-regime (De & Bhattacharya, 1997). The common flow-modules, namely, rectangular channel, tubular and radial cross-flow were considered. The relations included the effect of suction through the membrane, and they were compared with the standard correlations to quantify the effect of the suction. The Sherwood relations were used with the osmotic-pressure model to predict the permeate flux in RO and UF. The equations were solved without CFD and results were compared with experimental data.

It was proposed and experimentally validated a numerical model to predict laminar flows hydrodynamics and concentration polarization of salt aqueous solutions in a slit (V. M. Geraldes, Semião, & de Pinho, 1998). The physical modeling for flow and mass transfer process was incorporated with the osmotic pressure equation and a variation law for the membrane intrinsic rejection coefficient. A new mass transfer correlation was proposed. Periodic boundary condition did not be considered, and the model was solved considering the walls as permeable.

Later on it was proposed a mathematical model to predict the concentration polarization in nanofiltration / reverse osmosis (Vítor Geraldes, Semião, & Norberta Pinho, 2000). The convection and diffusive terms of those equations were discretized by different schemes for comparison purposes. Also different grids are used within the best scheme (hybrid scheme) to evaluate the model sensitivity to the grid refinement. The model was experimentally validated. Periodic boundary condition did not be considered, and the model was solved considering the walls as permeable.

3.3.2. Second period (2000-2005)

It was presented and experimentally validated a numerical model based on the finite volume formulation to predict laminar flows hydrodynamics and mass transfer of aqueous solutions ($570 < Sc < 3200$) in the feed channel of spiral-wound and plate-and-frame systems (Vítor Geraldes, Semião, & De Pinho, 2001). Parameters pertaining to the solute transport inside the membrane are incorporated on the mass transfer boundary condition at the membrane surface. It was proposed a correlation for the concentration boundary layer thickness, a measure of the concentration polarization, in the operating condition ranges of $250 < Re < 1000$, $0.02 < Re_p < 0.1$ and $800 < Sc < 3200$. Periodic boundary condition did not be considered, and the model was solved considering the walls as permeable.

As boundary conditions in CFD to predict the solute concentration at the feed side of the membrane surface were used experimental data of permeate solute concentrations and fluxes, generated by NF test in a NF laboratory slit (De Pinho, Semião, & Geraldes, 2002). It was determined the experimental intrinsic rejection coefficients as a function of the transmembrane pressure. Neutral and salt solutions were studied. Periodic boundary condition did not be considered, and the model was solved considering the walls as permeable.

It was studied the hydrodynamic and mass transfer of an aqueous solution inside a slit filled with ladder-type spacers (Vítor Geraldes, Semiao, & De Pinho, 2004). The upper-wall was impermeable and the lower-wall was semipermeable. Two different cases were simulated: 1. Filaments adjacent to the membrane; 2. Filaments adjacent to the impermeable wall. Results were validated against experimental values. The results showed that the average concentration polarization for case 1 does not depend on the distance to the inlet as filaments disrupt periodically the concentration boundary layer and, therefore, the concentration polarization is controlled. A cleaning effect was exhibited for case 2, but the concentration polarization increased uninterruptedly as the concentration boundary layer grows continuously across the channel length. The numerical predictions were restricted to a single channel segment contained between two consecutive and transversal filaments in the case 1. The boundary conditions were corrected and wrote as quasi-periodic type with a water recovery factor corresponding to the channel segment. The model was solved considering walls as permeable.

3.3.3. Third period (2005-2010)

It was investigated the impact of spacer configurations (cavity, zigzag and submerged) and mesh length on the alleviation of concentration polarization and the enhancement of permeate flux in crossflow RO membrane channels (Ma & Song, 2006). It was demonstrated that the average permeate flux could be significantly enhanced by the spacers, especially those with zigzag configurations. The results suggested that

different mesh length should be used in membrane modules for feed waters of different salinities to obtain the maximum permeate flux enhancement. Periodic boundary condition did not be considered, although the periodicity of the results was analyzed. The model was solved considering the walls as permeable.

It was obtained a mass transfer correction factor to account for the suction effect in nanofiltration/reverse osmosis membrane modules (Vitor Geraldes & Afonso, 2006). This factor corrects conventional mass-transfer coefficients at vanishing mass-transfer rates obtained through CFD of a symmetrical NF/RO rectangular channel with fully developed laminar flow. It was found that the correction factor depends only on $\mathrm{Pe}/\mathrm{Sh}^0$ through a correlation, where Pe is the permeation Peclet number and Sh^0 is the Sherwood number for impermeable walls. This correlation was used to predict the average concentration polarization index. The correlation was suitable to determine mass-transfer coefficients at high mass-transfer rates, independently of the membrane module geometry and the flow regime.

The flow structure and solute concentration distribution was determined in a NF/RO plate-and-frame module with radial thin feed channels that had considerable entrance and outlet effects (Cavaco Morão, Brites Alves, & Geraldes, 2008). The CFD simulations showed important 3D effects and allowed the determination of a mass-transfer correlation at vanishing mass-transfer rates. It was validated that the previously developed factor correction for high mass-transfer rates is still valid in the module here investigated, although a semipermeable wall was used (with low permeate flux). In order to apply the periodic boundary condition, a numerical artifact was introduced to keep the average concentration of solutes constant. This artifact consisted in a source of solvent at the outlet surface balancing the mass of permeated solvent.

It was described a numerical methodology for the simulation of flow and mass transfer in spacer-filled channels presenting high Schmidt numbers (Santos, Crespo, & Geraldes, 2010). It was used a source term related to an artificial production of mass in the computational domain necessary to avoid solute mass fraction depletion due to mass transfer at the walls. The results in the mass transfer correlation were a contribution to other results obtained by other authors in a smaller range of Sc number. Periodic boundary conditions were considered, and the membrane surfaces were considered as impermeable dissolving walls.

3.3.4. Last period (2010-2015)

Entrance effects on the Sherwood number were investigated experimentally in a narrow rectangular channel, both with ribbed walls and filled with different mesh-type spacers, for Reynolds numbers varying between 5 and 500 (Rodrigues, Geraldes, de Pinho, & Semião, 2012). The results showed that for laminar flow regime strong entrance effects can prevail over the entire channel whenever transverse or oblique filaments are not in contact with the electrodes. Diamond-shape spacers presented no mass-transfer entrance effects even at a lowest Reynolds number. This work gave guidelines for using periodic boundary conditions in the CFD simulation of flow and mass-transfer inside spacer-filled narrow rectangular channels with relevance for modules design. For laminar flow, the periodic boundary condition must be used with care because strong mass-transfer entrance effects may be found, depending on the spacer geometry.

In this work it is used a semi empirical approach where the average and local mass transfer coefficients are first computed by CFD using dissolving walls (C. Completo, Semiao, & Geraldes, 2016). Then the coefficients are used to predict the local and average concentration polarization for semi-permeable membrane, using a mass transfer correction factor correlation. In this way is possible to design new feed channels with bas-reliefs similar to the ones that have been developed for efficient passive micromixers, with a complex 3D shape. The results were validated using detailed CFD simulations (semi-permeable wall) and with experimental concentration distribution obtained by holographic interferometry. Local and average concentration polarization and permeate flux were predicted with an error less than 15%. The model was solved without periodic boundary conditions, but the periodic behavior of the concentration field was assessed by using both the $\overline{w}_{Am,loc}/\overline{w}_{Ab,loc}$ and $w_A/\overline{w}_{Ab,loc}$ ratios.

As implementation of reverse electrodialysis (RED) is economically limited by the relatively high ion-exchange membrane price, this work develops a promising alternative, the utilization of profiled membranes, since the reliefs formed on their surface keeps the membranes separated and provides channels for solution flow (Pawlowski, Geraldes, Crespo, & Velizarov, 2016). The highest net power density values were obtained for corrugations shape and arrangement in a form of chevrons due to the increase of the available membrane area and a good balance between enhancement of mass transfer and the increase of the pressure drop. The proposed design could be extended to other electromembrane process. The CFD simulations were performed using periodic inlet/outlet boundary conditions until the concentration distribution in the channel becomes quasi-periodic. A mass source term is used to replace the mass transferred out of the periodic channel segment through a specified membrane area. The walls were considered with a constant concentration and type permselective, so only the respective counter-ions are transported from a concentrated to a diluted stream through the membranes. The mass flux in the system is only of Coulombic nature.

3.4. Model

3.4.1. Conservation laws

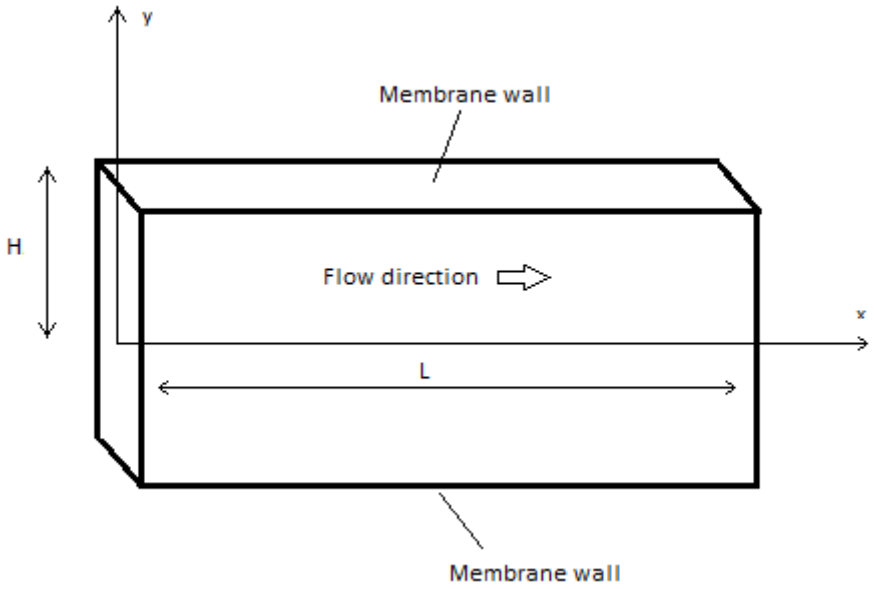


Figure 8. Control volume of the duct

Mass conservation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Momentum conservation in the x-direction and y-direction:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \upsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \upsilon\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

Species conservation:

$$\frac{\partial w_A}{\partial t} + u\frac{\partial w_A}{\partial x} + v\frac{\partial w_A}{\partial y} = \frac{D_{AB}}{\rho}\left(\frac{\partial^2 w_A}{\partial x^2} + \frac{\partial^2 w_A}{\partial y^2}\right)$$

Boundary conditions:

At $t = 0, x > 0, y > 0$

$u = 0$

$v = 0$

$w_A = 0$

$p = p_0$

At $x = 0$ and $0 > y > H$

$u = a \cdot u_0[1 - (y/H)^2]$

$a = 3/2$ if parallel plates

$a = 2$ if circular duct

$v = 0$

$w_A = w_{A0}$

$p = p_0$

At $x = L$ and $0 > y > H$

$\partial u/\partial x \approx 0$

$\partial v/\partial x \approx 0$

$\partial w_A/\partial x \approx 0$

$p = p_0$

At $y = 0$ and $x > 0$

$\partial u/\partial y = 0$

$v = 0$

$\partial w_A/\partial y = 0$

$\partial p/\partial y = 0$

If semi-permeable wall:

At $y = H$ and $x > 0$

$$u = 0$$

$$v = J_p = L_p\left[\Delta p - \sigma\left(w_{Am} - w_{Ap}\right)\right] = L_p(\Delta p - \sigma w_{Am} R_{real})$$

$$R_{real} = \left(w_{Am} - w_{Ap}\right)/w_{Am} \qquad (24)$$

$$\partial w_A/\partial y = \left(J_p w_{Am} R_{real}\right)/D_{AB}$$

If impermeable dissolving wall:

At $y = H$ and $x > 0$

$$u = 0$$

$$v = 0$$

$$w_A = w_{Am} = \text{cte or } \partial w_A/\partial y = \text{cte}$$

3.4.2. Conservation laws with periodic boundary conditions

If the system is a periodic domain, some modifications in the equations are necessary.

Momentum conservation in the x-direction:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \upsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{1}{\rho}\nabla p$$

To restrict the computation domain to a single periodic segment of the channel, the pressure has to be decomposed as (Pawlowski et al., 2016):

$$p = p^* + \beta \cdot x$$

Where p^* is the periodic component of the pressure, β is the average pressure gradient in the channel and x is the local x-coordinate.

Species conservation:

$$\frac{\partial w_A}{\partial t} + u\frac{\partial w_A}{\partial x} + v\frac{\partial w_A}{\partial y} = \frac{D_{AB}}{\rho}\left(\frac{\partial^2 w_A}{\partial x^2} + \frac{\partial^2 w_A}{\partial y^2}\right) + S$$

$$S = \frac{-D_{AB}\iint_{A_m}(\partial w_A/\partial y)_m dA}{\iiint_V |u| w_A dV} \cdot |u| \cdot w_A$$

The field S represents the mass source term, which is used to replace the mass transfer outside (or inside) the periodic channel segment through a specified membrane area (A_m). Since each channel segment is divided in small control volumes and S depends on the position (x, y, z) inside the channel segment, according to (Pawlowski et al., 2016), the solute is preferentially re-injected into regions with higher velocity, while the concentration close the membrane surface is almost not affected by such a numerical approach. In such a way, it is possible to solve the continuity equation for an element of fluid that enters in the channel and moves with the fluid average velocity.

3.4.3. Mass transfer parameters

- Mass transfer coefficient

The local mass transfer coefficient at each membrane point is computed as follows:

$$k_{c,loc} = \frac{1}{w_{Ab} - w_{Am}} \left(-D_{AB} \frac{\partial w_A}{\partial y} \right)_m \qquad (25)$$

Where w_{Ab} and w_{Am} are respectively solute concentration in the bulk and at membrane surface.

The local Sherwood number is calculated as follows:

$$Sh_{local} = \frac{k_{c,loc} \cdot D_{AB}}{\upsilon} \qquad (26)$$

$$Sh(x) = \frac{1}{A_m} \int_{A_m} Sh_{loc} \, dA \qquad (27)$$

- Estimation of concentration polarization with the Film Theory method

The Film Theory Model assumes that axial (crossflow) solute convection near the membrane surface is negligible, and describes the concentration profile by a one-dimensional mass balance equation (Keir, 2012).

The mass transfer coefficient in this case is defined as:

$$k_c = \frac{1}{w_{A0} - w_{Am}} \left(-D_{AB} \frac{\partial w_A}{\partial y} \right)_m \qquad (28)$$

k_c is the mass transfer coefficient from impermeable systems. Two types of empirical correlations are frequently used to obtain k_c: correlations from mass transfer data in impermeable soluble systems (uniform surface concentration) and correlation from momentum or heat transfer data in impermeable systems, coupled with theoretical mass/heat/momentum analogies (uniform temperature or uniform heat flux). The use of the film equation with mass transfer coefficient data from impermeable systems has been theme of discussion in the literature (Miranda & Campos, 2002).

A solute mass balance gives:

$$J_p \cdot w_A = J_p \cdot w_{Ap} + D_{AB} \cdot \frac{dw_A}{dy}$$

$$\begin{cases} y = H \rightarrow w_A = w_{Am} \\ y = 0 \rightarrow w_A = w_{A0} \end{cases}$$

Integrating the mass balance:

$$J_p = k_c \cdot \ln\left(\frac{w_{Am} - w_{Ap}}{w_{A0} - w_{Ap}}\right) \qquad (29)$$

Defining the term ϕ as:

$$\phi = J_p / k_c \qquad (30)$$

And combining Equation 24, Equation 29 and Equation 30, gives:

$$\phi = \ln\left[\frac{w_{Am} \cdot R_{real}}{w_{A0} - w_{Am}(1 - R_{real})}\right] \qquad (31)$$

Clearing solute mass fraction at the membrane wall:

$$w_{Am} = \frac{w_{A0}\exp(\phi)}{R_{real} + (1 - R_{real})\exp(\phi)} \qquad (32)$$

As it can be seen in Equation 32, once the permeate rate and the mass transfer coefficient are known, it is possible to determine solute mass fraction at the membrane wall.

The concentration index can be calculated as:

$$M = \frac{w_{Am} - w_{A0}}{w_{A0}} \qquad (33)$$

To estimate the permeate rate it is necessary an iterative process which consists in combining Equation 14 and Equation 29, so:

$$L_p(\Delta p - \sigma w_{Am} R_{real}) = k_c \cdot \ln\left(\frac{w_{Am} - w_{Ap}}{w_{A0} - w_{Ap}}\right) \qquad (34)$$

From Equation 34, solute mass fraction at the membrane wall can be calculated iteratively, and then the permeate rate with Equation 14 or Equation 29.

- Estimation of concentration polarization with the mass transfer correction factor Ξ method

A correction factor for nanofiltration and reverse osmosis was proposed in the literature (Vitor Geraldes & Afonso, 2006). This factor allows calculate concentration polarization taking into account the suction effect (permeation) for a wide range of operating conditions. The correlation obtained for the correction factor can be used to predict the average concentration polarization index.

A solute mass balance in the vicinity of the membrane gives:

$$J_p w_{Am} = D_{AB} \left.\frac{\partial w_A}{\partial y}\right|_m + J_p w_{Ap} \qquad (35)$$

It is considered a modified mass transfer coefficient $k_{c,p}$ which takes into account the suction effect:

$$k_{c,p} = \Xi \cdot k_c \qquad (36)$$

Where k_c is the mass transfer coefficient from impermeable walls, and Ξ the correction.

Equation 36 together with Equation 35 gives:

$$J_p = k_{c,p}\frac{(w_{Am} - w_{A0})}{(w_{Am} - w_{Ap})} \qquad (37)$$

Equation 30 together with Equation 37 and Equation 24 gives:

$$J_p/k_{c,p} = \phi/\Xi = \frac{(w_{Am} - w_{A0})}{(w_{Am} - w_{Ap})} = \frac{(w_{Am} - w_{A0})}{w_{Am}R_{real}} \qquad (38)$$

Clearing wAm:

$$w_{Am} = \frac{w_{A0}}{1 - R_{real}\cdot(\phi/\Xi)} \qquad (39)$$

Equation 39 together with Equation 33 gives:

$$M = \frac{R_{real}}{\Xi/\phi - R_{real}} \qquad (40)$$

By multiplying both terms of Equation 40 by d_h/D_{AB}, where d_h is the characteristic length or hydraulic diameter and D_{AB} is the mass diffusivity, the following equation is obtained:

$$M = \frac{R_{real}}{Sh/Pe - R_{real}} \qquad (41)$$

Sh is the average Sherwood number and Pe is the average permeation Peclet number.

According to Equation 41, the ratio Sh/Pe must be > 1 such that the concentration polarization indexes are not negative. On the other hand, as the permeation Peclet number increases, w_{Am} increases, and given that R_{real} is assumed to be constant, w_{Ap} increases. When Pe becomes very large, the polarization concentration index M reaches a maximum: $M_{max} = R_{real}/(1 - R_{real})$. Therefore, Equation 41 requires that $Sh/Pe \rightarrow 1$ as $Pe \rightarrow \infty$. Besides these two restrictions of Sh/Pe, another one results from the fact that the Sherwood number becomes independent of the permeation velocity (and therefore of Pe), for low mass-transfer rates (low Pe) at the membrane surface. Defining Sh^0 as the Sherwood number for vanishing mass-transfer rates at the membrane surface or impermeable dissolving wall, $Sh \rightarrow Sh^0$ as $Pe \rightarrow 0$. A postulated correlation that satisfies the aforementioned restrictions is (Vitor Geraldes & Afonso, 2006):

$$Sh = Pe + \frac{Sh^0}{[1 + \xi(Pe/Sh^0)]^{c_1}} \qquad (42)$$

Where c_1 is a positive fitting parameter and ξ is a function that satisfies the asymptotic conditions $\xi \rightarrow 0$ as $Pe \rightarrow 0$ and $\xi \rightarrow \infty$ as $Pe \rightarrow \infty$. An appropriate function, $\xi = c_2(Pe/Sh^0)^{c_3}$, with positive fitting parameters c_2 and c_3, can be selected by trial and error. By defining $\phi = Pe/Sh^0$ and $\Xi = Sh/Sh^0$ as the correction factor for Sh^0 to obtain Sh at high mass-transfer rates (that is, to take into account the effect of suction in NF/RO systems), the following equation is obtained after diving Equation 42 by Sh^0 (Vitor Geraldes & Afonso, 2006):

$$\Xi = \phi + \frac{1}{[1 + \xi(\phi)]^{c_1}}$$

$$= \phi + \frac{1}{[1 + c_2\phi^{c_3}]^{c_1}} \qquad (43)$$

Thereby, the average Sherwood number for a given membrane module consists of the product of Sh^0 by the correction factor Ξ. Once Sh^0 and Ξ are known, the concentration polarization index can easily be determined by Equation 40.

Once M is determined, w_{Am} can be calculated with Equation 33, and also be compared with the results from semi-permeable walls simulations.

A method to calculate the permeate rate, if it is not known from experimental work, it is again an iterative process. Permeate rate can be calculated with Equation 14 and with Equation 37, so:

$$L_p(\Delta p - \sigma w_{Am} R_{real}) = k_{c,p} \frac{(w_{Am} - w_{A0})}{(w_{Am} - w_{Ap})} \qquad (44)$$

From Equation 44, solute mass fraction at the membrane wall will be calculated iteratively, and also the permeate rate. This iteration is more complicated than in the Film Theory because $k_{c,p}$ depends of the permeate rate value.

In this work, the parameters c_1, c_2 and c_3 have been determined in order to compare this method with the Film Theory resolution.

3.4.4. The Graetz Problem

Convective heat and mass transfer have been usually treated from an analytical point of view and although the analytical approach is very meaningful, it may not offer a practical solution to every problem. There are many situations for which no mathematical models have as yet been successfully applied. Even in those cases for which an analytical solution is possible, it is necessary to verify the results (Welty, Wicks, Wilson, & Rorrer, 2008).

The resolution of mass transfer with laminar flow forced convection inside a duct will be used in this work to validate the developed model in different geometries.

The first analytical solution for laminar flow forced convection inside tubes was formulated by Graetz in 1885. The assumptions basic to the Graetz solution are as follows (Welty et al., 2008):

1. The velocity profile is parabolic and fully developed before any energy exchange between the tube wall and the fluid occurs
2. All properties of the fluid are constant
3. The surface temperature of the tube is constant at a value Ts during the energy transfer

Figure 9 presents the results graphically for two different boundary conditions at the wall, (1) a constant wall temperature and (2) uniform heat input at the wall. It can be applied to solute mass fraction instead of temperature.

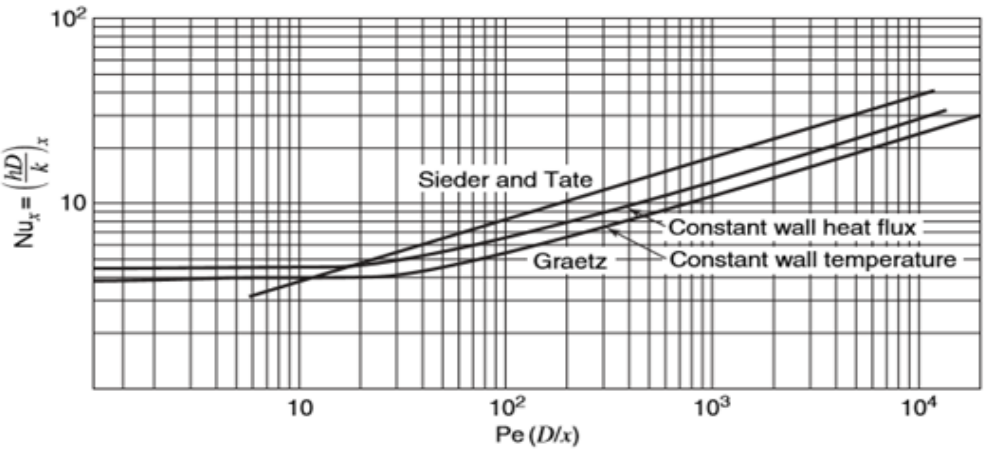


Figure 9. Variation in the local Nusselt number for laminar flow in tubes (Welty et al., 2008)

The analytical results approach constant limiting values for large values of x. A correct simulation of the mass transfer process inside a duct must satisfy the analytical solution of the Graetz problem.

Different geometries were simulated in this works and their results (Sh vs Pe(d_h/x)) were compared with results from the literature.

Objectives

The objective of this work is the simulation employing Computational Fluid Dynamics (CFD) of the membrane separation process, in order to improve knowledge of the process and optimize the operation.

This general objective is materialized in the following partial objectives:

1. Learning of the use of OpenFOAM, with C++ code in Linux distribution, and also the rest of the needed softwares.

2. Formulation in C++ code and implementation in OpenFOAM of a suitable mathematical model which takes into account the momentum transfer and mass transfer in membranes.

3. Implementation of periodic boundary conditions to the model in OpenFOAM.

4. Validation of the model in different ducts, in 2D and 3D, with periodic and no periodic boundary conditions, comparing the results with literature.

5. Simulation of a semi-permeable membrane with periodic boundary conditions and analysis of the concentration polarization at the membrane surface.

6. Determination of the mass transfer correction factor Ξ, which allows estimate concentration polarization from the simulation of an impermeable dissolving wall domain with periodic boundary conditions.

7. Comparison between the mass transfer correction factor Ξ method and the well-known Film Theory method.

8. Study of the effects of several variables in concentration polarization:

    a. Effect of geometry duct
    b. Effect of spacers
    c. Effect of transmembrane pressure according to geometry and spacers

Methodology

1. Softwares used

1.1. OpenFOAM version 3.0.1.

CFD simulations can be performed using a variety of software packages. Most of them are only commercially available and hard coded to apply to a fixed range of problems. In this work the software package OpenFOAM® was used. OpenFOAM (Open Field Operation and Manipulation) is an open source C++ toolbox of pre-written solvers and utilities in a framework that is completely open, readily extensible and can be customized as needed. It is therefore a very powerful platform for performing any kind of numerical simulation in continuum mechanics. Since the OpenFOAM default library provides no model capable of including mass transport of solute in the flow, the model was introduce in this work (Aslak, 2014)

As OpenFOAM comes stripped of any graphic user interface, users will thus have to get acquainted with the use of *Terminal* on either a Mac or Linux machine. In this work the simulations were processed on Linux operating system (Intel® Core™ i5-4200M CPU 2.50GHz x 2).

Figure 10 shows the general structure of OpenFOAM. The library is primarily used to create *applications* which are executables that fall into two categories: *utilities* that perform pre- and post-processing tasks, involving data manipulation and algebraic calculations, and *solvers* that can be executed on fully pre-processed cases to solve specific problems, i.e. run simulations (Aslak, 2014).

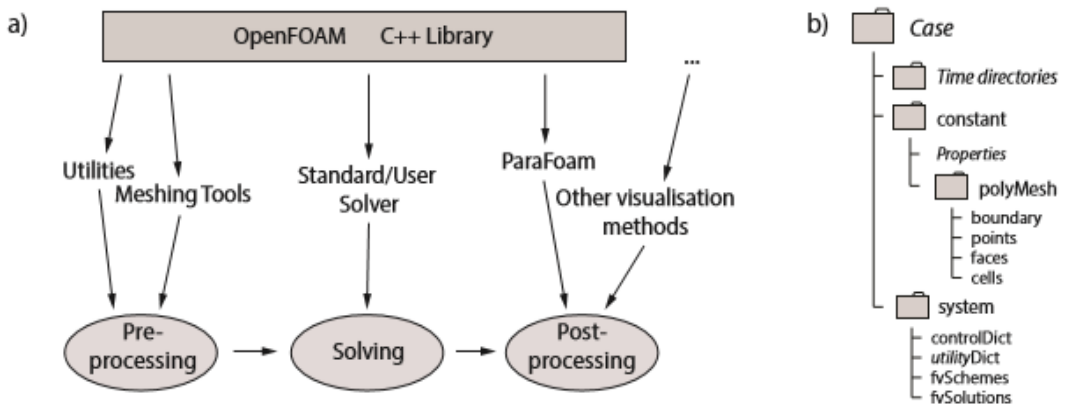


Figure 10. a) An overview of the OpenFOAM structure. b) Example of a case directory. Cursive letter in the figure indicates type/class of folders or files, not file name (Aslak, 2014).

When working with a specific case in OpenFOAM the user will typically navigate in a systematic network of folders containing the files that define the problem at hand, such as it is illustrated in Figure 10.b. The necessary files for a case to run are shown in the figure. The system-directory must contain a `controlDict` which is a dictionary of time-specific user defined information such as end time, time step size, write interval, etc. Furthermore, it must contain the two files `fvSchemes` and `fvSolutions` which specify the mathematical methods used by the solver. In the constant-directory physical properties such as gravitation or, as in this work, fluid and membrane transport properties are specified. The constant/polyMesh-directory contains geometric information about the mesh in a points file that specify point locations in the mesh, and a faces file that use the points to define the faces of the mesh. Furthermore, in OpenFOAM, since each face is shared by two CVs, one of them is denoted “owner” and the other “neighbor”, with a vector pointing from the face of the neighbor into the volume of the owner. The two files neighbor and owner are therefore also included in the polyMesh directory to account for this. All mesh data is specified in array form such that specific points, faces and cells have array numbers. By face number referencing, the boundary file defines the problem domain boundaries. Finally it is necessary to specify a 0 time directory, which contains initial

conditions at boundaries of the problem domain (Aslak, 2014). For the model used, it was required to have files U, p and wA specifying the initial conditions for velocity, pressure and solute mass fraction, respectively. However, in general this folder will include any variable that changes with time.

In overview, simulation in OpenFOAM can be performed in the following steps (Aslak, 2014):

1. Setting up the case directory with necessary folders and files. A common way to do this is by copying and altering files from similar tutorial cases given in the OpenFOAM library.
2. Creating a mesh using `blockMesh`, `snappyHexMesh` or other relevant utilities or softwares.
3. Executing utilities on the case, e.g. to create specific boundaries, set initial fields, refine the mesh in certain regions, etc.
4. Decomposing the case into separate domains using the `decomposePar` utility such that a solver can be run in parallel on multiple processors.
5. Executing the solver.
6. Reconstructing the decomposed case using the `reconstructPar` utility.
7. Visualizing the results by executing `paraFoam` which launches the third-party visualization software ParaView and loads the time directories.

Step 4 and 6 are not necessary for the simulation to run, they just offer a neat way of shortening solution time –at the cost of computational power. Using decomposition methods to perform tasks faster can also be applied in step 2 when creating the mesh and in step 3 for faster execution of certain utilities (Aslak, 2014).

1.2. ParaView version 4.4.0.

This is the main post-processing software distributed with OpenFOAM. It is an open-source data analysis and visualization tool and it can either be invoked by using paraFoam wrapper script or by converting the data to VTK format and open with ParaView. Data processing can be done either interactively in a 3D environment or using command line batch processing.

1.3. SolidWorks 2016

SolidWorks is a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) computer program that runs on Microsoft Windows. This tool helps to create 2D or 3D solid models without any complexity, faster and in the cost effective way. SolidWorks allows create .stl fies with the membrane design.

1.4. Swak4foam

Swak4foam stands for “Swiss Army Knife for Foam” and is a set of libraries and utilities to perform various kinds of data manipulation and representation. It can be used to set complicated boundary conditions depending on expressions, for instance in this work the concentration gradient in semi-permeable case.

1.5. cfMesh version 1.1.1.

cfMesh is a cross-platform library for automatic mesh generation that is built on top of OpenFOAM®. cfMesh supports various 3D and 2D workflows, built by using components from the main library, which are extensible and can be combined into various meshing workflows. The core library utilizes the concept of mesh modifiers, which allows for efficient parallelization using both shared memory parallelization (SMP) and distributed memory parallelization using MPI. In addition, special care has been taken on memory usage, which is kept low by implementing data containers (lists, graphs, etc.) that do not require many dynamic memory allocation operations during the meshing process (Juretic, 2010).

In this work cfMesh allows generate the mesh in the cases of more complex geometry - membrane with spacers - instead of use the OpenFOAM mesh generator.

2. Simulation

The broad strategy of CFD is to take a continuous domain representing a known problem and replace it with a discrete domain to create a finite amount of grid locations. It is then possible to solve the equations throughout the whole domain, only at selected points, which can be interpolated to give values at any location.

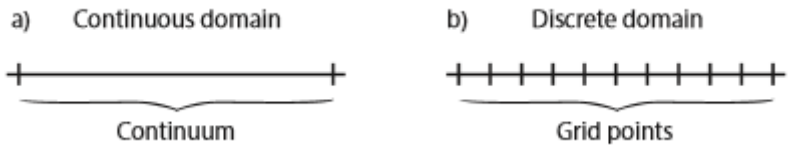


Figure 11. Illustration of the difference between a) a continuous domain and b) a discrete domain for a 1D case (Aslak, 2014).

Discretization yields a huge set of algebraic equations that is solved repetitively in a computer, until a steady state is reached. The resulting data can then be analyzed in different ways in order to gain insight about the simulated phenomenon (Aslak, 2014).

For any given case there are three main steps to a simulation process: pre-processing, solving and post-processing. These are described in the following.

2.1. Pre-processing

In this initial step the case is defined such that it can be processed by a solver. In outline the typical procedure is:

1. Defining the geometry
2. Generating the mesh within the volume of the geometry
3. Specifying appropriate boundary conditions for the different surfaces of the domain
4. Setting fluid properties
5. Setting the model according to the flux regime
6. Selecting the time step and other time parameters
7. Defining numerical schemes for each term of the equations
8. Selection of the solvers, tolerances and algorithms

2.1.1. Case geometry

OpenFOAM solves the case in 3 dimensions by default but can be instructed to solve in 2 dimensions by specifying a 'special' empty boundary condition on boundaries normal to the (3rd) dimension for which no solution is required. In this work the cases are solved in 2 dimensions except one of them.

Firstly simulations of three different geometries were conducted. It was simulated only a portion of the total volume, which is the representative portion due to the symmetry, as it can be seen in the following figures. This allows reduce the computational load. The aim of these simulations is the validation of the model solving the Graetz problem. At the walls of the ducts there is a constant solute mass fraction which diffuses to the center of the duct. After a certain distance, a periodic mass transfer behavior is reached. With the simulations it can be observed that the simulated cases comply with the analytical and numerical solutions from literature.

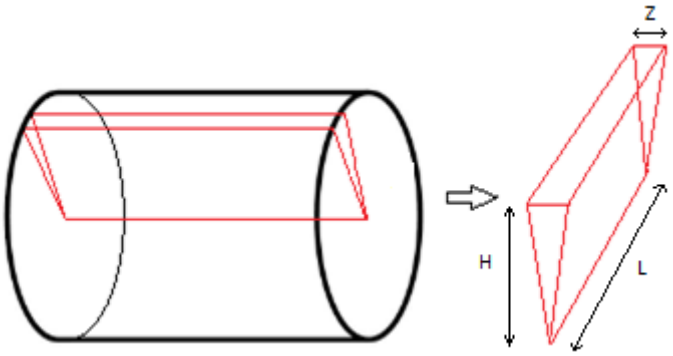


Figure 12. Geometry 1

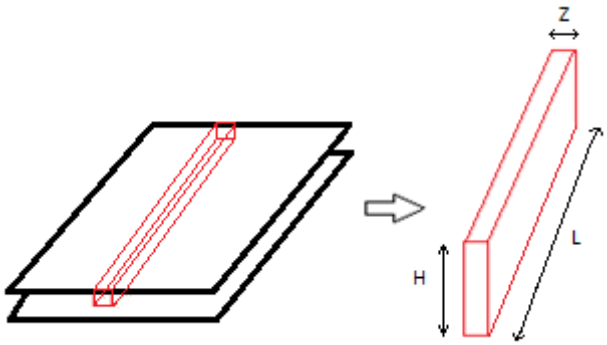


Figure 13. Geometry 2

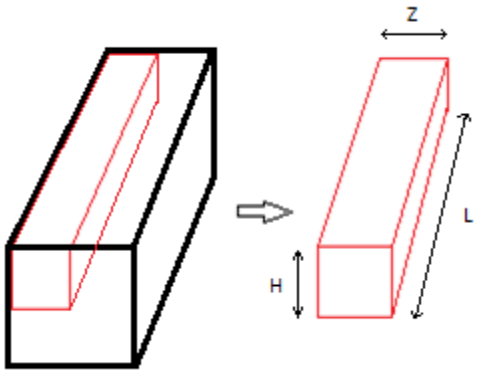


Figure 14. Geometry 3

Geometry 1 and Geometry 2 are 2-dimensional, while Geometry 3 is 3-dimensional. Geometry 1 is equivalent to a circular duct, Geometry 2 to a parallel plates and Geometry 3 to a square duct. The circular duct and parallel plates are the most common ducts in membrane modules (tubular membranes and flat membranes, respectively). Geometry 3 was studied in order to solve a 3-dimensional case and due to validation purposes.

The three geometries were simulated with periodic boundary conditions at the inlet and the outlet. Also they were simulated with no periodic conditions, and in these cases it was simulated all the length of the duct, while with periodic conditions just a part (usually one tenth of the length). The simulation with periodic conditions is equivalent to a periodic domain which is repeated along the length of the duct.

Geometry 1 and Geometry 2 were also simulated as membranes, determining the permeate flux and the other typical membrane parameters. Then, it was also studied Geometry 2 with obstacles or spacers to improve the mass transfer and reduce the problem of concentration polarization and the subsequent flux decline. Rectangular and semi-cylindrical spacers were used (Geometry 4 and Geometry 5, respectively). These geometries can be seen in the following figures.

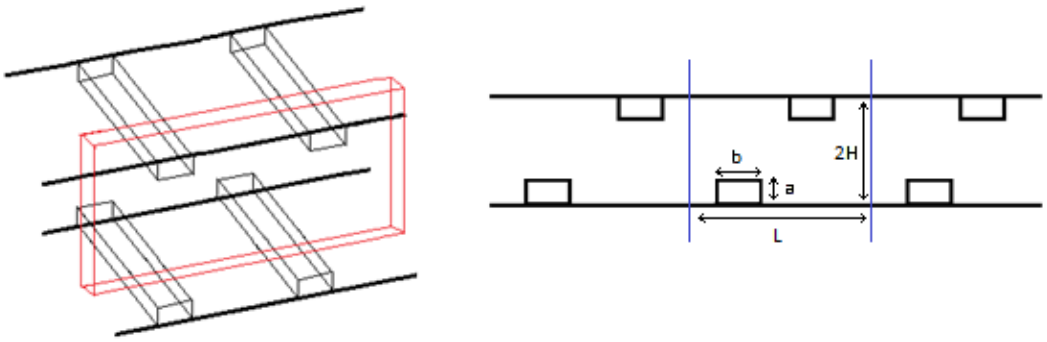


Figure 15. Geometry 4

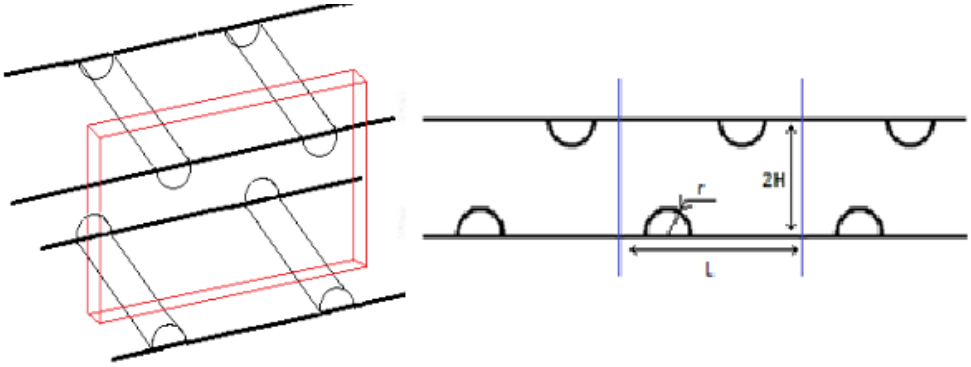


Figure 16. Geometry 5

Geometry 4 and Geometry 5 were simulated with periodic boundary conditions, so a periodic domain was studied, which is equivalent to a real duct with the correct treatment of the boundary conditions and equations.

2.1.2. Mesh generation

The quality of a flow problem solution is governed by the quality of the mesh. In general the finer the grid is, the better solution. It is, however, not wise to make the grid equally fine at all locations as this would demand excessive computing power to solve. The optimal mesh is therefore designed such that it is sufficiently fine in places where detailed flow information is desired and coarse elsewhere. It is noted that, at times, pre-processing can be the most time consuming step of the three, mainly due to difficulties with mesh generation.

For the simulation of the flow and mass-transfer, the extremely thin concentration boundary layer requires a large number of control volumes near the membrane surface to solve the concentrations profiles accurately.

In order to assess the grid independence of the model output, distinct meshes were compared with different cell size and with different refinement near the membrane surface. The results and validation of the mesh is shown in the Results and discussion chapter.

The mesh generator supplied with OpenFOAM, called `blockMesh`, generates meshes from a description specified in an input dictionary, `blockMeshDict` located in the `constant/polyMesh` directory for each given case. After running the `blockMesh` command, the mesh is built and geometry data is stored in the `polyMesh` folder. The principle behind `blockMesh` is to decompose the domain geometry into a set of 1 or more three dimensional, hexahedral blocks. Edge of the blocks can be straight lines, arcs or splines. The mesh is ostensibly specified as a number of cells in each direction of the block, sufficient information for `blockMesh` to generate the mesh data. One example of the `blockMeshDict` entries can be found in Appendix I. The following tables show the main inputs of the `blockMesh` files:

Table 2.Mesh parameters in Geometry 1, Geometry 2 and Geometry 3

	Geom. 1	Geom. 2	Geom. 3
x direction	$4 \cdot 10^{-4}$ m	$2 \cdot 10^{-4}$ m	$5 \cdot 10^{-5}$ m
y direction	$2 \cdot 10^{-4}$ m	$5 \cdot 10^{-5}$ m	$1 \cdot 10^{-4}$ m
z direction	$5 \cdot 10^{-5}$ m	$1 \cdot 10^{-5}$ m	$1 \cdot 10^{-4}$ m
Cells x direction	100	50	40
Cells y direction	100	30	60
Cells z direction	1	1	60
Refinement x direction	1	1	1
Refinement y direction	0.3	0.5	0.5
Refinement z direction	1	1	1.5

Table 3. Mesh parameters in long Geometry 1, long Geometry 2 and long Geometry 3

	Long geom. 1	Long geom. 2	Long geom. 3
x direction	$4 \cdot 10^{-3}$ m	$2 \cdot 10^{-3}$ m	$2 \cdot 10^{-3}$ m
y direction	$2 \cdot 10^{-4}$ m	$5 \cdot 10^{-5}$ m	$1 \cdot 10^{-4}$ m
z direction	$5 \cdot 10^{-5}$ m	$1 \cdot 10^{-5}$ m	$1 \cdot 10^{-4}$ m
Cells x direction	1000	500	500
Cells y direction	100	30	40
Cells z direction	1	1	40
Refinement x direction	1	1	1
Refinement y direction	0.3	0.5	0.5
Refinement z direction	1	1	1.5

With these entries, the number of cells were 10,000 (Geom. 1), 1,500 (Geom. 2), 144,000 (Geom. 3), 100,000 (Long Geom. 1), 15,000 (Long Geom. 2) and 800,000 (Long Geom. 3).

By typing `checkMesh` in the terminal the specific quality measures of the mesh are created in the screen. This command ends with the script `mesh OK` if everything is correct. An example of `checkMesh` file can be found in Appendix II.

All the values that `checkMesh` gives seem to be good for the correct performance of the simulations. The maximum and minimum values of the areas and volumes are positive. Non-orthogonality, or the angle between the center to the center line between to neighbor faces and the normal line from the common border of the faces, is equal to zero, and the maximum skewness remains below a value of 10, which is the critical value for obtaining convergence. Therefore, the values for non-orthogonality and skewness will not affect the stability of the solution.

The figures showing the meshes of the geometries which were generated by `blockMesh` are in Appendix III.

One challenge with the text-based construction of mesh is keeping track of the multiple coordinate points within a complex geometry. This can be negated by using a CAD software to produce the geometry described with STL surfaces. In this work the software used was SolidWorks, useful to create the ducts with obstacles - Geometry 4 and Geometry 5. OpenFOAM can read these STL files and convert them in to mesh, using other software: cfMesh. Firstly, the geometry is design with SolidWorks, then the boundary surfaces are exported as STL files in asci code. Then an unique asci code STL file is done with the surfaces. The dictionary `meshDict` in system directory calls the geometry and generate the mesh. Some of the meshes' parameters are:

Table 4. Mesh parameters in Geometry 4 and Geometry 5

	Geom. 4	Geom. 5
Maximum cell size	$1.84 \cdot 10^{-6}$ m	$1.84 \cdot 10^{-6}$ m
Refinement Levels	1	1
Refinement thickness	$5 \cdot 10^{-6}$ m	$5 \cdot 10^{-6}$ m
Boundary cell size	$5 \cdot 10^{-7}$ m	$5 \cdot 10^{-7}$ m

After the mesh is generated, it is necessary to define the type of patches, which is detailed in the point 2.2.1. Previous steps.

The meshDict file is in Appendix IV.

The figures showing the meshes which were generated by cfMesh are also in Appendix III.

2.1.3. Boundary conditions

In general the boundary faces are an inlet, an outlet, a bottom wall, a top wall and the couple of faces perpendicular to the 3rd direction. For the purpose of applying boundary conditions, a boundary is generally broken up into a set of patches. One patch may include one or more enclosed areas of the boundary surface which do not necessarily need to be physically connected. The patches are called: `inlet`, `outlet`, `topWall`, `bottomWall`, `topEmptyFaces` and `bottomEmptyFaces`. The geometries with obstacles (Geometry 4 and Geometry 5) have also the obstacles as patches –`ObsTop` and `ObsBottom`.

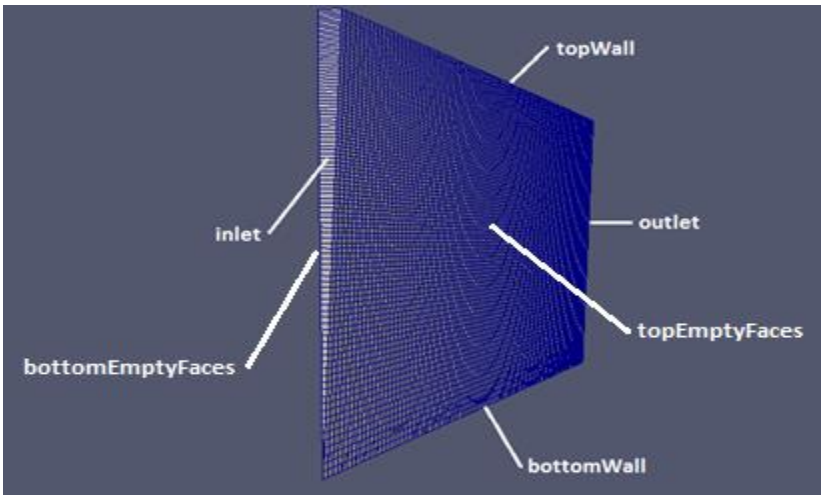


Figure 17. Patches in Geometry 1, Geometry 2, Geometry 3, long Geometry 1, long Geometry 2 and long Geometry 3

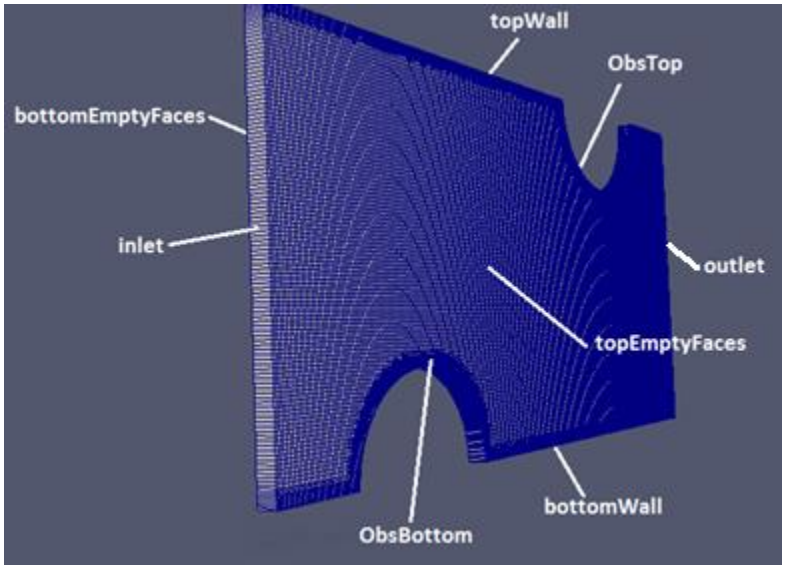


Figure 18. Patches in Geometry 4 and Geometry 5

A type is assigned to every patch as part of the mesh description. The types can be seen in the following tables:

Table 5. Type of patches in Geometry 1, Geometry 2, Geometry 3, Geometry 4 and Geometry 5

	Geom. 1	Geom. 2	Geom. 3	Geom. 4	Geom. 5
Inlet	cyclic	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic	cyclic
topWall	wall	wall	wall	wall	wall
bottomWall	empty	symmetryPlane	symmetryPlane	wall	wall
TopEmptyFaces	wedge	empty	symmetryPlane	empty	empty
BottomEmptyFaces	wedge	empty	wall	empty	empty
ObsTop	-	-	-	wall	wall
ObsBottom	-	-	-	wall	wall

Table 6. Type of patches in long Geometry 1, long Geometry 2 and long Geometry 3

	Long Geometry 1	Long Geometry 2	Long Geometry 3
Inlet	patch	patch	patch
Outlet	patch	patch	patch
topWall	wall	wall	wall
bottomWall	empty	symmetryPlane	symmetryPlane
TopEmptyFaces	wedge	empty	symmetryPlane
BottomEmptyFaces	wedge	empty	wall

If a patch is type `cyclic`, means that periodic conditions are applied to that patch, and if it is `empty`, means that is not important because is perpendicular to the direction where the values do not change (in a 2D simulation, the 3rd direction). If the patch is `patch` type, means that is an inlet or an outlet. For 2D axi-symmetric cases, e.g. Geom. 1, the geometry is specified as a `wedge` of small angle (e.g. 1°) and 1 cell thick, running along the center line, straddling one of the coordinate planes, and the axi-symmetric wedge planes must be specified as separate patches of `wedge` type. The `wall` and `symmetryPlane` patch types means what their names imply. The types of patch are in the `blockMesh` file for the geometries generated by

`blockMesh` and in the boundary file for the geometries generated by cfMesh, in Appendix I and Appendix V, respectively.

In OpenFOAM, it is necessary to specify the initial field values and boundary conditions. All the values for these fields are stored in the `0` folder as text files. From earlier in the mesh generation process it is important to remember which boundary entries are defined as a `patch`, as a `wall`, as a `symmetryPlane`, as an `empty` or as a `cyclic` type.

The initial values and boundary conditions are detailed in the Results and Discussion chapter. The main characteristics are:

- Solute mass fraction

The simulations with dissolving walls use a `fixedValue` at the `topWall`, while the simulations in which the permeation is studied use the value of the solute mass fraction gradient, which is provided by groovyBC, a utility of the software swak4foam. The condition of `zeroGradient` in the outlet and at the obstacles means that there is not transfer of mass at those boundaries.

- Velocity

The value of the velocity at the walls and obstacles is set to 0. The condition of `zeroGradient` is applied in the outlet, and a `fixedValue` is considered in the inlet.

- Pressure

Since the velocity is fixed in the inlets and the walls, the dynamic pressure is set to `zeroGradient`, so that the gradient normal to the faces is zero, which enables the pressure to float as the velocity is fixed. The outlet boundary condition for the pressure is set to 0, except in the case of permeation, where is higher.

In the simulations with periodic conditions, the initial values of the variables are established in the `internalField`. If the conditions are not periodic, the initial values of the variables are indicated in the inlet, and the initial values in the `internalField` are set to 0.

2.1.4. Fluid and membrane properties

The fluid properties are defined in the `transportProperties` dictionary, and can be viewed in Apendix VIII. The keyword `transportModel` is set to `Newtonian`. A Newtonian fluid is characterized by a constant kinematic viscosity which is kept unchanged with the rate of deformation. The fluid is considered water and typical properties have been selected.

The parameters of the membrane have been taken from one of the master tasks from last course.

Table 7. Fluid and membrane properties

	Symbol	Value	Units
Density	Dens	1000	kg/m^3
Kinematic viscosity	nu	10^{-6}	m^2/s
Diffusivity	DAB	10^{-9}	m^2/s
Hydraulic diameter	d	$[2 \cdot 10^{-4} - 4 \cdot 10^{-4}]$	m
Membrane permeability	Lp	$2 \cdot 10^{-11}$	m^2s/kg
Real rejection	Rreal	0.8	-
Osmotic coefficient	osmCoef	$7.093 \cdot 10^6$	kg/ms^2

2.1.5. Flux regime modeling

In the `turbulenceProperties` directory, the `simulationType` is set to `laminar` as all the simulations have been developed in the laminar region, so no turbulence models are used.

2.1.6. Time step and data output control

In the `controlDict` file the time adjustments can be made, which can be show in Appendix VI. It is not useful to have a fixed time step because the propagation of the velocity is not easily predicted, so `adjustTimeStep` is set to `yes`. The time step is modified in order to maintain the Courant number under a specific value. `maxCo` is the keyword for maximum Courant number and is set to `1`. The Courant number is defined as

$$\mathrm{Co} = \frac{\Delta t |u|}{\Delta x} \qquad (45)$$

Where u is the velocity in the x -direction, Δt is the size of the time step and Δx is the grid size. When a time step becomes too large the solution will diverge and the simulation has to stop. To avoid this from happening in the case of this study, the time step must obey the Courant-Friedrichs-Lewy condition which, for an 1D case, if defined as Co < 1 (Aslak, 2014).

- `maxDeltaT`. Determines the upper limit to the time step, and it is set to `2`, although never is reached.
- `writeInterval` sets the time when the results are written, even if in OpenFOAM the calculation are performed at arbitrary time steps, and it depends of the case but generally is set to `0.05`.
- `writeControl` is set to `adjustableRunTime` to allow this.
- `startFrom` keyword is set to `startTime`
- `startTime` is set to `0`, so the first fields data input is read from the `0`/directory.
- `endTime` is a variable that has to be adjusted from case to case, because depending on the result needed more time will be needed. In these cases it varies from 10 to 30 seconds.
- `stopAt` is set to `endTime`.
- `writeFormat` is set to the default value `ascii`, but it could also be written in the binary format by typing `binary`.

- writePrecision is set to 6, adopting the value from the chanel365 tutorial
- writeCompression is set to off, because the data files are not too big, but if the reduction is needed it can be set to compressed and the space occupied by the cases will be smaller.
- timeFormat is set to general
- timePrecision is set to 6 which are default values for OpenFOAM.

2.1.7. Numerical schemes

The fvSchemes dictionary in the system directory sets the numerical schemes for terms, such as derivatives in equations, that are calculated during a simulation. This dictionary is subdivided into the categories listed:

- ddtSchemes

According to OpenFOAM, this keyword represents the choice of time scheme. In this case Euler is adopted and indicates a first order bounded implicit scheme, which is sufficiently accurate due to the small time steps created by the Courant number restriction.

- gradSchemes

This keyword defines the discretization schemes for the gradient terms, and in this case is default Gauss linear. Default means that the Gauss linear scheme will be applied to all the gradient terms. The Gauss entry specifies the standard finite volume discretization of Gaussian integration which requires the interpolation of values from cell centers to face centers (Greenshields, 2016). The interpolation scheme is then given by the linear entry, meaning linear interpolation or central differencing. Also the scheme Gauss cubic was used, which is a third-order scheme, but the simulation results were the same that with Gauss linear.

- divSchemes

It defines the convection schemes in the momentum equation, ∇UU, defined as div(phi,U) in OpenFOAM. Here the scheme Gauss linearUpwind grad(U) is used. The upwind differencing is the most stable interpolation method available in OpenFOAM. The convection scheme for the term of the species conservation equation is Gauss limitedLinear01 1, which means that linear scheme limits towards upwind in regions of rapidly changing gradient, and the coefficient 1 is strongest limiting, tending towards linear if the coefficient were 0. The term 01 after limitedLinear specializes the scheme for stronger bounding between 0 and 1. Following, the numerical scheme of div((nuEff*dev2(T(grad(U))))), is set to Gauss linear because only works with this option (Greenshields, 2016).

- laplacianSchemes

This keyword defines the laplacian scheme, and it is applied to the terms with the laplacian operator ∇^2. The Gauss scheme is the only choice of discretization and requires a selection of both an interpolation scheme for the diffusion coefficient and a surface normal gradient scheme. In this case it is set to default Gauss linear limited 1. The linear interpolation is used for interpolation of the diffusivity. It is used the same array of snGradSchemes based on level on non-orthogonallity – limited 1 means that to maintain second-order accuracy an explicit non-orthogonal correction can be added to the orthogonal component (Greenshields, 2016).

- `interpolationSchemes`

For the interpolation schemes OpenFOAM offers centered, upwind convection, TVD (total variation diminishing) and NVD (normalized variable diagram) (Greenshields, 2016). Here a centered interpolation scheme is used under `interpolationSchemes`, `default linear`, which is the one most generally used.

- `snGradSchemes`

This keyword indicates the surface normal gradient schemes, and evaluates the gradient normal to the face center shared by two cells. It is set as `default limited 1`.

The `fvSchemes` file can be seen in Appendix VII.

2.1.8. Solution and algorithm control

The equation solvers, tolerances and algorithm are controlled form the `fvSolution` dictionary, in `system` directory. The settings specify how to solve the equations based on matrix inversions. Often the equations to be solved in OpenFOAM result in large matrices. These matrices are however mostly built by zero entries. Therefore, the traditional algebraic techniques become inefficient and iterative methods are adopted instead.

`fvSolution` contains a set of subdictionaries that are specific to the solver being run. These subdictionaries are:

- `Solvers`

A solver is assign to each variable. In the case of pressure, GAMG is selected, Generalized Geometric-Algebraic Multi-Grid. The solver PBiCG (Preconditioned Bi-Conjugate Gradient), for asymmetric matrices, is used for velocity and solute mass fraction. The solvers distinguish between symmetric matrices and asymmetric matrices. If the user specifies a symmetric solver for an asymmetric matrix, or vice versa, an error message will be reported. For the preconditioner keyword, DILU is selected for velocity and concentration equations, which means Diagonal Incomplete-LU, for asymmetric matrices (Greenshields, 2016).

Before solving an equation for a particular field, the initial residual is evaluated based on the current values of the field. After each solver iteration the residual is re-evaluated. The solver stops if any one of the following conditions is reached (Greenshields, 2016):

a) The residual falls below the solver tolerance, `tolerance`
b) The ratio of current to initial residuals falls below the solver relative tolerance, `relTol`
c) The number of iterations exceeds a maximum number of iterations, `maxIter`

Equations are very often solved multiple times within one solution step, or time step. For example, a pressure equation is solved according to the number specified by `nCorrectors`. Where this occurs, the solver is very often set up to use different settings when solving the particular equation for the final time, specified by a keyword that adds `Final` to the field name. If it specified to solve pressure for example 4 times within one time step, then the first 3 solutions would use the settings for `p` with the correspond tolerance in a way that the cost of solving each equation is relatively low. Only when the equation is solved the final (4^{th}) time, it solves to a residual level specified by tolerance (since `relTol` is now 0, effectively deactivating it) for greater accuracy, but at greater cost (Greenshields, 2016).

- PIMPLE

This is the second subdictionary of `fvSolution`, which is the main algorithm. The PIMPLE algorithm is an iterative procedure for coupling equations for momentum and mass conservation for transient problems. The algorithm solves a pressure equation to enforce mass conservation, with an explicit correction to velocity to satisfy momentum conservation. The looping is controlled by input parameters that listed below (Greenshields, 2016):

a) `nCorrectors`. Sets the number of times the algorithm solves the pressure equation and momentum corrector in each step. Value: `3`
b) `nNonOrthogonalCorrectors`. Specified repeated solutions of the pressure equation, used to update the explicit non-orthogonal correction of the Laplacian term. Value: `0`
c) `nOuterCorrectors`. It enables looping over the entire system of equation within on time step, representing the total number of times the system is solved. Value: `4`
d) `pRefCell` and `pRefValue`. These entries are used by the solver when case demands it. In a closed incompressible system. Pressure is relative: it is the pressure range that matters not the absolute values. The values depend if there is permeate flux or not.

- relaxationFactors

This is the last subdictionary of `fvSolution`. It controls under-relaxation, a technique used for improving stability of a computation. Under-relaxation works by limiting the amount which a variable changes from one iteration to the next, either by modifying the solution matrix and source prior to solving for a field or by modifying the field directly (Greenshields, 2016). An under-relaxation factor specifies the amount of under-relaxation. The values in this work were `p` (`0.5`), `U` (`0.8`) and `wA` (`0.8`).

The `fvSoltuion`-directory can be viewed in Appendix VIII.

2.2. Solving

2.2.1. Previous steps

Prior to the running of the solver, some other steps have to be done. First one, erasing the previous data on the case. Due to the iterative character of the simulations it is really important to erase the data previously obtained because if not, the results could be just partially overwritten to the previous simulation and maybe not correct. For this purpose, the command `foamCleanTutorials` is used. This way, every simulation runs without previous data.

The meshes of Geometry 1, Geometry 2 and Geometry 3 are generated by running `blockMesh`. Then the command `checkMesh` confirms if the mesh is appropriate. In the case of Geometry 4 and Geometry 5, the mesh is generated with the software cfMesh, so it is necessary to run the command `cartesian2DMesh`. Then, the file boundaries in `polyMesh` directory is modified to establish the type of patch (`wall`, `empty` and `patch`), because with cfMesh all the boundaries are empty by default. Finally, the dictionary `createPatchDict` is executed by running `createPatch -overwrite`, which converts the inlet and outlet boundaries in cyclic boundaries, instead of patch. It is necessary to use the tool `createPatch` to generate cyclic boundaries in a mesh made with cfMesh, otherwise several errors will appear about matching faces, and even without these errors, there will be numerical errors in the simulations. The file `createPatchDict` can be seen in Appendix IX. After the generation of the mesh by this way, the command `checkMesh` is executed to confirm the mesh is adequate.

2.2.2. Solvers

OpenFOAM's C++ code is flexible with respect to solver implementation, allowing users to generate solvers by following existing solvers' file structuring as templates. The solvers developed via template method are `MypimpleFoamwA` and `MypimpleFoamUW`. It was also create the solver `MypimpleFoamU`, which is basically equal to the OpenFOAM template `pimpleFoam`. In `MypimpleFoamwA` and `MypimpleFoamUW`, the solute mass fraction equations and the determination of the mass transfer coefficients have been implemented, as well as all the other parameters needed. Some of the simulations were developed with a constant concentration at the wall (impermeable dissolving wall). In these cases, the velocity and pressure can be solved separately from the solute mass fraction, so first velocity and pressure are solved and when a steady state is reached, solute mass fraction is solved. In this way, the solver `MypimpleFoamU` solves `U` and `p`, and `MypimpleFoamwA` solves `wA`. The cases with semi-permeable walls would be simulated with the solver `MypimpleFoamUW`, which solves all the variables `U`, `p` and `wA` 'simultaneously'. This is due to in these cases the solute mass fraction at the wall depends on the velocity at the wall in the y-direction.

The template solver `pimpleFoam`, included in OpenFOAM, is a large time-step transient solver for incompressible flow using the PIMPLE (merged PISO-SIMPLE) algorithm. PISO is an acronym for Pressure Implicit Splitting of Operators for time dependent flows while SIMPLE stands for Semi-Implicit Method for Pressure Linked Equations which is used for steady state problems. The PIMPLE algorithm has two loops, inner and outer. In the outer loop all equations are solved while in the inner loop only the continuity one. It is important to have the continuity error under control. It is important also for the Courant number not to increase too much (Greenshields, 2016)

As each iteration is equivalent to a pseudo time step, the properties are under-relaxed in order to stabilize the method and improve convergence. Thus, the PIMPLE algorithm is more robust than the PISO algorithm since it is able to apply the PISO algorithm to both steady state and transient situations (Koch, 2015).

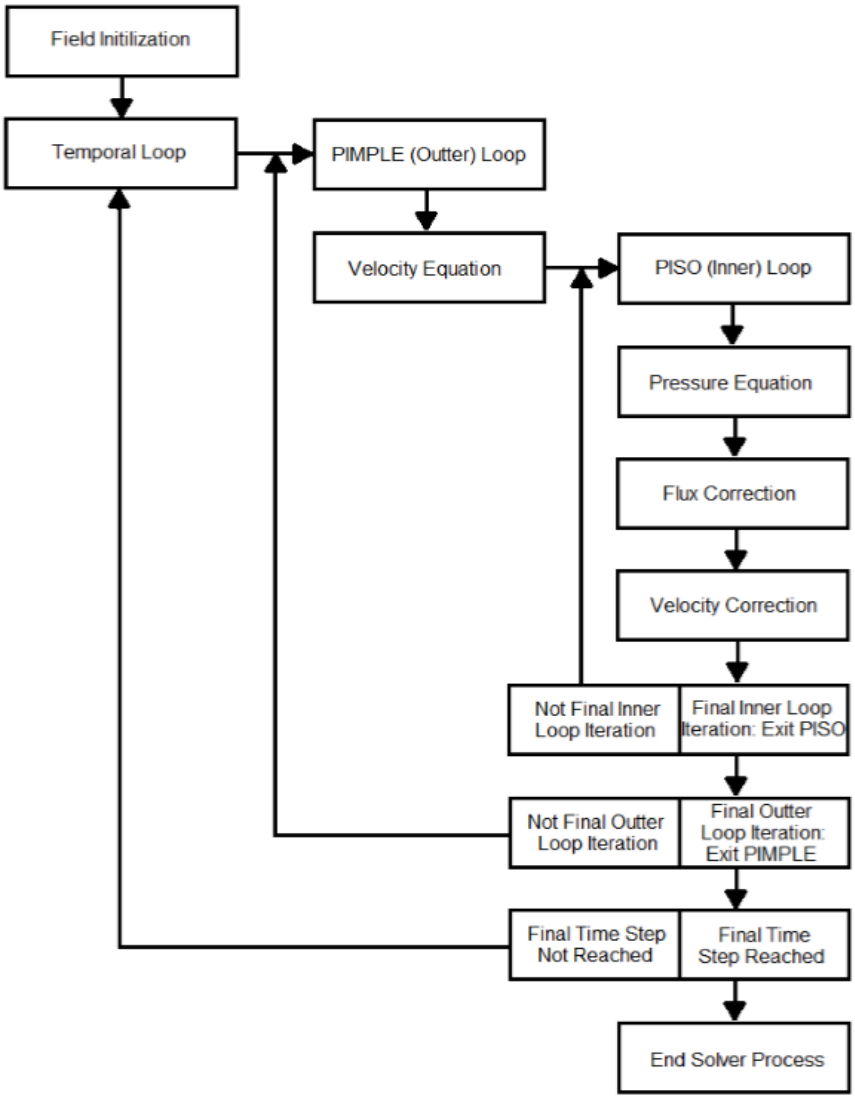


Figure 19. PIMPLE algorithm (Koch, 2015)

The solvers consist in several files which interact with each other and various lower-level OpenFOAM functionalities. The main files are `MypimpleFoamU`.C, `MypimpleFoamwA`.C and `MypimpleFoamUW`.C, which contains the basic iterative procedure structure, and calls other header files for certain calculations. These files begin with a list of included header files which are either lower than the top-level code of the solver, or are files which are specific to and found only within the file system of the specific solver.

When a simple case of mass transfer is simulated, without membrane conditions, three essential files are called within the solver code to enable the solving of the pressure, velocity and solute mass fraction within the PIMPLE iterative structure: the `UEqn`.H, `pEqn`.H and `wAEqn`.H user-defined files. When a membrane is simulated (the solute mass fraction is not constant at the walls) the called files are `UEqn`.H, `pEqn`.H, `wAEqn`.H and also `membraneBC`.H, which includes the necessary equations to determine the solute mass fraction variation, permeate flux, retentate flux, solute mass fractions and other typical membrane results.

The `createFields`.H file included within the initial header calls manages the loading and instantiation of the field variables for pressure, solute mass fraction, velocity, and case-defined constants required by equations within later header file calls.

Although it is possible to run in parallel, this is not done in any of the cases because the number of ells is not that big, and the size of the geometries is relatively small.

2.3. Post-processing

Once the OpenFOAM solver is complete, the field values are stored separately in text files. Inside each file, the value for each cell of the given field is listed. Analyzing these values as they are presented, without any positional data, does not offer much insight. However, OpenFOAM comes with its own post-processing software ParaView, which is launched by writing `paraFoam -builtin` in the terminal.

ParaView combines the field values and positional data with the mesh model to create a visual representation of the OpenFOAM simulation. All the fields included in the simulation can be inspected and colored according to magnitude. In addition, there are several tools to closer inspect the results, such as graphing tools, hiding parts of the mesh and so on.

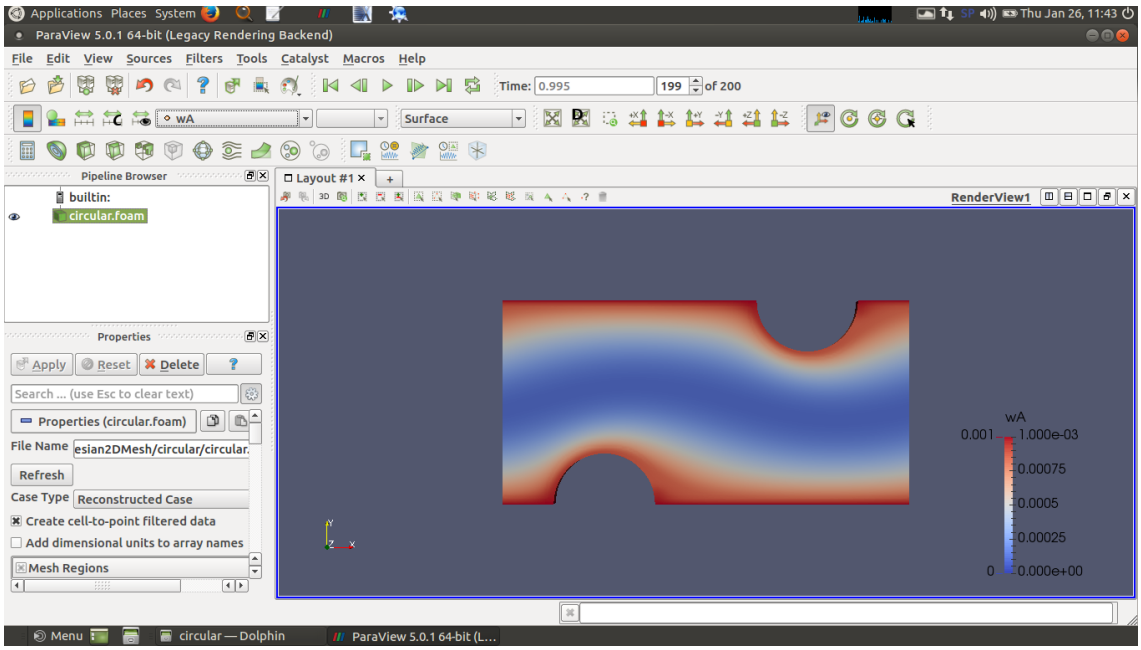


Figure 20. Paraview interface

Results and discussion

1. Mesh independency

The main distinction between the geometries is the presence or absence of spacers. It has been studied the mesh independency of Geometry 1, without spacers, and also of Geometry 5, with spacers. The results can be applied to the other geometries used in this work, Geometry 2 and Geometry 4.

1.1. Geometry 1

It was analyzed the mesh independency in the determination of pressure and velocity fields. Since Geometry 1 is a simple geometry, most of the studied meshes provide a good result, as it can be seen in the following table.

Table 8. Different meshes for Geometry 1

	Number of cells	Refinement	Relative error (%)
Mesh 1	30 x 30 x 1	0.3	0.500
Mesh 2	100 x 100 x 1	0.3	0.020
Mesh 3	170 x 170 x 1	0.3	0.017

The relative error was calculated comparing the velocity profile with the analytical velocity profile for a circular duct.

Mesh 2 was selected for the simulations. It can be considered good enough and the results are practically the same if the number of cells is increased.

As has already been said, the refinement in the vicinity of the wall is a critical parameter in the mass transfer study. Three degrees of refinement were analyzed according to their effect in the mass transfer. It was compared the Sherwood number along the duct. According to OpenFOAM syntax, a refinement of 1 is equal to a null refinement, while a refinement close to 0 is equivalent to an 'infinite' refinement. The results are in the following figure:

Table 9. Different refinements for Geometry 1

	Number of cells	Refinement
Mesh A	100 x 100 x 1	0.9
Mesh B	100 x 100 x 1	0.3
Mesh C	100 x 100 x 1	0.2

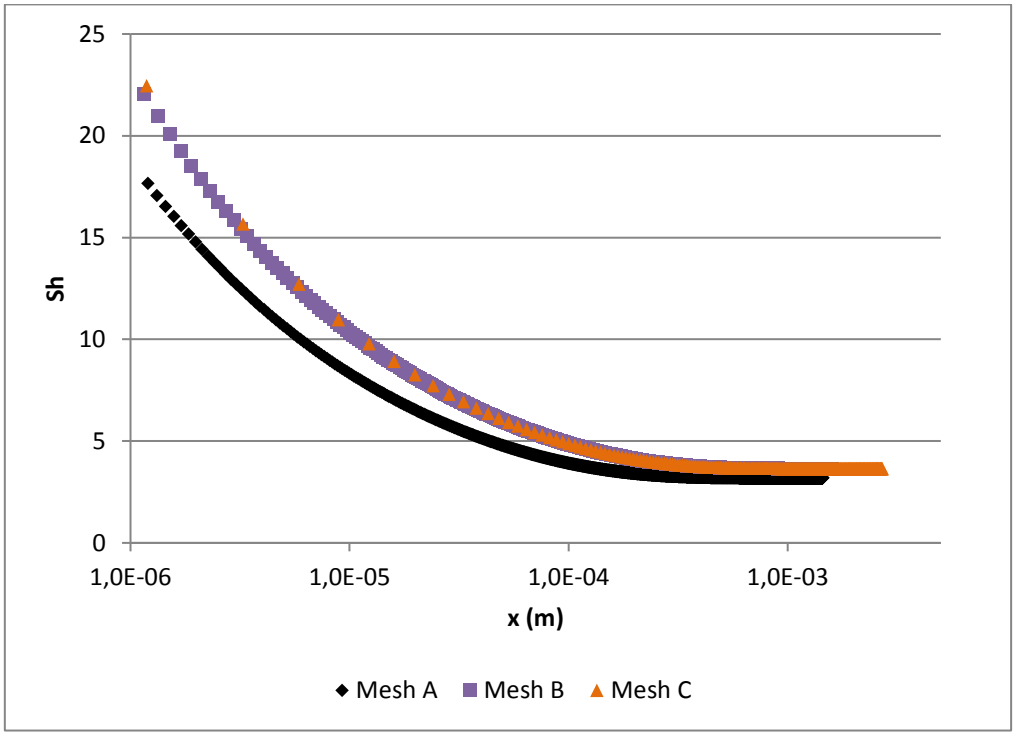


Figure 21. Sherwood number vs x-direction with three different meshes in Geometry 1

Mesh B was selected for the simulations. The results are the same that with a higher refinement, so it is a suitable mesh.

1.2. Geometry 5

In the case of geometries with spacers, it is even more important an adequate mesh in order to obtain a correct pressures and velocity fields. The results can be seen in the following figures:

Table 10. Different meshes for Geometry 5

	Max. cell size (m)	Refinement thickness (m)
Mesh I	$1.84 \cdot 10^{-5}$	$5 \cdot 10^{-6}$
Mesh II	$1.84 \cdot 10^{-6}$	$5 \cdot 10^{-6}$
Mesh III	$1.70 \cdot 10^{-6}$	$5 \cdot 10^{-6}$

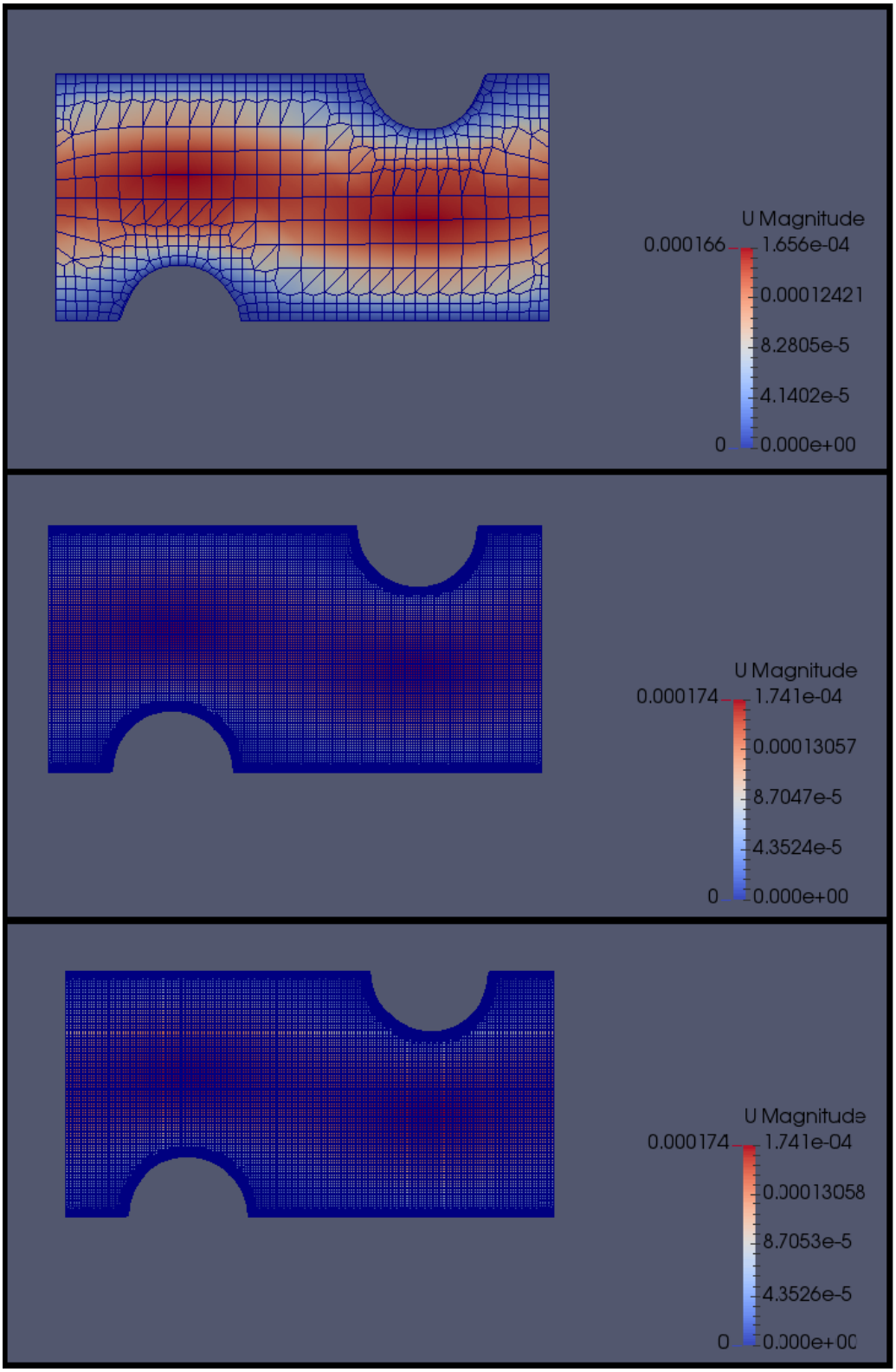


Figure 22. Velocity field with three different meshes in Geometry 5

The chart legend shows that Mesh II is good enough for the simulations, because the results does not change if the number of cells is higher.

It was also analyzed the effect of the refinement in the determination of Sherwood number along the duct:

Table 11. Different refinements for Geometry 5

	Refinement thickness (m)	Boundary cell size (m)
Mesh a	0	-
Mesh b	$5 \cdot 10^{-6}$	$5 \cdot 10^{-7}$
Mesh c	$6 \cdot 10^{-6}$	$4.5 \cdot 10^{-7}$

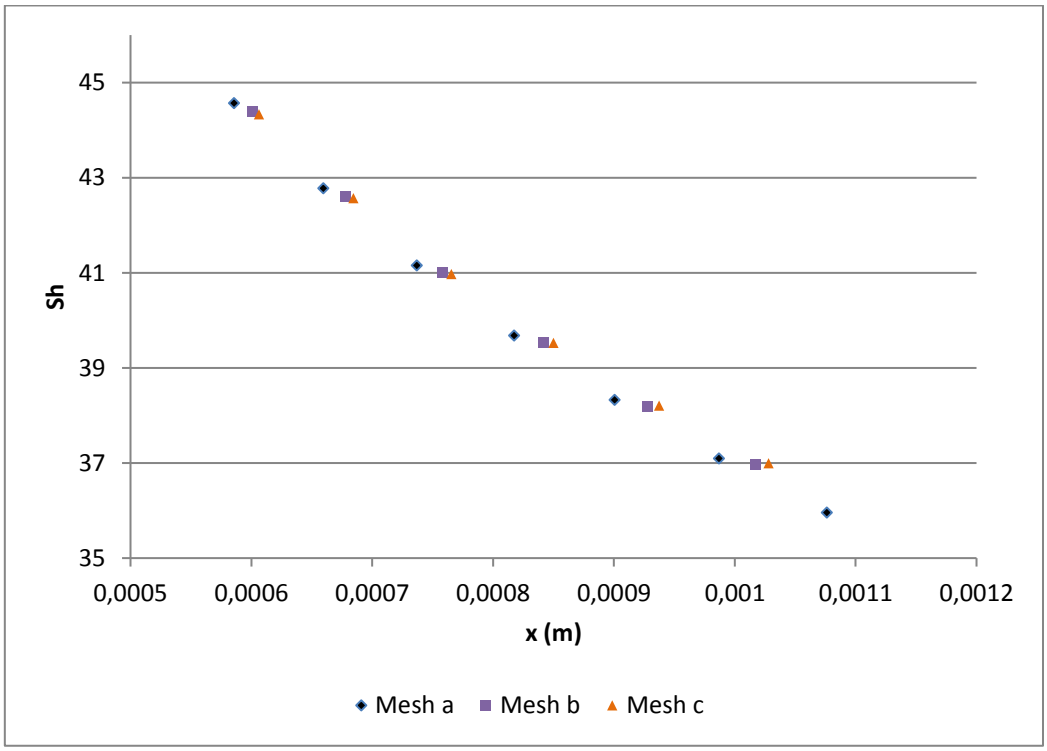


Figure 23. Sherwood number vs x-direction with three different meshes in Geometry 5

As Figure 23 shows, the meshes with refinement give practically the same results, while Mesh A, without refinement, gives a little different result. Generally, the results are good because the cell size is suitable. Mesh b was selected for the simulations.

2. Validation

With the validation it is verified that the resolution of the equations is suitable, and also that the application of periodic boundary conditions is adequate. Geometries without obstacles were validated, as the results for them are presented in literature, both analytic and numeric.

2.1. Geometry 1 and long Geometry 1

In the case of tubular configuration, one periodic domain and one non periodic domain were simulated. The non periodic domain was ten times larger than the periodic one, both had the same dimensions except length. An average velocity was selected and also a constant solute mass fraction at the membrane wall, which was simulated as impermeable.

Case sizes

Table 12. Size of Geometry 1 for the validation

Periodic domain (Geom. 1)	
H (m)	$2 \cdot 10^{-4}$
L (m)	$4 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
$d_h(m)$	$4 \cdot 10^{-4}$

Table 13.Size of long Geometry 1 for the validation

Real duct (Long Geom. 1)	
H (m)	$2 \cdot 10^{-4}$
L (m)	$4 \cdot 10^{-3}$
Z (m)	$1 \cdot 10^{-5}$
$d_h(m)$	$4 \cdot 10^{-4}$

The selected meshes are in Methodology chapter.

Initial and boundary condition:

Table 14. Initial conditions of Geometry 1 for the validation

Initial conditions. Periodic domain. (Geom. 1)	
U	$1 \cdot 10^{-4}$
V	0
w_A	0
P	0

Table 15. Boundary conditions of Geometry 1 for the validation

Boundary conditions. Periodic domain. (Geom. 1)				
	u	v	w_A	p
topWall	0	0	0.001	$\partial p/\partial y = 0$
bottomWall	empty	empty	empty	empty
Inlet	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic
bottomEmptyFaces	wedge	wedge	wedge	wedge
topEmptyFaces	wedge	wedge	wedge	wedge

Table 16. Initial conditions of long Geometry 1 for the validation

Initial conditions. Real duct (Long Geom. 1)	
u	0
v	0
w_A	0
P	0

Table 17.Boundary conditions of long Geometry 1 for the validation

Boundary conditions. Real duct. (Long Geom. 1)				
	U	v	w_A	p
topWall	0	0	0.001	$\partial p/\partial y = 0$
bottomWall	empty	empty	empty	empty
inlet	$2 \cdot 1 \cdot 10^{-4} \cdot [1 - (y/H)^2]$	0	0	$\partial p/\partial x = 0$
outlet	$\partial u/\partial x = 0$	$\partial v/\partial x = 0$	$\partial w_A/\partial x = 0$	0
bottomEmptyFaces	Wedge	wedge	wedge	wedge
topEmptyFaces	Wedge	wedge	wedge	wedge

Solver

First it was used the solver `MypimpleFoamU` to determine the velocities and pressures fields, after that the solver `MypimpleFoamwA` to determine the solute mass fractions and mass transfer parameters. The velocity and pressure final values are selected as initial conditions of the domain before run the solver `MypimpleFoamwA`. The solvers are running until the steady state is reached.

Results

The Sherwood number is draw against a variant of the Graetz number, as in the Introduction chapter.

The values obtained are compared with each other and with the literature values, according to the literature Graetz solution (Welty et al., 2008).

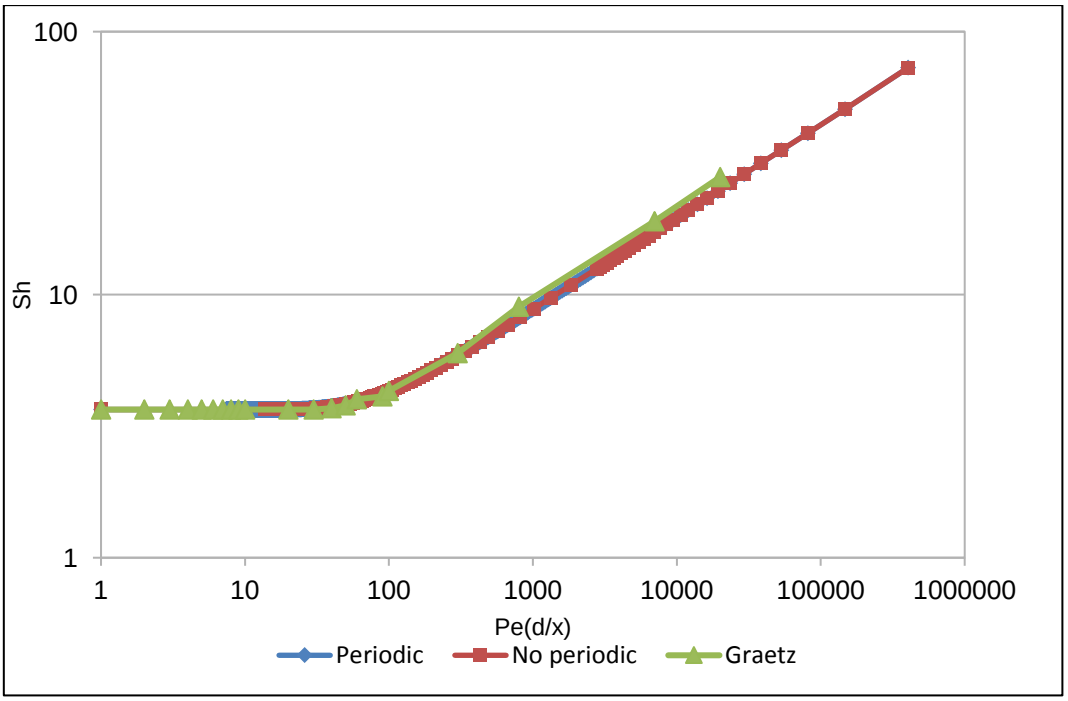


Figure 24. Sherwood number vs Pe(d/X) in Geometry 1 and long Geometry 1

The asymptotic value from literature is 3.658 which is the same value obtained from CFD. The biggest relative error between the CFD values and literature values is 12%. It is noted that possibly the CFD values are more accurate than the literature values, because the Graetz solution is an approximation, which is useful to validate purposes, but which is not the exact one.

2.2. Geometry 2 and long Geometry 2

Case sizes

Table 18. Size of Geometry 2 for the validation

Periodic domain (Geom. 2)	
H (m)	$5 \cdot 10^{-5}$
L (m)	$2 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
$d_h(m)$	$2 \cdot 10^{-4}$

Table 19. Size of long Geometry 2 for the validation

Real duct (Long Geom. 2)	
H (m)	$5 \cdot 10^{-5}$
L (m)	$2 \cdot 10^{-3}$
Z (m)	$1 \cdot 10^{-5}$
$d_h(m)$	$2 \cdot 10^{-4}$

The selected meshes are in Methodology chapter.

Initial and boundary condition:

Table 20. Initial conditions of Geometry 2 for the validation

Initial conditions. Periodic domain. (Geom. 2)	
u	$1 \cdot 10^{-4}$
v	0
w_A	0
p	0

Table 21. Boundary conditions of Geometry 2 for validation

Boundary conditions. Periodic domain. (Geom. 2)				
	u	v	w_A	p
topWall	0	0	0.001	$\partial p/\partial y = 0$
bottomWall	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
Inlet	cyclic	cyclic	cyclic	Cyclic
Outlet	cyclic	cyclic	cyclic	Cyclic
bottomEmptyFaces	empty	empty	empty	Empty
topEmptyFaces	empty	empty	empty	Empty

Table 22. Initial conditions of long Geometry 2 for the validation

Initial conditions. Real duct. (Long Geom. 2)	
u	0
v	0
w_A	0
p	0

Table 23. Boundary conditions of long Geometry 2 for validation

Boundary conditions. Real duct. (Long Geom. 2)				
	u	v	w_A	p
topWall	0	0	0.001	$\partial p/\partial y = 0$
bottomWall	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
inlet	$(3/2) \cdot 1 \cdot 10^{-4} \cdot [1 - (2y/H)^2]$	0	0	$\partial p/\partial x = 0$
outlet	$\partial u/\partial x = 0$	$\partial v/\partial x = 0$	$\partial w_A/\partial x = 0$	0
bottomEmptyFaces	empty	empty	empty	empty
topEmptyFaces	Empty	empty	empty	empty

Solver

First it was used the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

Results

In this case the correlation from the literature (Eldridge, 2014) was:

$$\mathrm{Sh}(x) = 3.12(x/d_h)^{-1/3}$$

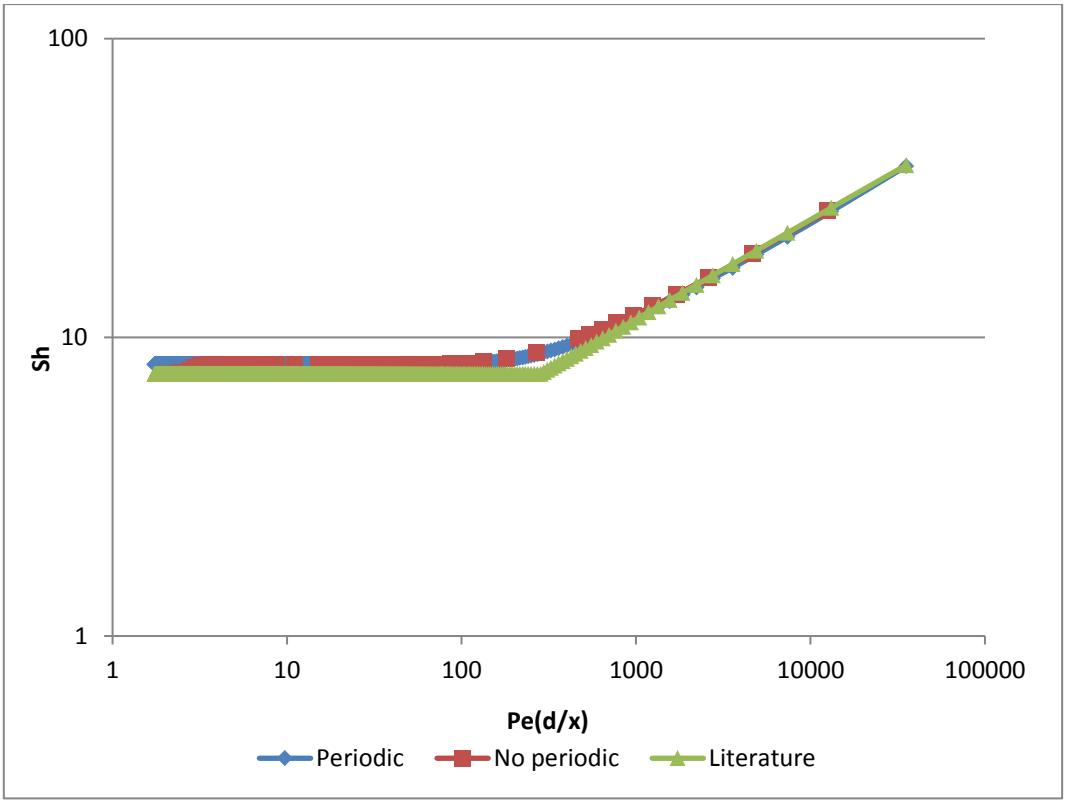


Figure 25. Sherwood number vs Pe(d/X) in Geometry 2 and long Geometry 2

The asymptotic Sherwood number from literature is 7.55, and the CFD values were 7.58 for non periodic simulation and 8.11 for periodic simulation. The literature correlation was obtained analytically with the same hypothesis that were used in Graetz solution, but in this case applied to parallel plates. The maximum relative deviation between the literature results and CFD results is a 16%, but again it can be due to CFD resolution does not simplify any term from the conservative laws.

2.3. Geometry 3 and long Geometry 3

Case sizes

Table 24. Size of Geometry 3 for the validation

Periodic domain (Geom. 3)	
H (m)	$1 \cdot 10^{-4}$
L (m)	$5 \cdot 10^{-5}$
Z (m)	$1 \cdot 10^{-4}$
$d_h(m)$	$2 \cdot 10^{-4}$

Table 25. Size of long Geometry 3 for the validation

Real duct (Long Geom. 3)	
H (m)	$1 \cdot 10^{-4}$
L (m)	$2 \cdot 10^{-3}$
Z (m)	$1 \cdot 10^{-4}$
$d_h(m)$	$2 \cdot 10^{-4}$

The selected meshes are in Methodology chapter.

Initial and boundary condition:

Table 26. Initial conditions of Geometry 3 for the validation

Initial conditions. Periodic domain. (Geom. 3)	
u	$1 \cdot 10^{-4}$
v	0
w_A	0
p	0

Table 27. Boundary conditions of Geometry 3 for the validation

Boundary conditions. Periodic domain. (Geom. 3)				
	u	v	w_A	P
topWall	0	0	0.001	$\partial p/\partial y = 0$
bottomWall	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
topEmptyFaces	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
bottomEmptyFaces	0	0	0.001	$\partial p/\partial y = 0$
Inlet	cyclic	cyclic	cyclic	Cyclic
Outlet	cyclic	cyclic	cyclic	Cyclic

Table 28. Initial conditions of long Geometry 3 for the validation

Initial conditions. Real duct. (Long Geom. 3)	
u	0
v	0
w_A	0
p	0

Table 29. Boundary conditions of long Geometry 3 for the validation

Boundary conditions. Real duct. (Long Geom. 3)				
	u	v	w_A	P
topWall	0	0	0.001	$\partial p/\partial y = 0$
bottomWall	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
topEmptyFaces	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
bottomEmptyFaces	0	0	0.001	$\partial p/\partial y = 0$
inlet	$1 \cdot 10^{-4}$	0	0	$\partial p/\partial x = 0$
outlet	$\partial u/\partial x = 0$	$\partial v/\partial x = 0$	$\partial w_A/\partial x = 0$	0

Unlike in the other geometries, in this case the velocity profile is not known so it cannot be selected as boundary conditions at the inlet, so a flat profile is selected. This profile will develop with the time until the steady state is reached.

Solver

First it was used the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

Results

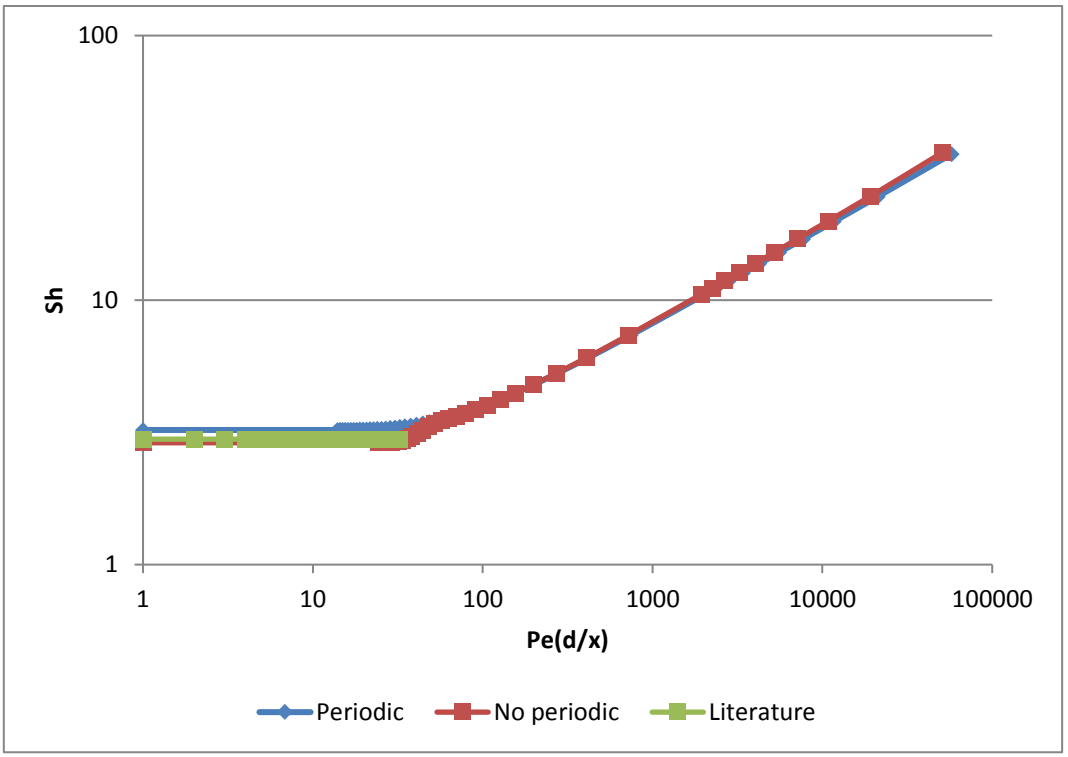


Figure 26. Sherwood number vs Pe(d/X) in Geometry 3 and long Geometry 3

There is not an analytical solution for this section, and the numerical solutions from literature show just the final asymptotic value of the Sherwood number, which is 2.98 (Bergman & Incropera, 2011). It is not a common geometry in heat/mass transfer, so it is not very studied. The maximum relative error between literature and CFD results is 3%.

2.4. Periodicity analysis

Periodicity in mass transfer is essential to apply periodic conditions. Periodicity was studied in Geometry 1 comparing data from the simulation of a real membrane duct with data from the simulation of a periodic domain of the membrane duct.

Mass transfer periodicity can be analyzed showing the relation between the solute mass fraction at the membrane wall and the solute mass fraction at the bulk along the duct length, as in the literature (Carlos Completo, Semiao, & Geraldes, 2016). Although these solute mass fractions change from one position to another position along the duct, the relation between them will be constant from a certain distance, which means that mass transfer periodicity has been reached in the duct. Results from the periodic domain can be compared with real results once the periodicity is reached.

It was simulated a periodic domain with the following characteristics:

Case sizes

Table 30. Size of the Geometry 1 for the periodicity analysis

Periodic domain (Geom. 1)	
H (m)	$2 \cdot 10^{-4}$
L (m)	$4 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
$\mathbf{d_h(m)}$	$4 \cdot 10^{-4}$

Initial and boundary condition:

Table 31. Initial conditions of Geometry 1 for the periodicity analysis

Initial conditions. Periodic domain. (Geom. 1)	
u	0.1
v	0.1
w_A	0
p	0

Table 32. Boundary conditions of Geometry 1 for the periodicity analysis

Boundary conditions. Periodic domain. (Geom. 1)				
	u	v	w_A	p
topWall	0	0	1	$\partial p/\partial y = 0$
bottomWall	empty	empty	empty	empty
Inlet	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic
bottomEmptyFaces	wedge	wedge	wedge	wedge
topEmptyFaces	wedge	wedge	wedge	wedge

Solver

First it was used the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

Results

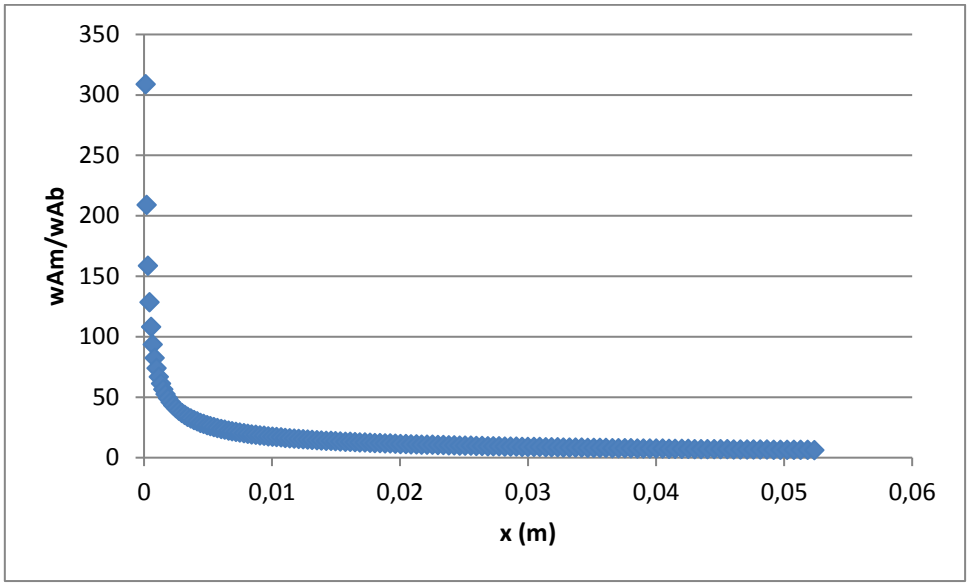


Figure 27. wAm/wAb vs x-direction in Geometry 1

According to the results, it can be considered that mass transfer periodicity is reached after 0.025 m under these conditions.

3. Estimation of concentration polarization

As it has been explained in the Introduction chapter, with the correction factor it is possible to determine the concentration polarization from the simulation of an impermeable duct with a constant solute mass fraction at the membrane wall. Mass transfer coefficients along the length of the impermeable duct will be used, and also it is necessary to known or calculate the permeate flux of the real duct. The use of the mass transfer correction factor Ξ has numerous advantages, the main of these is that the factor can be applied to the cases with high permeate flux, while the well-known film theory, cannot be used, as it will be demonstrated.

The following simulations were carried out:

3.1. Semi-permeable duct

In order to simulate a duct with permeation it was followed the next strategy: an approximation which consists of dividing the real duct into several periodic domains with impermeable walls. From one domain to another, mass balances are calculated in order to correct the velocity and solute mass fraction to take into account the permeation. This strategy has also been used in literature, as it was explained in the Introduction chapter.

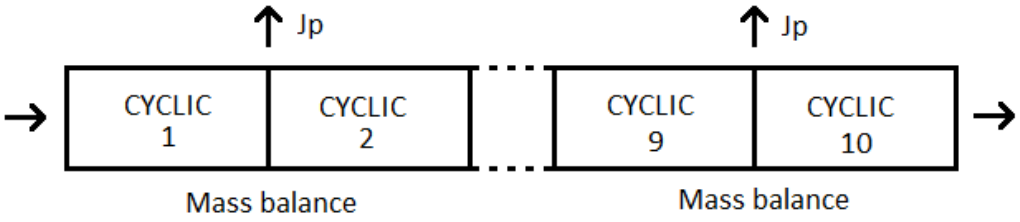


Figure 28. Real semi-permeable duct divided in impermeable periodic domains

The membrane boundary conditions explained in the Introduction chapter are used in the periodic domain, except the condition of velocity in the y direction. In order to maintain the periodicity between inlet and outlet of the domain, impermeable conditions have been selected as boundaries.

The first periodic domain is simulated until the solute has moved a distance equal to a tenth of the total real duct length. The inlet conditions of the next domain will be the outlet conditions of the previous domain, except the velocity and solute concentration because they need to be corrected. It is due to the permeation that velocity is decreasing along the duct and solute mass concentration of the bulk is increasing.

Ten periodic domains were simulated, which are equivalent to the real duct. It is worth remembering that mass transfer periodicity in this duct is not reached until an approximate distance of 0.025 m. So the first 5 periodic domains are not representative of the results, because periodicity has not been reached. The results from the sixth domain (0.024 m) to the sixteenth domain can be compared with the results from the simulation of the periodic impermeable domain with solute mass fraction constant at the membrane wall.

Case size

Table 33. Size of the Geometry 1 for the determination of mass transfer correction factor Ξ

Periodic domain (Geom. 1)	
H (m)	$2 \cdot 10^{-4}$
L (m)	$4 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
d_h(m)	$4 \cdot 10^{-4}$

The mesh characteristics are in the Methodology chapter.

Initial and boundary conditions

Table 34. Initial conditions of the Geometry 1 for the determination of mass transfer correction factor Ξ

Initial conditions. Periodic domain. (Geom. 1)	
u	0.1
v	0
w_A	$5 \cdot 10^{-5}$
p	100000

The permeate rate is calculated in each domain although it is not selected as boundary condition. The value of the permeate flux is necessary to correct the velocity and solute mass fraction from one domain to the next domain, so permeate rate is calculated as $J_p = L_p(\Delta p - \chi w_{Am} R_{real})$. The solute mass fraction boundary condition was obtained from Equation 16.

Table 35. Boundary conditions of the Geometry 1 for the determination of mass transfer correction factor Ξ

Boundary conditions. Periodic domain. (Geom. 1)				
	u	v	w_A	p
topWall	0	0	$\partial w_A/\partial y = (J_p w_{Am} R_{real})/D_{AB}$	$\partial p/\partial y = 0$
bottomWall	empty	empty	empty	empty
Inlet	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic
bottomEmptyFaces	wedge	wedge	wedge	wedge
topEmptyFaces	wedge	wedge	wedge	wedge

The values of the membrane parameters and properties of fluid are in the Methodology chapter.

Solver

The case can be simulated with the solver `MypimpleFoamUW`, or also with the solver `MypimpleFoamU` and then `MypimpleFoamwA`. It was used the solver `MypimpleFoamUW` due to previously some simulations with real permeation and without periodic boundary conditions were carried out, although the time required for these simulations was excessive for just one computer so they are not showed in this work.

The following table and figures show the velocity and the solute mass fraction in the bulk changing along the duct. Solute mass fraction is calculated in the file `membraneBC`.H, and is the solute mass fraction from the retentate flow in each domain. The velocity is calculated from the retentate mass flow, also in `membraneBC`.H file.

Results

Table 36. Velocity and solute mass fraction in the bulk along the real semi-permeable duct

	x (m)	x 'real' (m)	u (m/s)	w_{Ab}
Initial conditions	0	-	0.1	$5.000 \cdot 10^{-5}$
Domain a	0.004	-	0.0992	$5.032 \cdot 10^{-5}$
Domain b	0.008	-	0.0984	$5.065 \cdot 10^{-5}$
Domain c	0.012	-	0.0972	$5.098 \cdot 10^{-5}$
Domain d	0.016	-	0.0964	$5.132 \cdot 10^{-5}$
Domain e	0.020	-	0.0956	$5.166 \cdot 10^{-5}$
Domain 1	0.024	0.004	0.0948	$5.201 \cdot 10^{-5}$
Domain 2	0.028	0.008	0.0940	$5.236 \cdot 10^{-5}$
Domain 3	0.032	0.012	0.0932	$5.272 \cdot 10^{-5}$
Domain 4	0.036	0.016	0.0924	$5.308 \cdot 10^{-5}$
Domain 5	0.040	0.020	0.0916	$5.345 \cdot 10^{-5}$
Domain 6	0.044	0.024	0.0908	$5.382 \cdot 10^{-5}$
Domain 7	0.048	0.028	0.0900	$5.420 \cdot 10^{-5}$
Domain 8	0.052	0.032	0.0892	$5.459 \cdot 10^{-5}$
Domain 9	0.056	0.036	0.0884	$5.498 \cdot 10^{-5}$
Domain 10	0.060	0.040	0.0876	$5.538 \cdot 10^{-5}$

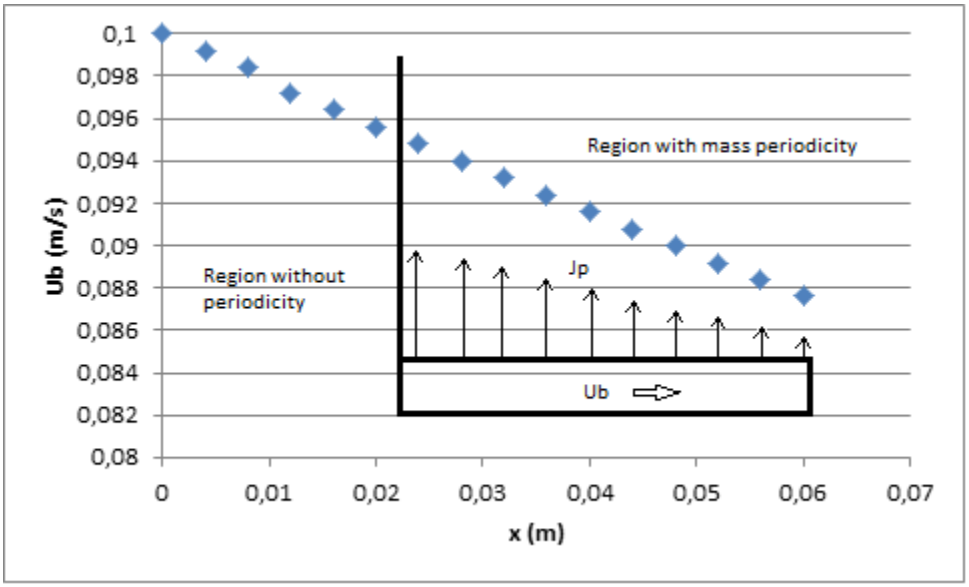


Figure 29. Velocity in the bulk vs x-direction in a semi-permeable tubular duct

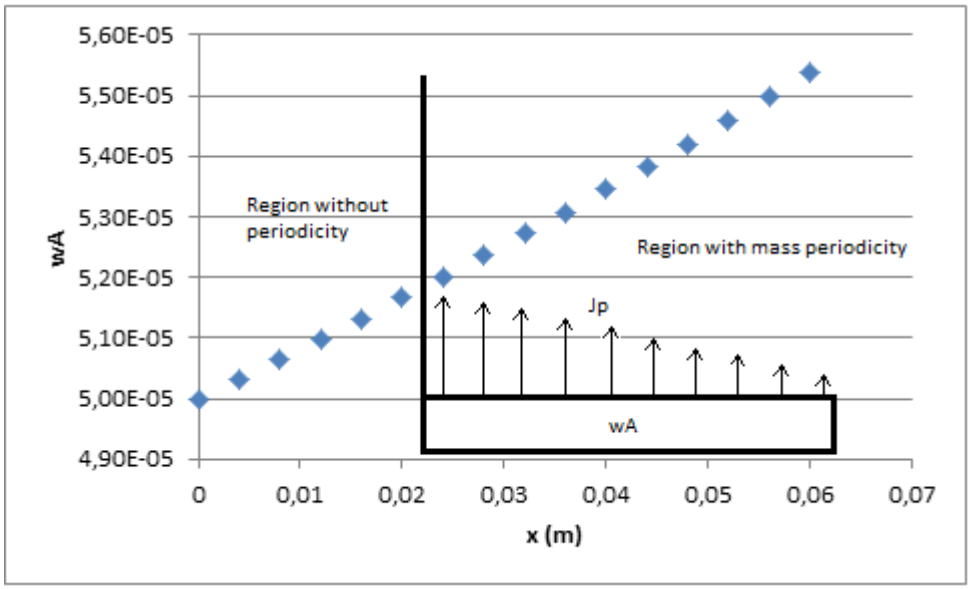


Figure 30. Solute mass fraction in the bulk vs x-direction in a semi-permeable tubular duct

Once the total length has been simulated, the solute mass fraction at the membrane wall and the permeate rates are selected. These values can be seen in Figure 31 and Figure 34 as 'CFD' values.

In order to determine the parameters for the correlation of the correction factor Ξ, the following simulation was also carried out, with impermeable dissolving walls.

3.2. Impermeable periodic domain

At the wall the solute mass fraction is an arbitrary constant value. The solute diffuses to the center of the duct along the length. The domain is simulated until the solute moves a distance equal to the distance of the real duct (0.04 m).

Case size

Table 37. Size of the Geometry 1 for the simulation of an impermeable periodic domain

Periodic domain (Geom. 1)	
H (m)	$2 \cdot 10^{-4}$
L (m)	$4 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
$d_h(m)$	$4 \cdot 10^{-4}$

Initial and boundary conditions

Table 38. Initial conditions of the Geometry 1 for the simulation of an impermeable periodic domain

Initial conditions. Periodic domain (Geom. 1)	
u	0.1
v	0
w_A	0
p	0

Table 39. Boundary conditions of the Geometry 1 for the simulation of an impermeable periodic domain

Boundary conditions. Periodic domain. (Geom. 1)				
	u	v	w_A	p
topWall	0	0	1	$\partial p/\partial y = 0$
bottomWall	empty	empty	empty	empty
Inlet	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic
bottomEmptyFaces	wedge	wedge	wedge	wedge
topEmptyFaces	wedge	wedge	wedge	wedge

Solver

The simulation was solved with the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

Results

After the simulations, ten values of the mass transfer coefficient k_c are selected.

With the permeate rate values from the previous simulation and these k_c values, the term ϕ can be calculated as $\phi = J_p/k_c$. With this term and the following equations, the mass transfer correction factor Ξ is determined:

$$\Xi = \phi + \frac{1}{[1 + c_2\phi^{c_3}]^{c_1}}$$

$$w_{Am} = \frac{w_{A0}}{1 - R_{real} \cdot (\phi/\Xi)}$$

The parameters c_1, c_2 and c_3 are calculated iteratively in order to reduce the total relative error in solute mass fraction at the membrane, comparing these results with the results from the previous simulation. It is worth remembering that the parameters must be positive to enforce the restrictions explained in the Introduction chapter. The obtained values for the selected operation conditions are:

Table 40. Parameters of the mass transfer correction factor Ξ for low permeation rates

c_1	0.02
c_2	0.01
c_3	5

Figure 31 shows the solute mass fraction at the membrane wall in the case of a real semi-permeable wall ('CFD') and in the case of an impermeable periodic domain ('Factor').

4. Comparison between correction factor Ξ and Film Theory

It was also calculated the solute mass fraction at the membrane wall applying the well-known Film Theory, in order to compare it with the correction factor Ξ method, and in this way studying the applicability limits of the Film Theory.

The calculation process is explained in the Introduction chapter.

Results

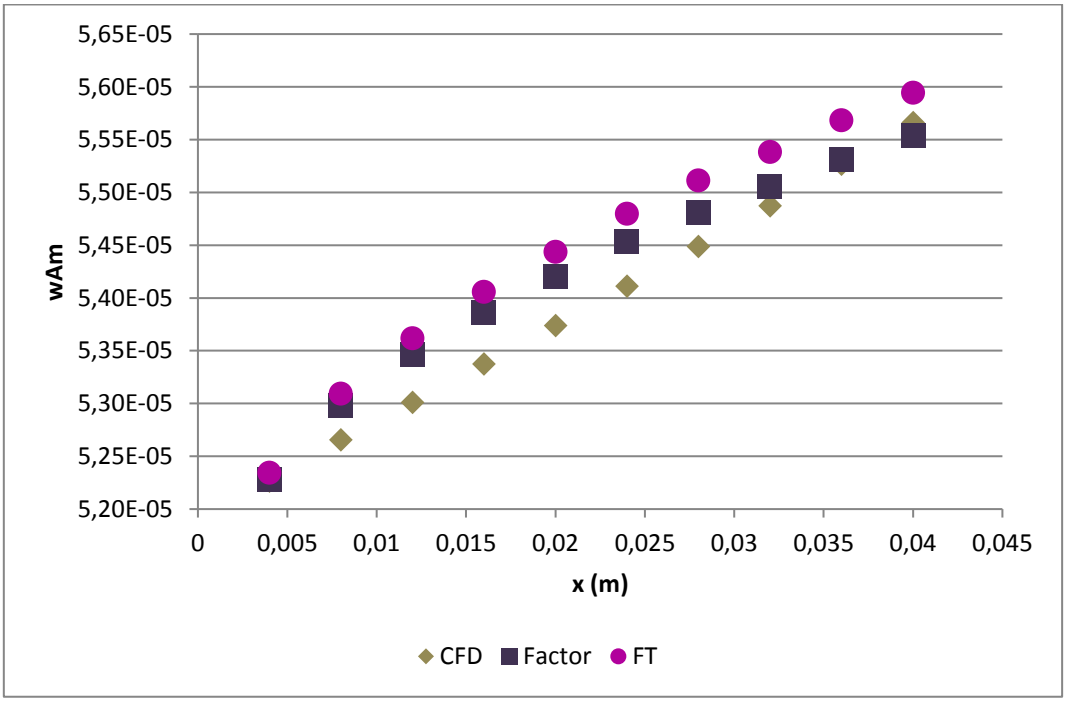


Figure 31. Solute mass fraction at the membrane wall vs x-direction in a real semi-permeable tubular duct

Table 41. Relative error of the solute mass fraction at the membrane wall with mass transfer correction factor Ξ and Film Theory

Domain	Relative error factor Ξ (%)	Relative error FT (%)
1	0.024	0.151
2	0.617	0.833
3	0.853	1.144
4	0.918	1.279
5	0.870	1.296
6	0.772	1.264
7	0.592	1.144
8	0.326	0.931
9	0.084	0.752
10	0.216	0.505

The relative error is small in both cases, but the best results are provided by the correction factor Ξ.

Film Theory can be applied to calculate solute mass fraction at the membrane wall but it is only suitable if $\phi < 0.2$, which means, relative low permeation fluxes. This can be seen in the following figure:

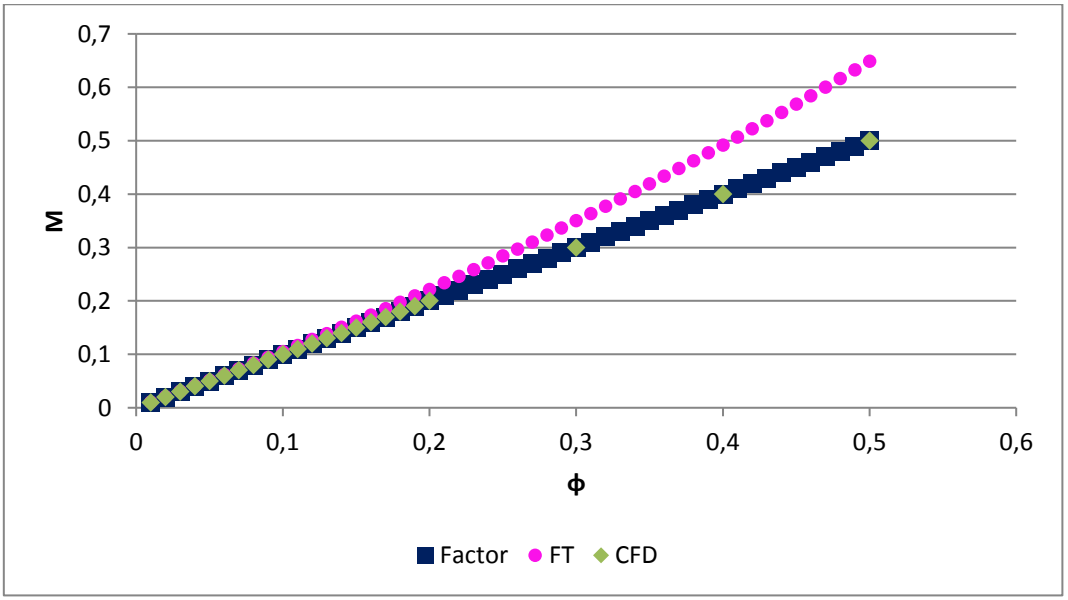


Figure 32. Concentration polarization index vs ϕ for low permeate rates with mass transfer correction factor (Factor), Film Theory (FT) and CFD

If the permeation flux is higher, Film Theory is also not adequate, as it can be seen in the following figure.

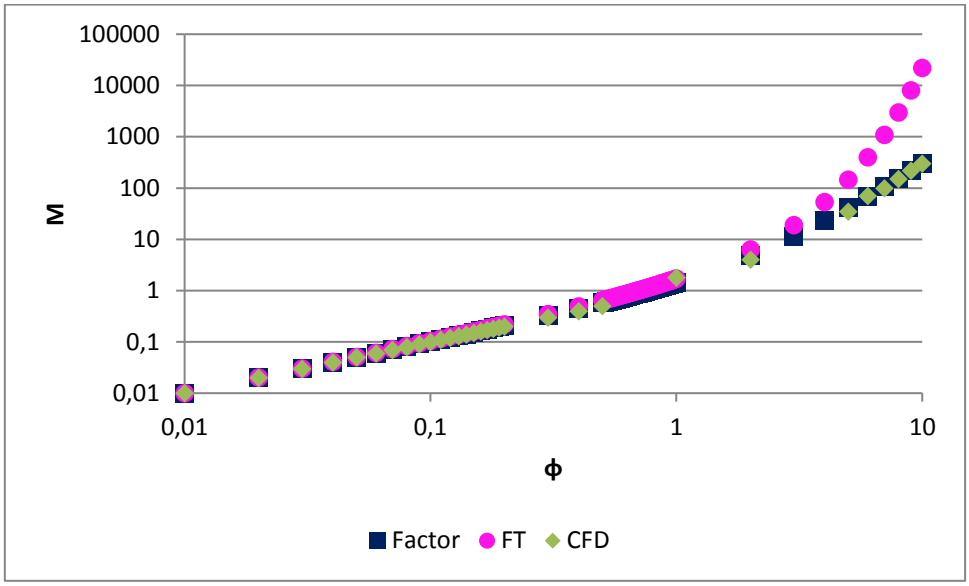


Figure 33. Concentration polarization index vs ϕ with mass transfer correction factor (Factor), Film Theory (FT) and CFD

CFD values were obtained from literature (Vitor Geraldes & Afonso, 2006) due to high permeate rates implies high computational load. The values of the parameters c_1, c_2 and c_3 have been modified to be suitable for all the range of permeate rate, and also taken from literature (Vitor Geraldes & Afonso, 2006). The new values, for bigger operation range, are:

Table 42. Parameters of the mass transfer correction factor Ξ

c_1	1.7
c_2	0.26
c_3	1,4

As it was explained in the Introduction chapter, the calculation of permeate rate from an impermeable simulation needs an iterative process, until the rate calculated with the hydraulic equation was equal to the permeate rate according to the method used. If the method used is the Film Theory, the iterative process can be seen in Equation 34. If the method is the correction factor Ξ, it can be seen in Equation 44.

The next Figure compare the values of the permeate rate from the correction factor Ξ, from the Film Theory and from CFD.

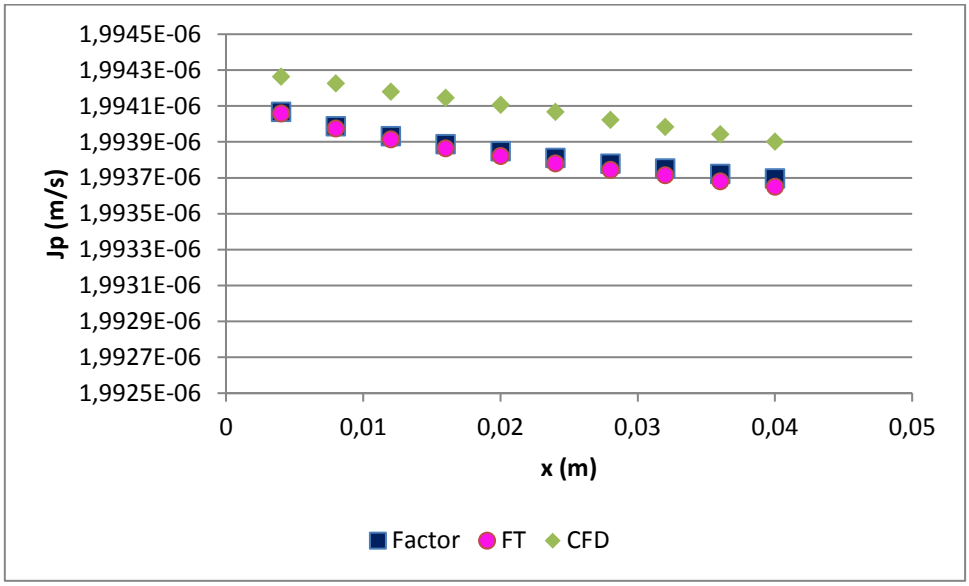


Figure 34. Permeate rate J_p vs x-direction in a real semi-permeable tubular duct

Table 43. Relative error in the calculation of permeate rates.

Domain	Relative error J_p Film Theory (%)	Relative error Factor (%)
1	0.010	0.010
2	0.013	0.012
3	0.013	0.013
4	0.014	0.013
5	0.014	0.013
6	0.014	0.013
7	0.014	0.013
8	0.014	0.012
9	0.013	0.012
10	0.013	0.011

According to the Table 43, both methods give good results, so in the following the Film Theory will be used to determine the permeate rate, and the correction factor Ξ will be used to calculate concentration polarization.

5. Effect of geometry of the duct in concentration polarization

It was studied the effect of the geometry analyzing the concentration polarization in Geometry 1 (tubular duct) and Geometry 2 (parallel plate). These are the geometries used in commercial membranes.

In the following simulations and according to the previous results, the solute mass fraction at the membrane wall is calculated with the correction factor Ξ method, and the permeate rate is calculated with the Film Theory method.

Case size

The case size of Geometry 1 is shown in Table 44.

Table 44. Size of Geometry 2 for the study of the effect of geometry

Periodic domain (Geom. 2)	
H (m)	$1 \cdot 10^{-4}$
L (m)	$2 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
$d_h(m)$	$4 \cdot 10^{-4}$

Initial and boundary conditions

The initial and boundary conditions of Geometry 1 are shown in Table 45 and Table 46.

Table 45. Initial conditions of Geometry 2 for the study of the effect of geometry

Initial conditions (Geom. 2)	
u	0.1
v	0
w_A	0
p	0

Table 46. Boundary conditions of Geometry 2 for the study of the effect of geometry

Boundary conditions. Periodic domain. (Geom. 2)				
	u	v	w_A	p
topWall	0	0	1	$\partial p/\partial y = 0$
bottomWall	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane
Inlet	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic
bottomEmptyFaces	empty	empty	empty	empty
topEmptyFaces	empty	empty	empty	empty

Solver

The simulation was solved with the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

Results

The results can be seen in Figure 35 and Figure 36. They show that parallel plates geometry gives a less concentration polarization effect and higher permeate flux than tubular duct.

6. Effect of spacers in concentration polarization

6.1. Rectangular spacers

It was studied the effect of rectangular spacers simulating the Geometry 4. The system was a periodic domain of parallel plates, with rectangular spacers in transverse direction to the flow.

Case size

Table 47. Size of Geometry 4 for the study of the effect of spacers

Periodic domain (Geom. 4)	
H (m)	$1 \cdot 10^{-4}$
L (m)	$2 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
a (m)	$5 \cdot 10^{-5}$
b (m)	$1 \cdot 10^{-4}$
d_h(m)	$4 \cdot 10^{-4}$

Initial and boundary conditions

Table 48. Initial conditions of Geometry 4 for the study of the effect of spacers

Initial conditions. Periodic domain. (Geom. 4)	
u	0.1
v	0
w_A	0
p	0

Table 49. Boundary conditions of Geometry 4 for the study of the effect of spacers

Boundary conditions. Periodic domain. (Geom. 4)				
	u	v	w_A	p
topWall	0	0	1	$\partial p/\partial y = 0$
bottomWall	0	0	1	$\partial p/\partial y = 0$
ObsTop	0	0	$\partial w_A/\partial y = 0$	$\partial p/\partial y = 0$
ObsBottom	0	0	$\partial w_A/\partial y = 0$	$\partial p/\partial y = 0$
Inlet	cyclic	cyclic	cyclic	cyclic
Outlet	cyclic	cyclic	cyclic	cyclic
bottomEmptyFaces	empty	empty	empty	empty
topEmptyFaces	empty	empty	empty	empty

Solver

The simulation was solved with the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

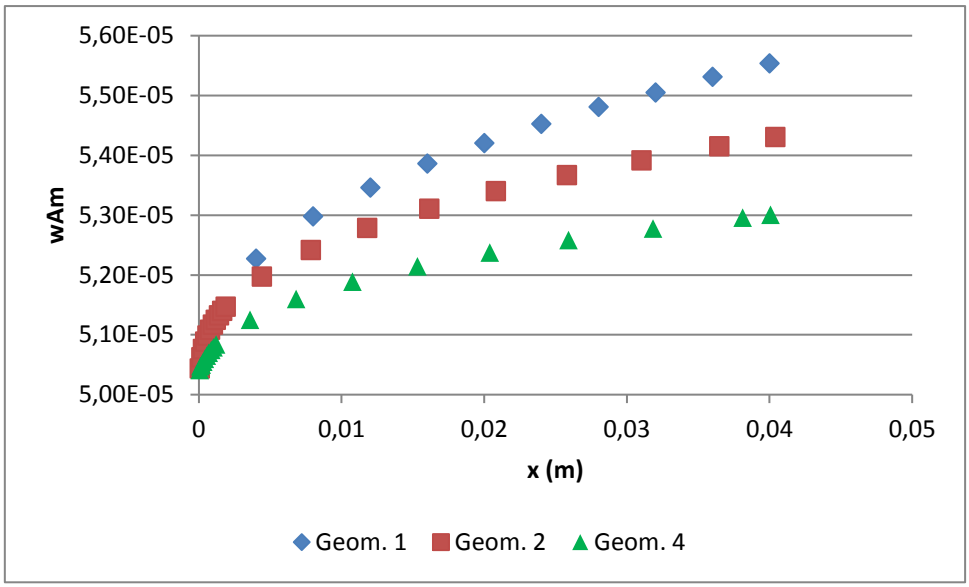


Figure 35. Solute mass fraction at the membrane wall vs x-direction in Geometry 1, Geometry 2 and Geometry 4

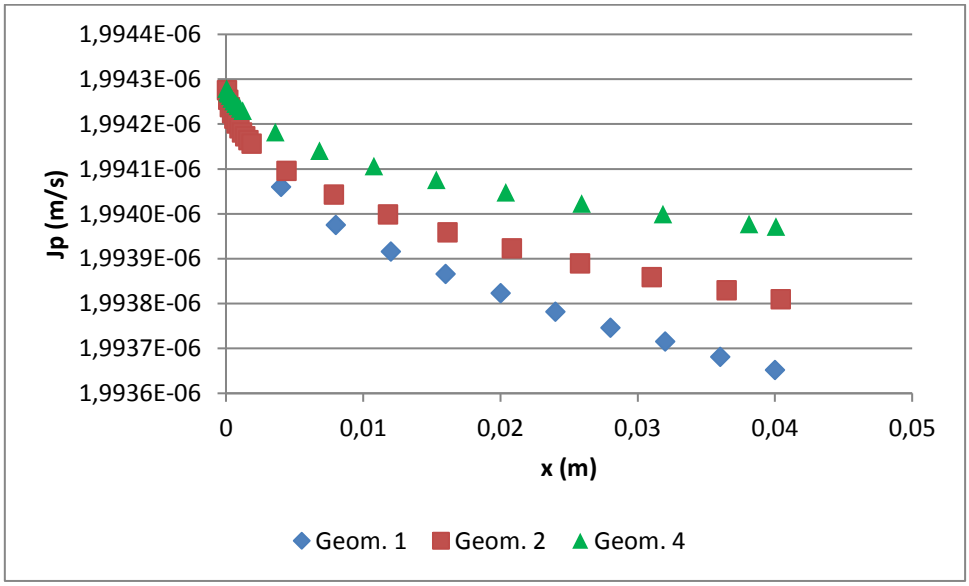


Figure 36. Permeate rates vs x-direction in Geometry 1, Geometry 2 and Geometry 4

It can be seen in the figures that the presence of rectangular spacers reduces the solute mass fraction at the membrane surface and consequently increases the permeate flux.

The operation with parallel plates allows reduce the permeate rate decline in a 28.6%, and the use of rectangular spacers reduces the decline in a 50 %.

6.2. Semi-cyclindrical spacers

It was studied the effect of semi-cylindrical spacers simulating the Geometry 5. The system was a periodic domain of parallel plates, with semi-cylindrical spacers in transverse direction to the flow.

Case size

Table 50. Size of Geometry 5 for the study of the effect of spacers

Periodic domain (Geom. 5)	
H (m)	$1 \cdot 10^{-4}$
L (m)	$2 \cdot 10^{-4}$
Z (m)	$1 \cdot 10^{-5}$
r (m)	$5 \cdot 10^{-5}$
d_h(m)	$4 \cdot 10^{-4}$

Initial and boundary conditions

The initial and boundary conditions are the same that in the previous simulation (Table 48 and Table 49).

Solver

The simulation was solved with the solver `MypimpleFoamU` and then `MypimpleFoamwA`.

Results

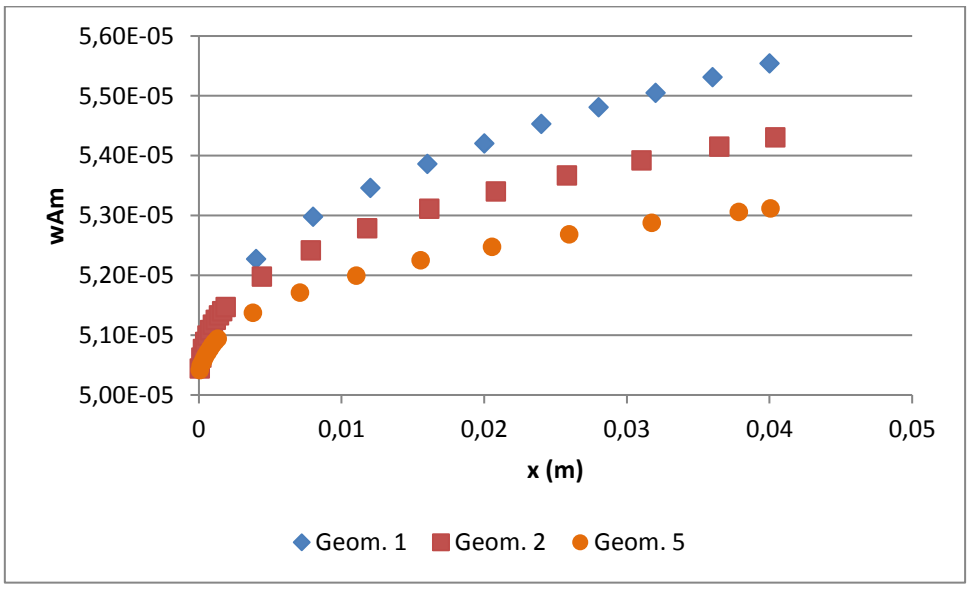


Figure 37. Solute mass fraction at the membrane wall vs x-direction in Geometry 1, Geometry 2 and Geometry 5

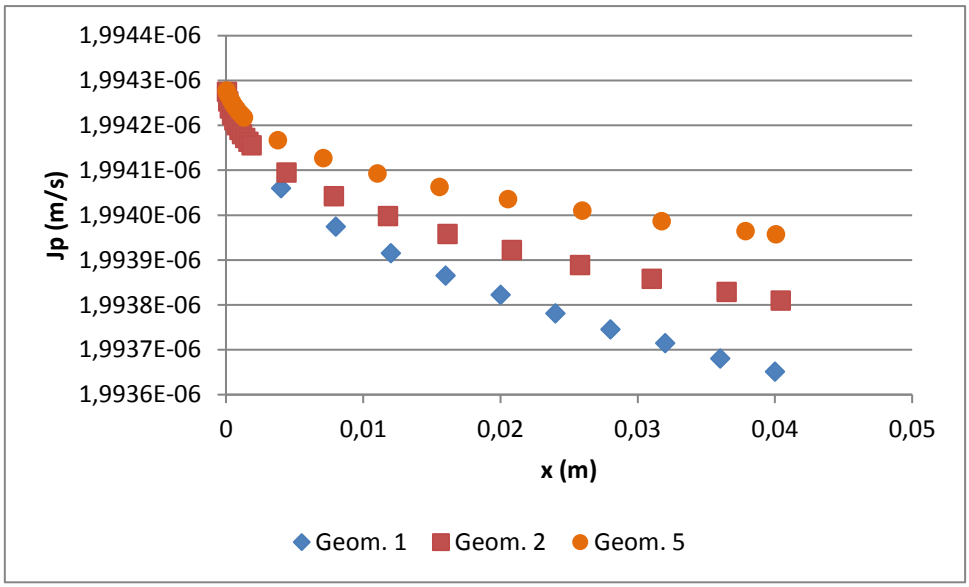


Figure 38. Permeate rates vs x-direction in Geometry 1, Geometry 2 and Geometry 5

It can be seen in the figures that the presence of semi-cylindrical spacers also reduces the solute mass fraction at the membrane surface and consequently increases the permeate rate.

The use of semi-cylindrical spacers reduces the permeate rate decline in a 49.7%.

6.3. Comparison between spacers

The geometry of the spacers is different but the dimensions are equivalent. Consequently, the effects of the two types of spacers are very similar. Rectangular spacers are a little better than semi-cylindrical, as it can be seen in the following figures.

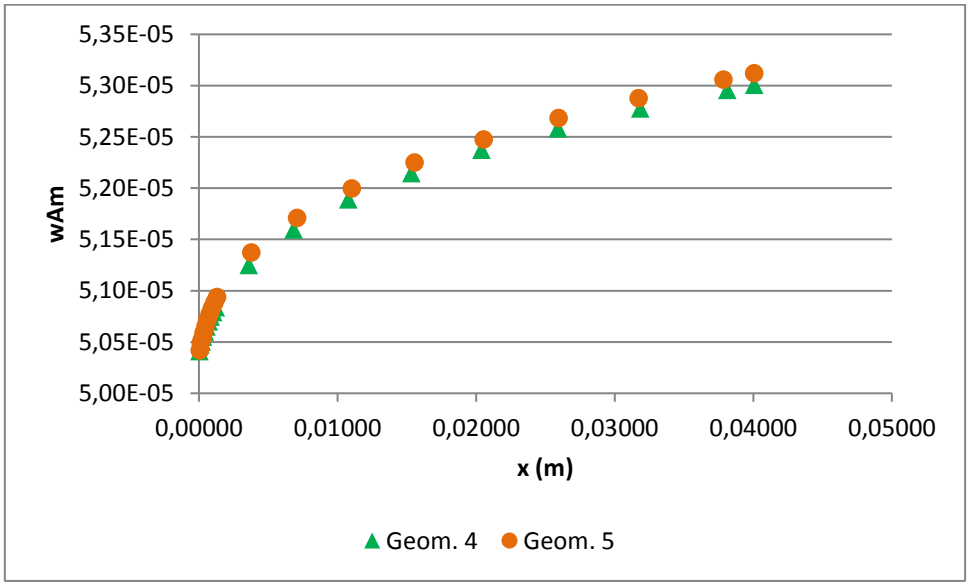


Figure 39. Solute mass fraction at the membrane wall vs x-direction in Geometry 4 and Geometry 5

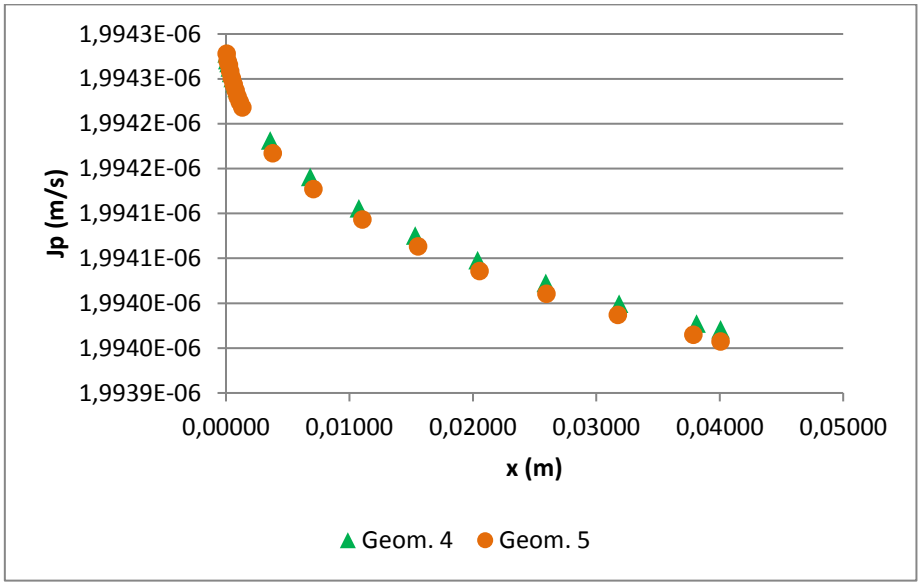


Figure 40. Permeate rates vs x-direction in Geometry 4 and Geometry 5

The use of rectangular spacers reduces the permeate rate decline in a 0.3% more than semi-cylindrical spacers.

7. Effect of transmembrane pressure in concentration polarization

7.1. Effect according to the geometry

It was calculated the permeate rate with three values of transmembrane pressure: 1 bar, 2 bar and 3 bar. The average permeation rate increases linearly with the transmembrane pressure, as it is logical and in consonance with Equation 14. A higher transmembrane pressure is beneficial in the sense that the average permeate rate increases, but it will also result in a higher concentration polarization, and consequently, a higher average mass solute fraction in the permeate flow. It is necessary the establishment of a commitment between the desirable permeate rate and the desirable limit in solute mass fraction in the permeate flow.

This effect of increasing in the average solute mass fraction in permeate is not equal in both geometries. As it can be seen in Figure 41, both in tubular duct and in parallel plates, the increasing in solute mass fraction in permeate is lineal, but it does not increase with the same magnitude. The increase in tubular geometry is higher than in parallel plates.

7.2. Effect according to the spacers

It was compared the transmembrane pressure effect when spacers are used. The presence of spacers provides a smaller increase of the average solute mass fraction in permeate, compared to the duct without spacers. The dependency of the average permeate rate with transmembrane pressure is in all the cases the same, but it is not the same for solute mass fraction in permeate.

Again the effect of both spacers is practically the same; a little better was the Geometry 4.

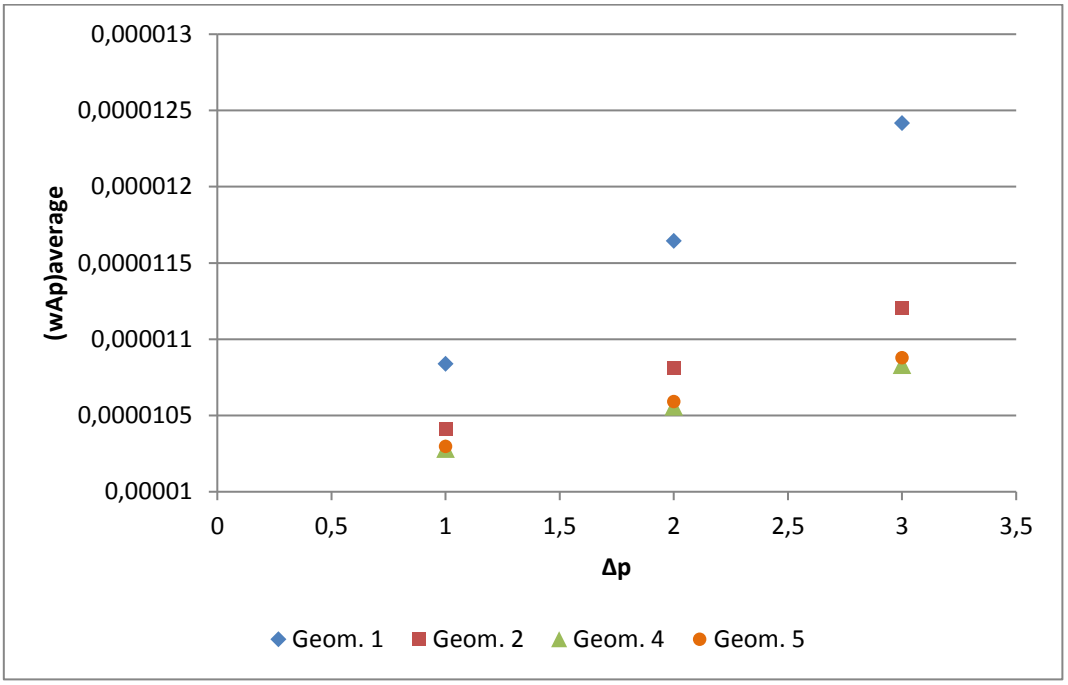


Figure 41. Average solute mass fraction at the permeate vs transmembrane pressure in Geometry 1, Geometry 2, Geometry 4 and Geometry 5

Geometry 1:

$$\overline{w_{Ap}} = 8 \cdot 10^{-7}\Delta p + 1 \cdot 10^{-5} \qquad (46)$$

$$R^2 = 0.9998$$

Geometry 2:

$$\overline{w_{Ap}} = 4 \cdot 10^{-7}\Delta p + 1 \cdot 10^{-5} \qquad (47)$$

$$R^2 = 0.9999$$

Geometry 3 and Geometry 4:

$$\overline{w_{Ap}} = 3 \cdot 10^{-7}\Delta p + 1 \cdot 10^{-5} \qquad (48)$$

$$R^2 = 1$$

Where w_{Ap} is the solute mass fraction at the permeate flow and Δp is the transmembrane pressure (bar).

The increase in transmembrane pressure from 1 bar to 3 bar gives an increase in the average solute mass fraction in permeate flux, which is 14.6% in Geometry 1, 7.6% in Geometry 2, 5.4% in Geometry 4 and 5.6% in Geometry 5.

Conclusions

The following conclusions are numbered according to the partial objectives of the work:

1. By using OpenFOAM the user can learn how to develop a fluid mechanic problem from the initial stage (creating the geometry, the mesh, etc.) to the very end, and also choose the numerical models to be used as well as wide range of choices in other aspects of the simulation. However, the interface is not as easy-to-use as in commercial CFD packages, so the user has to have basic knowledge about programming. As OpenFOAM is an open-source software, it seems to be a very interesting alternative to the commonly used commercial CFD packages.

2. It was created a new set of solvers: `MypimpleFoamU`, `MypimpleFoamwA` and `MypimpleFoamUW`. The first one solves the momentum transfer, the second one the mass transfer, and the last one both momentum and mass at the same time. These solvers can simulate membrane filtration process, solving transient and steady transfer, in 2D and 3D dimensions. Important parameters as discretization schemes, algorithms, time controls were selected so that the simulations were suitable.

3. The implementation of periodic boundary conditions allows simulate the system in just one periodic domain, which saves considerable computational load. The conservative laws were modified to consider periodic conditions, and they gave good results when the simulation was compared with the simulation without periodic conditions: the results were practically the same. Any system can be simulated with just one periodic domain which represents a repetitive portion of the geometry volume, as the momentum and mass transfer will be periodic after a certain length along the membrane.

4. It was demonstrated that 2D simulations with the solvers can describe the behavior of flat and parallel plate membranes in reasonable detail, as it was seen comparing the evolution of the Sherwood number versus $Pe(d_h/x)$ from the simulations and from the literature. The 3D simulation of a square duct also represented the behavior in a good way according to the literature results.

5. Through the correction of the inlet conditions between domains, a semi-permeable membrane was simulated with periodic domains. Considering the permeate flux allows a more realistic description of the hydrodynamics in the membrane than the usually impermeable approximation in simulation. With this simulation, concentration polarization can be also determined. The conjunction of realistic hydrodynamics and spatially variation of solute mass fraction at the membrane surface allows for the permeate flux to vary spatially throughout the membrane channel, and hence provide a more physical description of the channel hydrodynamics.

6. A generalized mass-transfer correction factor was obtained to correct mass transfer coefficients at impermeable walls, found to be independent of the membrane module geometry and the flow regime according to literature (Vitor Geraldes & Afonso, 2006). To find this factor, different values of the parameters c_1, c_2 and c_3 were proposed so that the results were as close as possible to those obtained in the simulation with permeation. Because this factor appears to be independent of the module geometry and the flow regime, it is suitable to predict the average concentration polarization index for the design and optimization of new membrane modules with complex geometries, because only a suitable correlation of Sherwood number must be found and corrected by the mass-transfer correction factor developed.

7. It was used both the well-known Film Theory method and the correction factor Ξ method to determine the concentration polarization. The film theory matches the CFD predictions worse than the correction factor, although the results can be accepted for low permeation rates ($\phi < 0.2$). The error increasing a lot for $\phi > 1$, so the Film Theory cannot predict the concentration polarization in an entirely correct way. Despite its simplicity, Film Theory is suitable for design purposes as long as $\phi < 0.3$ (low suction). The permeate rate was also calculated with both the Film Theory and the

correction factor Ξ, using iterative processes. For $\phi < 0.3$, both methods are good and give a low error in permeate rates according to the results obtained.

8. Studying the influence of several variables in concentration polarization, it was demonstrated the following:

    a. Effect of geometry
    Parallel pates configuration gives a lower concentration polarization because this geometry improves mass transfer compared to tubular geometry. This can also be seen in the asymptotic value reached by the Sherwood number in parallel plates, which is 7.55 while in tubular geometry is 3.658. Therefore, it is better to use parallel plates geometry if it is desirable a low concentration polarization and a high permeate flux. Parallel plates allows reduce the permeate rate decline in a 28.6%.

    b. Effect of spacers
    The presence of spacers gives a higher turbulence to the flux and in this way improves mass transfer, reducing concentration polarization. Both spacers produce similar effects, but rectangular spacers give some better results compared to semi-cylindrical spacers, which is logical because rectangular geometry is less 'smoother' than the semi-cylindrical, so it provides more turbulence. Rectangular spacers allow reduce the permeate rate decline in a 50%, and semi-cylindrical spacers in a 49.7%.

    c. Effect of transmembrane pressure
    The average permeate rate increases linearly with the transmembrane pressure according to the hydraulic permeability. However, this increase in permeate rate gives higher concentration polarization, and consequently, higher solute mass fraction in the permeate. So it is necessary the establishment of a commitment between the desirable permeate rate and the desirable limit in solute mass fraction in the permeate.

    This effect of increasing in the average solute mass fraction in the permeate is not equal in all the geometries. Parallel plates geometry gives better results than tubular duct. The increase in solute mass fraction with transmembrane pressure is linear in both cases, but is smaller in parallel plates, so it can be used a higher transmembrane pressure with less concentration polarization. With the tubular geometry, the average solute mass fraction in the permeate increases a 14.6% with transmembrane pressure (from 1 bar to 3 bar), while with parallel plates it increases a 7.6%.

    Regarding parallel plates with spacers, the increase in solute mass fraction in permeate in much lower than without spacers, which allows increase the pressure because the solute mass fraction will not increase a lot, although the economic cost of transmembrane pressure has to be considered. With rectangular spacers, the average solute mass fraction in the permeate increases a 5.4% with transmembrane pressure (from 1 bar to 3 bar), while with semi-cylindrical spacers it increases a 5.6%.

References

Aslak, U. (2014). *Computational Fluid Dynamics Simulations of Forward Osmosis Membrane Modules*. Technical University of Denmark.

Bergman, T. L., & Incropera, F. P. (2011). *Introduction to Heat Transfer*. (J. Wiley, Ed.).

Cao, Y. (2015). *Membranplatten des MBR-Verfahrens durch Einzel-luftblasen mit Hilfe von CFD*. Technische Universitat Darmstadt.

Carluccio, E., Starace, G., Ficarella, A., & Laforgia, D. (2005). Numerical analysis of a cross-flow compact heat exchanger for vehicle applications. *Applied Thermal Engineering*, *25*(13), 1995–2013. https://doi.org/10.1016/j.applthermaleng.2004.11.013

Cavaco Morão, A. I., Brites Alves, A. M., & Geraldes, V. (2008). Concentration polarization in a reverse osmosis/nanofiltration plate-and-frame membrane module. *Journal of Membrane Science*, *325*(2), 580–591. https://doi.org/10.1016/j.memsci.2008.08.030

Completo, C., Semiao, V., & Geraldes, V. (2016). Efficient CFD-based method for designing cross-flow nanofiltration small devices. *Journal of Membrane Science*, *500*. https://doi.org/10.1016/j.memsci.2015.11.012

Completo, C., Semiao, V., & Geraldes, V. (2016). Efficient CFD-based method for designing cross-flow nanofiltration small devices. *Journal of Membrane Science*, *500*, 190–202. https://doi.org/10.1016/j.memsci.2015.11.012

Darcovich, K., Dalcin, M., & Gros, B. (2009). Membrane mass transport modeling with the periodic boundary condition. *Computers & Chemical Engineering*, *33*(1), 213–224. https://doi.org/10.1016/j.compchemeng.2008.08.002

De, S., & Bhattacharya, P. K. (1997). Prediction of mass-transfer coefficient with suction in the applications of reverse osmosis and ultrafiltration. *Journal of Membrane Science*, *128*(2), 119–131. https://doi.org/10.1016/S0376-7388(96)00313-4

De Pinho, M. N., Semião, V., & Geraldes, V. V. (2002). Integrated modeling of transport processes in fluid/nanofiltration membrane systems. *Journal of Membrane Science*, *206*(1–2), 189–200. https://doi.org/10.1016/S0376-7388(01)00761-X

Ding, W. (2012). *Application of CFD in Membrane Technique*. Universität Duisburg-Essen.

Eldridge, B. (2014). Parallel Plate Heat Exchanger.

Fimbres-Weihs, G. A., & Wiley, D. E. (2007). Numerical study of mass transfer in three-dimensional spacer-filled narrow channels with steady flow. *Journal of Membrane Science*, *306*(1–2), 228–243. https://doi.org/10.1016/j.memsci.2007.08.043

Fimbres-Weihs, G. A., & Wiley, D. E. (2010). Review of 3D CFD modeling of flow and mass transfer in narrow spacer-filled channels in membrane modules. *Chemical Engineering and Processing: Process Intensification*, *49*(7), 759–781. https://doi.org/10.1016/j.cep.2010.01.007

Fimbres-Weihs, G. A., Wiley, D. E., & Fletcher, D. F. (2006). Unsteady flows with mass transfer in narrow zigzag spacer-filled channels: A numerical study. *Industrial and Engineering Chemistry Research*, *45*(19), 6594–6603. https://doi.org/10.1021/ie060243l

Fletcher, D. F., & Wiley, D. E. (2004). A computational fluids dynamics study of buoyancy effects in reverse osmosis. *Journal of Membrane Science*, *245*(1–2), 175–181. https://doi.org/10.1016/j.memsci.2004.07.023

Geraldes, V., & Afonso, M. D. (2006). Generalized Mass-Transfer Correction Factor for Nanofiltration and Reverse Osmosis. *AIChE Journal*, *52*, 3353–3362. https://doi.org/10.1002/aic

Geraldes, V. M., Semião, V. a., & de Pinho, M. N. (1998). Nanofiltration Mass Transfer at the Entrance Region of a Slit Laminar Flow. *Industrial & Engineering Chemistry Research*, *37*, 4792–4800. https://doi.org/10.1021/ie980198k

Geraldes, V., Semiao, V., & De Pinho, M. N. (2004). Concentration polarisation and flow structure within nanofiltration spiral-wound modules with ladder-type spacers. *Computers and Structures*, *82*(17–19), 1561–1568. https://doi.org/10.1016/j.compstruc.2004.03.052

Geraldes, V., Semião, V., & De Pinho, M. N. (2001). Flow and mass transfer modelling of nanofiltration. *Journal of Membrane Science*, *191*(1–2), 109–128. https://doi.org/10.1016/S0376-7388(01)00458-6

Geraldes, V., Semião, V., & Norberta Pinho, M. (2000). Numerical modelling of mass transfer in slits with semi-permeable membrane walls. *Engineering Computations*, *17*(3), 192–218. https://doi.org/10.1108/02644400010324857

Ghidossi, R., Veyret, D., & Moulin, P. (2006). Computational fluid dynamics applied to membranes: State of the art and opportunities. *Chemical Engineering and Processing: Process Intensification*, *45*(6), 437–454. https://doi.org/10.1016/j.cep.2005.11.002

Gill, W. N., Wiley, D. E., Fell, C. J. D., & Fane, A. G. (1988). Effect of viscosity on concentration polarization in ultrafiltration. *AIChE Journal*, *34*(9), 1563–1567. https://doi.org/10.1002/aic.690340919

Greenshields, C. J. (2016). OpenFOAM User Guide.

Juretic, F. (2010). cfMesh User Guide. https://doi.org/10.1007/SpringerReference_27988

Keir, G. P. (2012). Coupled modelling of hydrodynamics and mass transfer in membrane filtration by, (Hons I).

Koch, J. (2015). Genetic Algorithm Based Optimization of Baffle Positions in a Forward Osmosis Draw Channel. *Engineering and Applied Science Theses & Dissertations*, 171. https://doi.org/10.7936/K76M350H

Koutsou, C. P., Yiantsios, S. G., & Karabelas, A. J. (2009). A numerical and experimental study of mass transfer in spacer-filled channels: Effects of spacer geometrical characteristics and Schmidt number. *Journal of Membrane Science*, *326*(1), 234–251. https://doi.org/10.1016/j.memsci.2008.10.007

Lau, K. K., Abu Bakar, M. Z., Ahmad, A. L., & Murugesan, T. (2009). Feed spacer mesh angle: 3D modeling, simulation and optimization based on unsteady hydrodynamic in spiral wound membrane channel. *Journal of Membrane Science*, *343*(1–2), 16–33. https://doi.org/10.1016/j.memsci.2009.07.001

Li, B., Feng, B., He, Y.-L., & Tao, W.-Q. (2006). Experimental study on friction factor and numerical simulation on flow and heat transfer in an alternating elliptical axis tube. *Applied Thermal Engineering*, *26*(17–18), 2336–2344. https://doi.org/10.1016/j.applthermaleng.2006.03.001

Ma, S., & Song, L. (2006). Numerical study on permeate flux enhancement by spacers in a crossflow reverse osmosis channel. *Journal of Membrane Science*, *284*(1–2), 102–109. https://doi.org/10.1016/j.memsci.2006.07.022

Miranda, J. M., & Campos, J. B. L. M. (2002). Mass transfer in the vicinity of a separation membrane - The applicability of the stagnant film theory. *Journal of Membrane Science*, *202*(1–2), 137–150. https://doi.org/10.1016/S0376-7388(01)00747-5

Patankar, S. V., Liu, C. H., & Sparrow, E. M. (1977). Fully Developed Flow and Heat Transfer in Ducts Having Streamwise-Periodic Variations of Cross-Sectional Area. *Journal of Heat Transfer*, *99*(2), 180. https://doi.org/10.1115/1.3450666

Pawlowski, S., Geraldes, V., Crespo, J. G., & Velizarov, S. (2016). Computational fluid dynamics (CFD) assisted analysis of profiled membranes performance in reverse electrodialysis. *Journal of Membrane Science*, *502*, 179–190. https://doi.org/10.1016/j.memsci.2015.11.031

Rodrigues, C., Geraldes, V., de Pinho, M. N., & Semião, V. (2012). Mass-transfer entrance effects in narrow rectangular channels with ribbed walls or mesh-type spacers. *Chemical Engineering Science*, *78*, 38–45. https://doi.org/10.1016/j.ces.2012.04.023

Rosaguti, N. R., Fletcher, D. F., & Haynes, B. S. (2005). Laminar flow and heat transfer in a periodic serpentine channel. *Chemical Engineering and Technology*, *28*(3), 353–361. https://doi.org/10.1002/ceat.200407148

Santos, J. L. C., Crespo, J. G., & Geraldes, V. (2010). OpenFOAM simulation of mass transfer in spacer-filled channels. *European Conference on Computational Fluid Dynamics*, (June), 14–17. Retrieved from http://web.univ-ubs.fr/limatb/EG2M/Disc_Seminaire/ECCOMAS-CFD2010/papers/01454.pdf

Schretter, P. (2016). *Simulation of Membrane Modules with OpenFOAM*. Technische Universität Wien.

Schwinge, J., Wiley, D. E., & Fletcher, D. F. (2003). Simulation of Unsteady Flow and Vortex Shedding for Narrow Spacer-Filled Channels. *Ind. Eng. Chem*, *42*, 4962–4977. https://doi.org/10.1021/ie030211n

Welty, J. R., Wicks, C. E., Wilson, R. E., & Rorrer, G. L. (2008). *Fundamentals of Momentum, Heat, and Mass Transfer*. https://doi.org/10.1016/0017-9310(70)90063-3

Wiley, D., Fell, C., & Fane, A. G. (1985). Optimisation of Membrane Module Design for Brackish Water Desalination. *Desalination*, *52*, 249–265.

Yuan, Z. X., Tao, W. Q., & Wang, Q. W. (1998). Numerical prediction for laminar forced convection heat transfer in parallel-plate channels with streamwise-periodic rod disturbances. *International Journal for Numerical Methods in Fluids*, *28*(9), 1371–1387. https://doi.org/10.1002/(SICI)1097-0363(19981215)28:9<1371::AID-FLD774>3.0.CO;2-A

Appendix

BlockMeshDict

Geometry 1

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 0.0001;

vertices
(
    (0 0 0.05)
    (4 0 0.05)
    (0 2 0)
    (4 2 0)
    (0 2 0.1)
    (4 2 0.1)
);
blocks
(
    hex (0 1 3 2 0 1 5 4) (100 100 1) simpleGrading (1 0.3 1)
);
edges
(
);
boundary
(
    bottomWall
    {
        type            empty;
        faces ((0 1 1 0));
    }
    topWall
    {
        type            wall;
        faces ((2 4 5 3));
    }
    bottomEmptyFaces
    {
        type            wedge;
        faces           ((0 2 3 1));
    }
    topEmptyFaces
    {
        type            wedge;
```

```
        faces ((0 1 5 4));
    }
    outlet
    {
        type            cyclic;
        neighbourPatch  inlet;
        faces ((1 3 5 1));
    }
    inlet
    {
        type            cyclic;
        neighbourPatch  outlet;
        faces ((0 4 2 0));
    }

);
mergePatchPairs
(
);

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

checkMesh

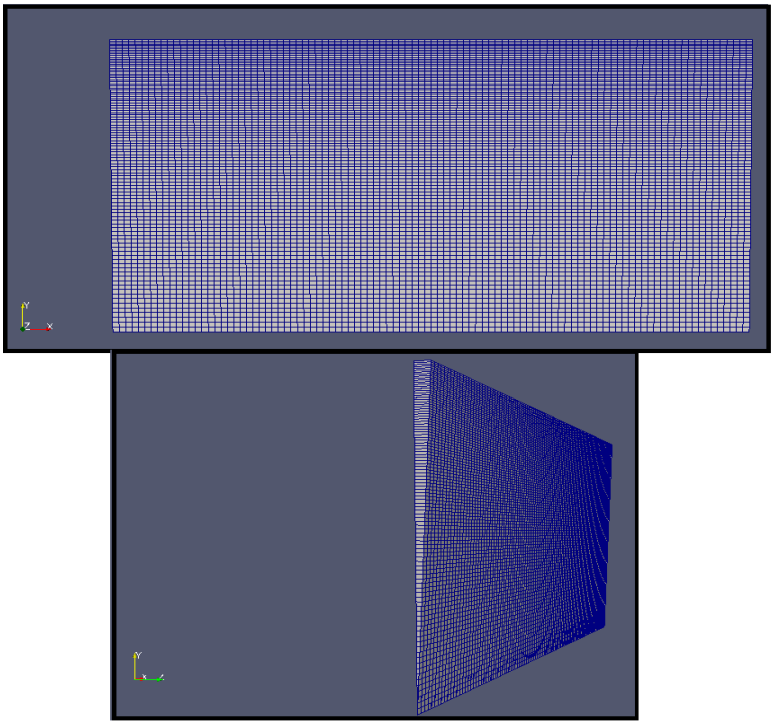
Geometry 1

```
/*---------------------------------------------------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
Build  : 3.0.1-d8a290b55d28
Exec   : checkMesh
Date   : Jan 26 2017
Time   : 15:58:49
Host   : "marta-VirtualBox"
PID    : 9548
Case   : /home/marta/OpenFOAM/marta-
3.0.1/run/TFM/GEOMETRIAS/MypimpleFoam_rev03_cyclic/channel395
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Allowing user-supplied system call operations

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

Create time

Create polyMesh for time = 0

Time = 0

Mesh stats
  points:           20301
  internal points:  0
  faces:            40100
  internal faces:   19800
  cells:            10000
  faces per cell:   5.99
  boundary patches: 6
  point zones:      0
  face zones:       0
  cell zones:       0

Overall number of cells of each type:
  hexahedra:     9900
  prisms:        100
  wedges:        0
  pyramids:      0
  tet wedges:    0
  tetrahedra:    0
  polyhedra:     0
```

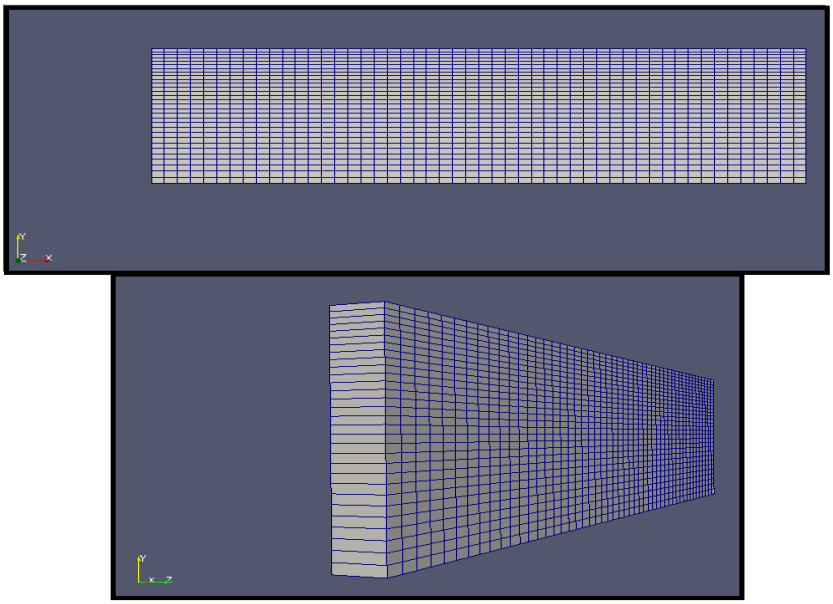
```
Checking topology...
  Boundary definition OK.
  Cell to face addressing OK.
  Point usage OK.
  Upper triangular ordering OK.
  Face vertices OK.
  Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
  Patch             Faces    Points   Surface topology
  bottomWall        0        0        ok (empty)
  topWall           100      202      ok (non-closed singly connected)
  bottomEmptyFaces  10000    10201    ok (non-closed singly connected)
  topEmptyFaces     10000    10201    ok (non-closed singly connected)
  outlet            100      201      ok (non-closed singly connected)
  inlet             100      201      ok (non-closed singly connected)

Checking geometry...
  Overall domain bounding box (0 0 0) (0.0004 0.0002 1e-05)
  Mesh has 2 geometric (non-empty/wedge) directions (1 1 0)
  Mesh has 3 solution (non-empty) directions (1 1 1)
  Wedge bottomEmptyFaces with angle 1.432096184 degrees
  Wedge topEmptyFaces with angle 1.432096184 degrees
  All edges aligned with or perpendicular to non-empty directions.
  Boundary openness (1.138354132e-18 1.144049779e-15 -4.192159356e-16) OK.
  Max cell openness = 2.123978664e-16 OK.
  Max aspect ratio = 3.890702404 OK.
  Minimum face area = 2.951226731e-13. Maximum face area = 4e-11.  Face area magnitudes OK.
  Min volume = 1.180490692e-18. Max volume = 4.912538618e-17.  Total volume = 4e-13.  Cell volumes
OK.
  Mesh non-orthogonality Max: 0 average: 0
  Non-orthogonality check OK.
  Face pyramids OK.
  Max skewness = 0.3325005205 OK.
  Coupled point location match (average 7.897149283e-22) OK.

Mesh OK.
```

Meshes with BlockMeshDict

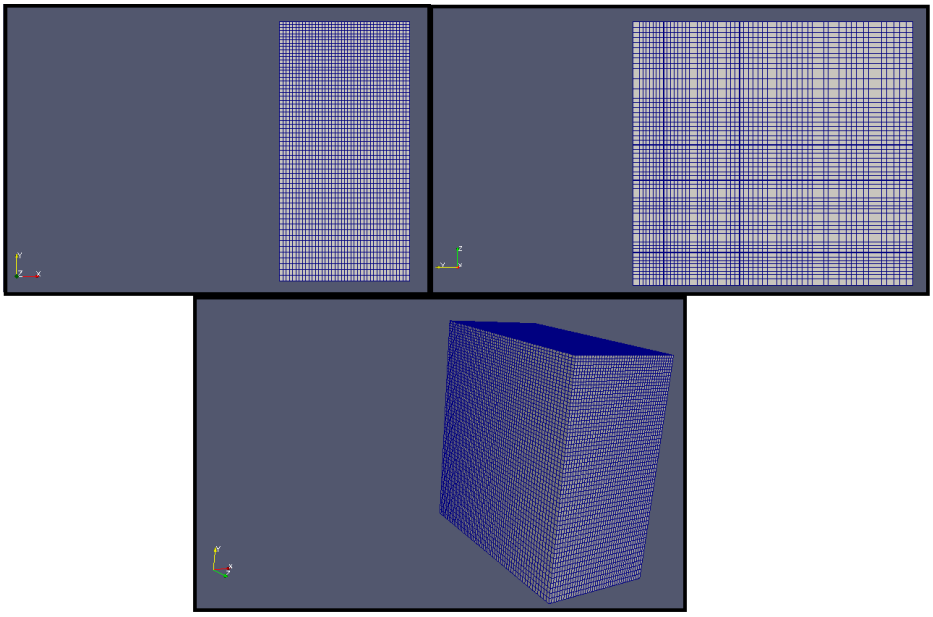
Geometry 1



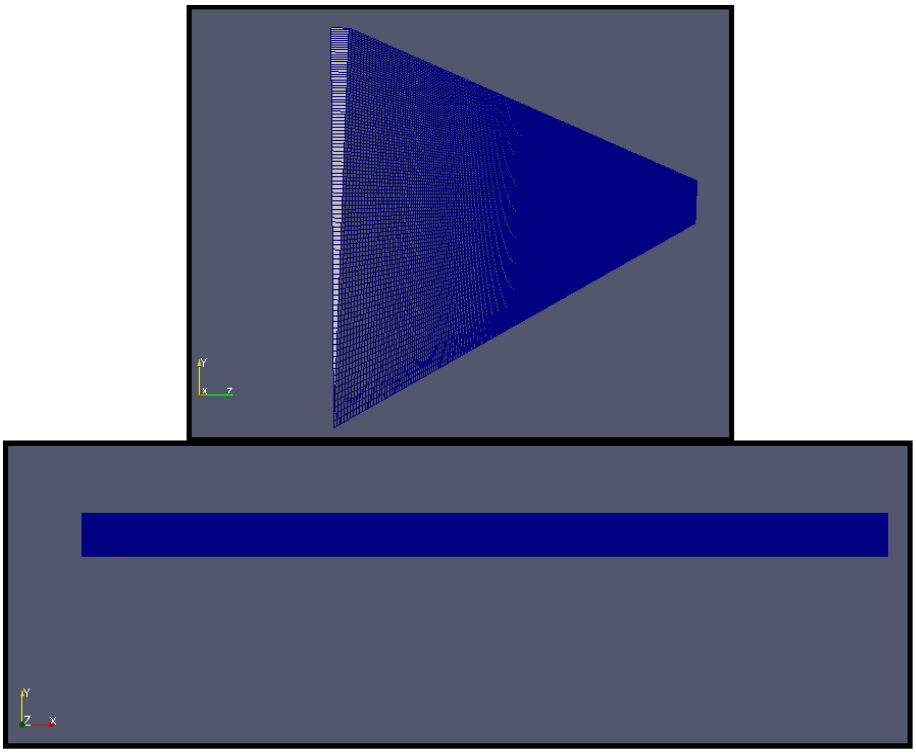
Geometry 2



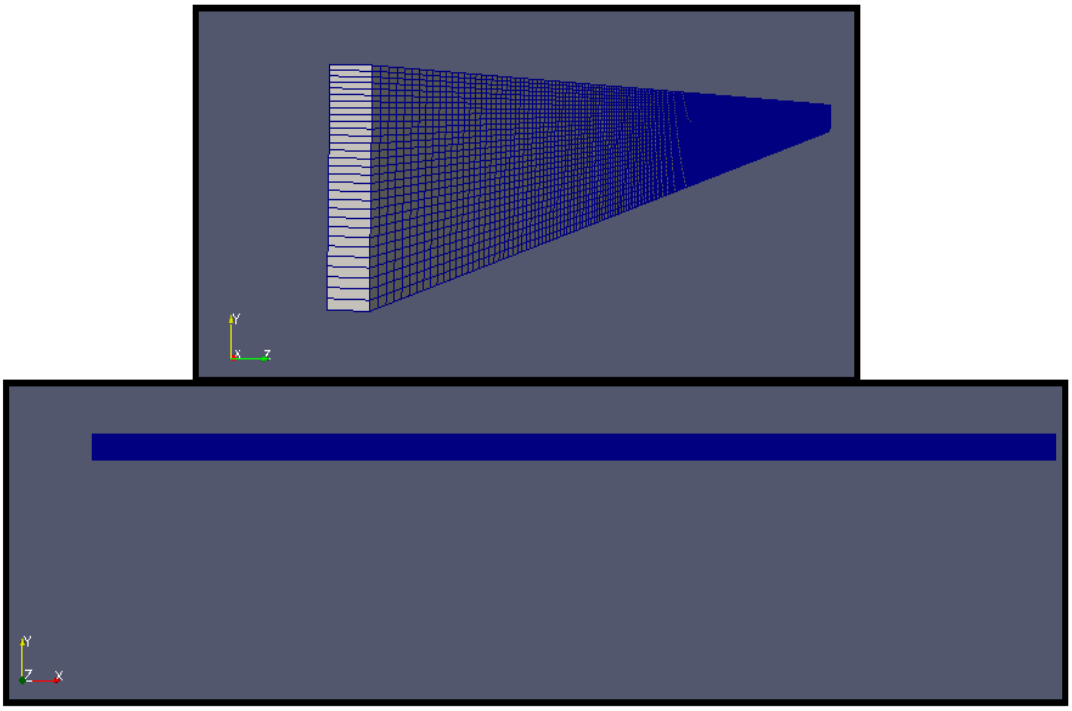
Geometry 3



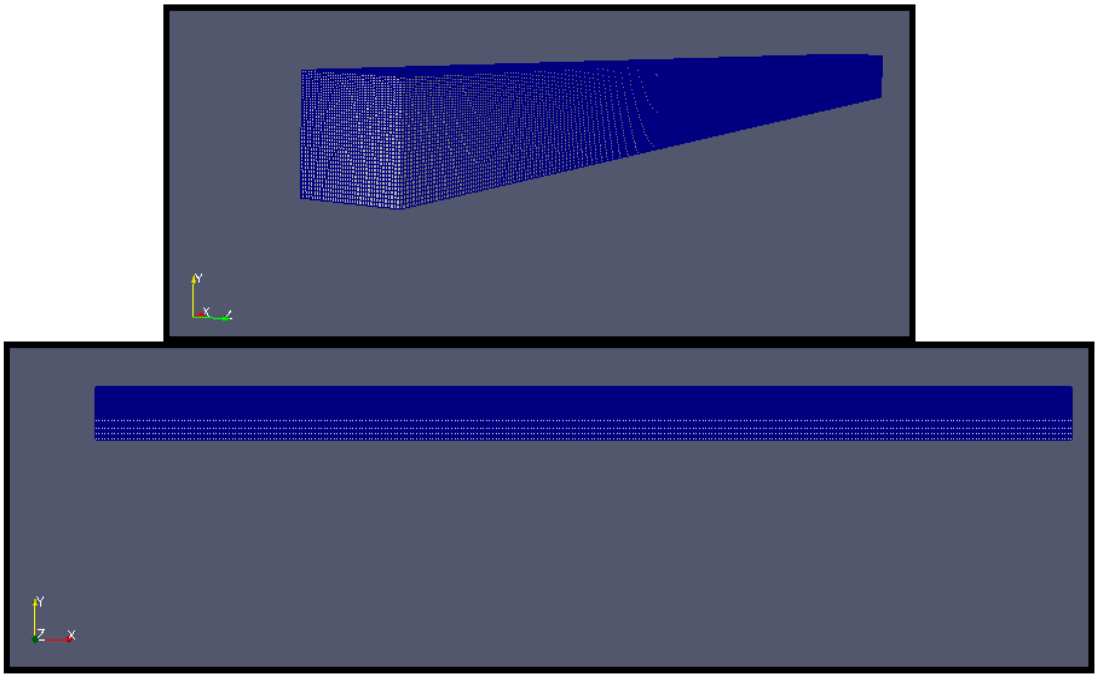
Long Geometry 1



Long Geometry 2

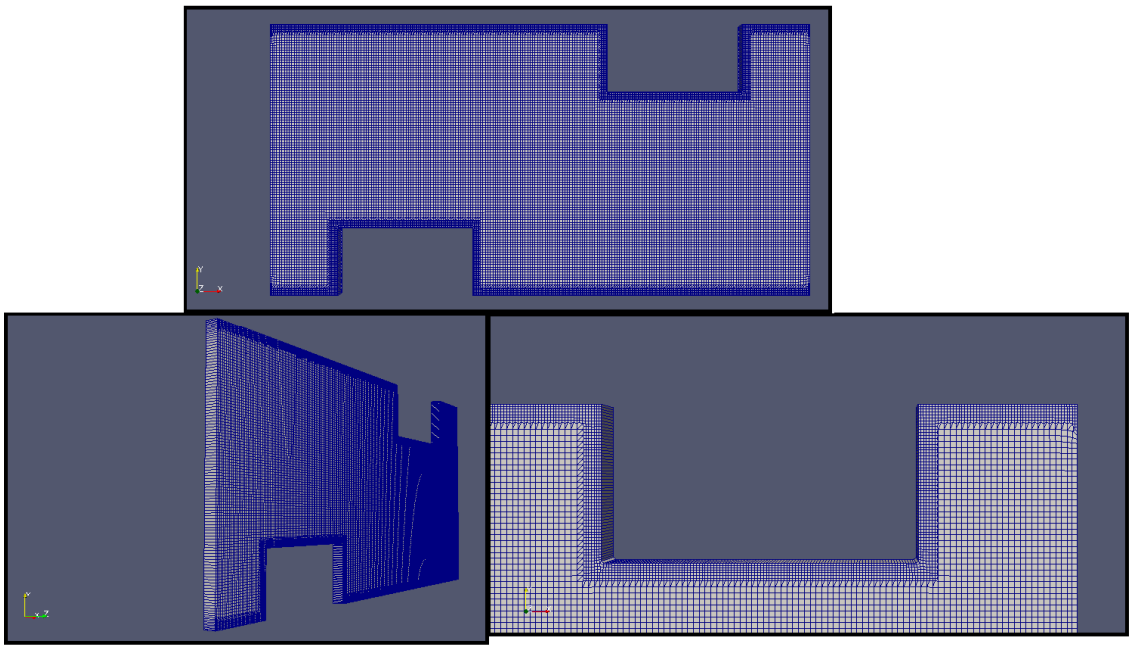


Long Geometry 3

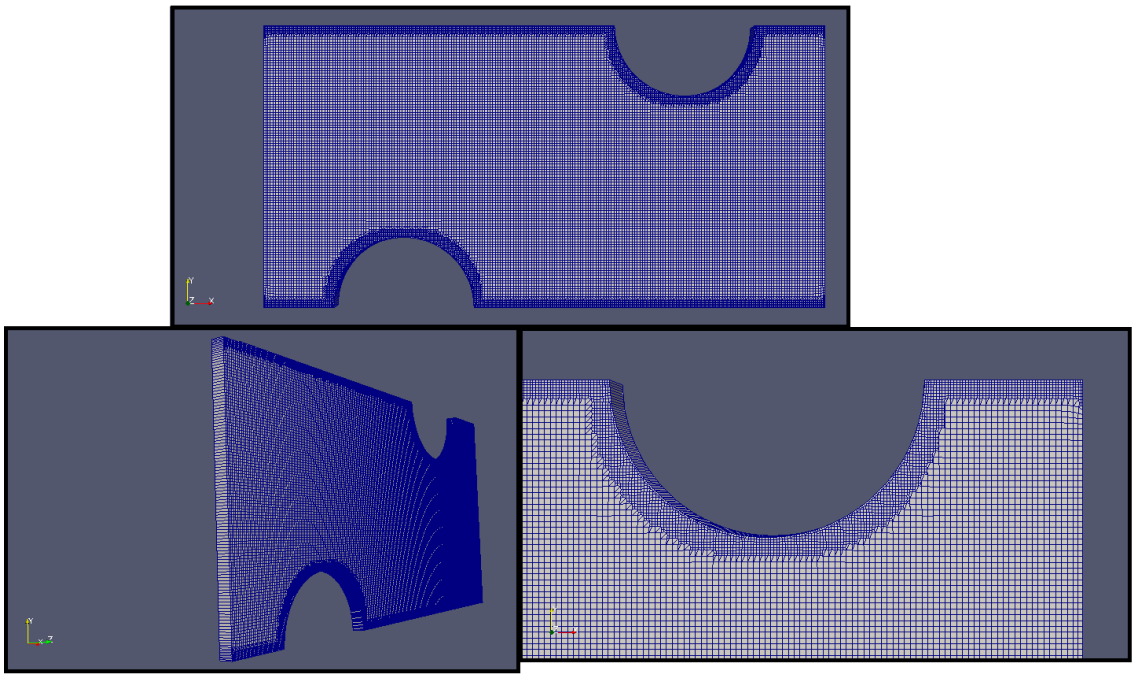


Meshes with cfMesh

Geometry 4



Geometry 5



MeshDict

Geometry 4 and Geometry 5

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM GUI Project: creativeFields            |
|  \\    /   O peration     | Version:  0.8.9.0                               |
|   \\  /    A nd           | Web: www.c-fields.com                           |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

FoamFile
{
version    2;
format     ascii;
class      dictionary;
location   "system";
object     meshDict;
}

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

maxCellSize          0.00000184;

surfaceFile          "geom.stl";

localRefinement
{
   "topWall.*"
   {
      additionalRefinementLevels 1;
      refinementThickness 0.000005;
   }
   "bottomWall.*"
   {
      additionalRefinementLevels 1;
      refinementThickness 0.000005;
   }
   "ObsTop.*"
   {
      additionalRefinementLevels 1;
      refinementThickness 0.000005;
   }

   "ObsBottom.*"
   {
      additionalRefinementLevels 1;
      refinementThickness 0.000005;
   }
boundaryCellSize 0.0000005;
}

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

Boundary files

Geometry 4 and Geometry 5

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       polyBoundaryMesh;
    location    "constant/polyMesh";
    object      boundary;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

8
(
    bottomWall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          328;
        startFace       52760;
    }
    ObsBottom
    {
        type            wall;
        inGroups        1(wall);
        nFaces          220;
        startFace       53088;
    }
    topWall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          328;
        startFace       53308;
    }
    ObsTop
    {
        type            wall;
        inGroups        1(wall);
        nFaces          220;
        startFace       53636;
    }
    bottomEmptyFaces
    {
        type            empty;
        inGroups        1(empty);
        nFaces          26578;
        startFace       53856;
    }
```

```
    topEmptyFaces
    {
        type            empty;
        inGroups        1(empty);
        nFaces          26578;
        startFace       80434;
    }
    inletCyclic
    {
        type            cyclic;
        inGroups        1(cyclic);
        nFaces          120;
        startFace       107012;
        matchTolerance  0.0001;
        transform       unknown;
        neighbourPatch  outletCyclic;
    }
    outletCyclic
    {
        type            cyclic;
        inGroups        1(cyclic);
        nFaces          120;
        startFace       107132;
        matchTolerance  0.0001;
        transform       unknown;
        neighbourPatch  inletCyclic;
    }
)

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

controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     MypimpleFoamU;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         30;

deltaT          0.0001;

writeControl    adjustableRunTime;

writeInterval   0.01;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

adjustTimeStep yes;

maxCo 1;

maxDeltaT 2;

libs ("libOpenFOAM.so"
     "libsimpleSwakFunctionObjects.so"
     "libswakFunctionObjects.so"
     "libgroovyBC.so"
```

```
);


functions
{
   fieldAverage1
   {
      type          fieldAverage;
      functionObjectLibs ( "libfieldFunctionObjects.so" );
      enabled       true;
      outputControl   outputTime;

      fields
      (
         U
         {
            mean       on;
            prime2Mean  on;
            base       time;
         }

         p
         {
            mean       on;
            prime2Mean  on;
            base       time;
         }

         wA
         {
            mean       on;
            prime2Mean  on;
            base       time;
         }

      );
   }
}

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

fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss linearUpwind grad(U);
    div(phi,wA)     Gauss limitedLinear01 1;
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear limited 1;

}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited 1;
}
// ************************************************************************
```

fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        tolerance       0;
        relTol          0.001;
        smoother        GaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }
    pFinal
    {
        $p;
        smoother        DICGaussSeidel;
        tolerance       1e-12;
        relTol          0;
    }
    "(U|wA)"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-12;
        relTol          0.001;
    }
    "(U|wA)Final"
    {
        $U;
        tolerance       1e-12;
        relTol          0;
    }
}

PIMPLE
{
    nOuterCorrectors 4;
    nCorrectors     3;
    nNonOrthogonalCorrectors 0;
```

```
    pRefCell        0;
    pRefValue       0;
}

relaxationFactors
{
    fields
    {
        p 0.5;
    }
    equations
    {
        U  0.8;
        wA 0.8;
    }
}
// ************************************************************************* //
```

createPatchDict

Geometry 4 and Geometry 5

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM GUI Project: creativeFields          |
|  \\    /   O peration     | Version:  0.8.9.0                             |
|   \\  /    A nd           | Web: www.c-fields.com                           |
|    \\/     M anipulation  |                                               |
\*---------------------------------------------------------------------------*/

FoamFile
{
version    2;
format     ascii;
class      dictionary;
location   "system";
object     createPatchDict;
}

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

pointSync true;

patches
(
   {
      // Name of new patch
      name inletCyclic;
      // Dictionary to construct new patch from
      patchInfo
      {
         type cyclic;
         neighbourPatch outletCyclic;
      }
      constructFrom patches;
      patches (inlet);
      set f0;
   }
   {
      name outletCyclic;
      patchInfo
      {
         type cyclic;
         neighbourPatch inletCyclic;
      }
      constructFrom patches;
      patches (outlet);
      set f0;
   }


);

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