

**RIB-ROUGHENED HEAT AND MASS TRANSFER ENHANCEMENT IN
MEMBRANE-BASED ENERGY RECOVERY VENTILATORS**

by

Alexander Sylvester

B.Sc., McGill University, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

October 2017

© Alexander Sylvester, 2017

Abstract

Energy Recovery Ventilators (ERVs) save energy by recovering sensible and latent heat from exhaust ventilation streams. They consist of compact channels in a cross- or counter-flow arrangement separated by moisture-permeable membranes. This work employs computational fluid dynamics with experimental validation to study angled rib mixing features to enhance heat and mass transport in membrane-based ERVs. The model simulates air-to-air heat and mass exchange between two rectangular ducts in a counter-flow arrangement. Local results from these simulations are used in a mathematical model to predict the effectiveness of a cross-flow ERV. The flow is steady and laminar. The channels are modeled with and without ribs for various channel aspect ratios and flow rates. Periodic inlet/outlet conditions are assumed for all cases to highlight the effect of the ribs. Ribs are either on the membrane surface or the opposite wall, however ribs on the membrane surface are the focus of this work due to their superior performance.

Results show that the ribs increase the channel Sherwood and Nusselt numbers by a larger fraction than the corresponding increase in friction factor. The improvement is larger for sensible recovery than latent recovery because the membrane's mass-transfer resistance is higher than its thermal resistance. For a typical commercial grade, counter-flow ERV, total effectiveness can be improved by over 10% for an equal pressure drop by adding ribs and slightly increasing the channel height.

Mass-transfer and pressure drop results from the simulations were validated experimentally for channels with and without ribs using a commercially available membrane (dPoint Technologies).

Ribs were either formed into the membrane surface or made as extensions of the channel wall. The experimental and simulation results agree with one another within the experimental uncertainty of the test apparatus and variability of the membrane permeability.

Typically, ERV performance targets (recovery effectiveness, pressure drop) are met by varying the channel dimensions or number of channels. The work presented here indicates that angled ribs could be used instead, which would not require altering the ERV footprint or using extra materials.

Lay Summary

Energy Recovery Ventilators (ERVs) are heat exchangers used in building ventilation systems. They are used to move heat and moisture between the outgoing and incoming building air streams and save energy. This is especially beneficial in hot, humid climates where air conditioners typically consume a lot of energy cooling and dehumidifying air.

ERVs consist of an array of channels separated by moisture-permeable membranes. This study investigates a way of improving the performance of ERVs by adding angled rib features to the channels. Most of the work was done with numerical simulations, with subsequent validation experiments.

Results show that the ribs increase the heat and moisture recovery, but also increase the fan power required to push air through the ERV. However, the overall impact is a net benefit. In fact, total effectiveness can be improved by over 10% for equal fan power by adding ribs and increasing the channel height.

Preface

The general research objective of geometric enhancement was identified by David Kadylak and Ryan Huizing of dPoint technologies and Professors Sheldon Green and Steven Rogak of the University of British Columbia. I subsequently identified specific geometries to study, setup and ran appropriate simulations, designed and executed experimental validation, and analyzed the resulting data. Simulations were performed using ANSYS Fluent®. The experimental testing apparatus described in Section 2.2.1 was designed and built in conjunction with PhD candidate Amin Engarnevis.

The preliminary work for this project was completed by MEng Student Rajitha Nakandala with supervision by Amin Engarnevis. It involved using ANSYS Fluent® to study mass transfer between two smooth triangular ducts in a counter-flow arrangement.

The Matlab model used to predict the cross-flow core performance with angled ribs in Section 3.2.2 was written by Amin Engarnevis and Steven Rogak.

The results described in Chapter 3 (except section 3.2.2) will be submitted for publication in the International Journal of Heat and Mass Transfer as “CFD Analysis of Rib-Roughened Channels in Membrane-Based Energy Recovery Ventilators”. The results of section 3.2.2 will be submitted for publication in the journal Energy and Buildings as “Conjugate heat and mass transfer modeling in membrane-based enthalpy exchangers: performance enhancement via rib-roughened channels”.

Table of Contents

Abstract.....	ii
Lay Summary	iv
Preface.....	v
Table of Contents	vi
List of Tables	ix
List of Figures.....	x
List of Symbols	xiii
List of Abbreviations	xiv
Acknowledgements	xv
Chapter 1: Introduction	1
1.1 Energy Recovery in Air Conditioning	1
1.1.1 Building Ventilation.....	1
1.1.2 Energy Recovery Ventilators.....	2
1.1.3 Other Applications	5
1.2 Theory.....	6
1.2.1 Heat and Mass Transport	7
1.2.2 Membrane Transport.....	8
1.2.3 Performance Metrics.....	10
1.2.4 Periodic Boundary Conditions.....	13
1.2.5 Geometric Enhancement.....	15
1.3 Current Study Objectives.....	19
Chapter 2: Methods	20

2.1	Simulation	20
2.1.1	Problem Description	20
2.1.2	Simulation Parameters	22
2.1.3	Membrane Modelling Approach.....	22
2.1.4	Convergence Criteria	23
2.1.5	Meshing and Mesh Validation.....	23
2.1.6	Post Processing	27
2.2	Experimental	28
2.2.1	Water Vapour Transport Station.....	28
2.2.2	Testing Module	29
2.2.3	Membrane Forming	31
Chapter 3: Results and Discussion		33
3.1	Simulations	33
3.1.1	Membrane Ribs.....	33
3.1.2	Ribs on Walls.....	41
3.2	Effectiveness Predictions.....	43
3.2.1	Counter-Flow Channel.....	43
3.2.2	Cross-Flow	45
3.3	Experimental Validation	51
Conclusions.....		57
References.....		59
Appendix A – User Defined Functions.....		61
A.1	Membrane transport, Nusselt number, Sherwood number, and Friction Factor.....	61

A.2	Periodic Boundary Conditions.....	70
A.3	Rectangular Channel Fully Developed Velocity Profile	79

List of Tables

Table 1.1 Previous studies investigating heat transfer enhancement in ducts with angled ribs ...	17
Table 2.1 Dimensions of the housing slot.....	31
Table 3.2 Measured dimensions of the flow channel inserts	55
Table 3.3 Air properties assumed at 50°C and 1atm	55

List of Figures

Figure 1.1 - Typical thermal comfort zone on a psychrometric chart	1
Figure 1.2 – Function of an ERV	3
Figure 1.3 – Composite polymer membrane construction	4
Figure 1.4 – ERV Construction. Cross-flow (left) and counter-flow (right).....	5
Figure 1.5 - dPoint fuel cell humidifier	6
Figure 1.6 – Membrane permeation/separation modes	9
Figure 1.7 - Hydrodynamically developing velocity profile	13
Figure 1.8 - Resistance in series model	16
Figure 1.9 - Common enhancement geometries	17
Figure 2.1 – Schematic of the simulation domain	21
Figure 2.2 – Isometric view of the mesh, mesh spacing has been doubled in each dimension for clarity	25
Figure 2.3 – Various views of the mesh. Mesh spacing has been doubled in each dimension for clarity	26
Figure 2.4 – Mesh sensitivity study	27
Figure 2.5 – Experimental test station	29
Figure 2.6 – Aluminum housing of testing module	30
Figure 2.7 – Flow configuration in the membrane housing module.....	31
Figure 2.8 - Membrane forming template. Dimension are in inches.....	32
Figure 3.1 - Transverse velocity fields 1, 5, and 8 step widths downstream of the rib for the middle of the three flow rates. (a) AR = 0.6, Re = 390. (b) AR = 1, Re = 468. (c) AR = 1.4, Re = 585.....	35

Figure 3.2 - Transverse velocity fields 1, 5, and 8 step widths downstream of the rib for AR = 1. (a) Re =703. (b) Re = 468. (c) Re = 234.....	36
Figure 3.3 – Speed contour plot for the AR = 1, Re = 468 case.....	37
Figure 3.4 – Contour plots of humidity (top) and temperature (bottom) on the membrane surface for the AR = 1, Re = 468 case. Looking down to on the feed side. Red (blue) indicates highest (lowest) humidity and temperature.	37
Figure 3.5 – Local performance data along the channel for AR = 1 and Re = 468.....	38
Figure 3.6 – Performance data of the ribbed and smooth channels for three different flow rates. (a) Nusselt number. (b) Sherwood number. (c) Friction factor times Reynolds number. (d) Relative performance of ribbed and smooth channels for the AR=1 case.....	41
Figure 3.7 - Diagram of ribs on walls.....	42
Figure 3.8 – Performance of smooth, membrane ribs, and wall ribs.....	42
Figure 3.9 – Performance data of a 1m long counter-flow channel.....	43
Figure 3.10 - Recovered power and fan power for AHRI standard summer operating conditions.	45
Figure 3.11 - Cross-flow ERV configuration.....	46
Figure 3.12– Cross flow core simulation domain.....	47
Figure 3.13– Cross flow local Nu and Sh values for various values of Re.....	48
Figure 3.14. - Variation of the effectiveness of enthalpy exchangers with smooth and ribbed channels for three different operating flowrates.. ..	50
Figure 3.15. - Total recovered power of enthalpy exchangers with smooth and ribbed channels for three different operating flowrates in AHRI Summer (cooling) operating conditions.	50
Figure 3.16 –3D-Printed Flow channel insert.....	52

Figure 3.17- Rib features formed into the membrane..... 53

Figure 3.18 - Rectangular channel cross-section and dimensions 54

Figure 3.19 – Experimental performance of smooth and ribbed channels with corresponding simulation predictions..... 56

List of Symbols

A = Area

c = Species mass concentration

D = Mass diffusivity

$D_h = 4A/P$ = Hydraulic diameter

f = Friction factor

H = Channel height

J'' = Mass flux per unit area

l = membrane thickness

Nu = Nusselt number

P = membrane permeability

q'' = Heat flux per unit area

Re = Reynolds number

Sh = Sherwood number

T = Temperature

W = Channel width

Y = Humidity mass fraction

λ = Thermal conductivity

Subscripts

$1,2,3,4$ = sweep inlet, sweep outlet, feed inlet, feed outlet

b = bulk fluid property

w = wall property

f = feed stream

s = sweep stream

List of Abbreviations

ERV – Energy Recovery Ventilator

HRV – Heat Recovery Ventilator

UDF – User Defined Function

WRR – Water Recovery Ratio

AR – Aspect Ratio

Acknowledgements

Studying on this project to further my education has been a truly fulfilling experience. Not only did I grow as an engineer and researcher, but also as a person. I would like to thank my supervisors, Dr. Steve Rogak and Dr. Sheldon Green for supervising my studies and providing deep and insightful guidance throughout my time at UBC.

I was fortunate enough to work collaboratively with dPoint Technologies during my studies. They are an exciting and innovative company and were collectively a pleasure to work with. In particular, I would like to thank David Kadylak and Dr. Ryan Huizing for their expert guidance and supervision during my thesis and a summer internship I spent at dPoint.

My fellow graduate students, Amin Engarnevis and Sarah Cosby, were wonderful to work with and made my workdays so much more pleasant. Talking things through with them always lead to solutions to our problems. I am happy to call them my friends.

Last but not least, I would like to thank my family. My parents and brother have always been incredibly supportive of my endeavors and even though we are usually thousands of kilometers apart, our skype conversations never failed to bring a smile to my face. I have also been incredibly lucky to have my loving partner, Kay Penney, by my side for the past 2 years. She moved across the continent, and into a new country, so we could stay together during my study and none of this would have been possible without her.

Chapter 1: Introduction

1.1 Energy Recovery in Air Conditioning

1.1.1 Building Ventilation

Air conditioning accounts for roughly 50% of the energy consumption in the building sector in North America [1], [2]. It is required in order to maintain occupant comfort and health in buildings, which is increasingly important as people spend more of their time indoors. Figure 1.1 shows the thermal comfort zone on a psychrometric chart for winter (1.0 clo) and summer (0.5 clo) clothing levels [3]. It is the goal of the air conditioning system to maintain conditions within the comfort zone. This is complicated by the need for fresh air supply to maintain air quality. Indeed, minimum flow rates are required by modern standards [4]. Therefore, outdoor air must be continually introduced into the building and conditioned before it is exposed to occupants, making for a tradeoff between energy expenditure, thermal comfort, and air quality.

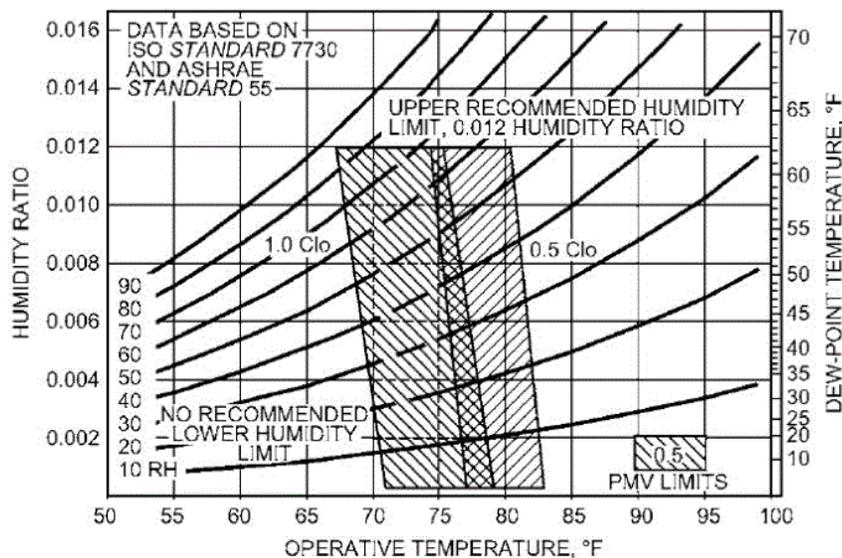


Figure 1.1 - Typical thermal comfort zone on a psychrometric chart [3]

A significant component of air conditioning is controlling the temperature and humidity. This is especially important in hot and humid climates where it accounts for 20-40% of the air conditioning load, and even more when 100% fresh air is required (eg. Hospitals) [5, pp. 1–2]. Humidity can be particularly energy-intensive to control due to the high heat of vaporization of water vapour. Further, the method often used to control humidity, in which air is dehumidified by cooling it to below its dew point and the re-heating it to comfortable levels, is particularly energy intensive [5, pp. 5–6].

Because of the importance of adequate air conditioning, its high energy cost, and the need to continually bring in air from the outdoors, it is desirable to recover energy from exhaust air that has already been conditioned. One popular technology for this is the energy recovery ventilator (ERV).

1.1.2 Energy Recovery Ventilators

Energy recovery ventilators (ERVs) provide a way to recover sensible and latent heat from exhaust air in ventilation systems. This reduces energy demand since intake air is partially conditioned by exhaust air. They differ from heat recovery ventilators (HRVs) which recover only sensible heat. ERVs work by passing exhaust and intake air through adjacent flow passages separated by a thermally conductive and porous material. This material must strike a balance between permeability to heat/moisture and impermeability to air-borne contaminants which would otherwise be recirculated back into the building, shown in Figure 1.2.

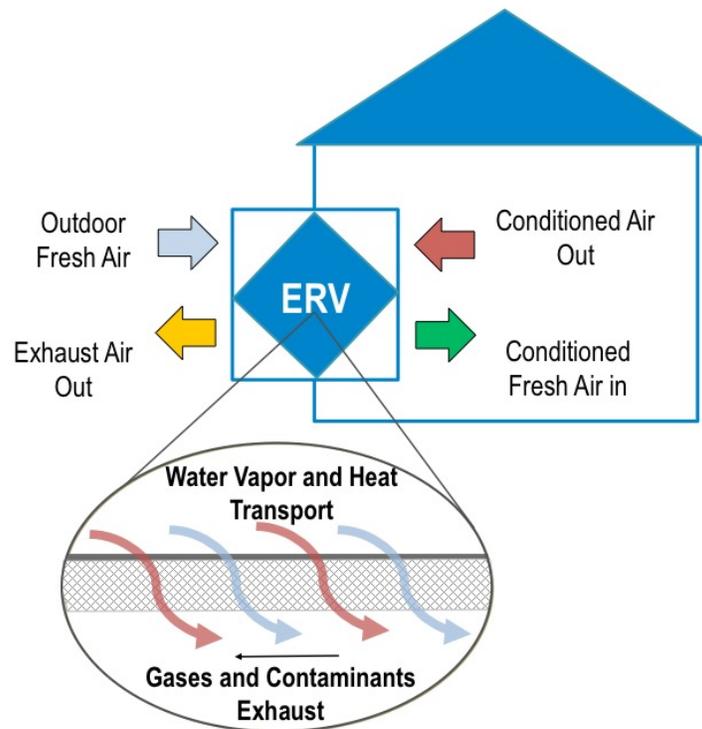


Figure 1.2 – Function of an ERV

Particularly promising materials for ERVs are polymeric membranes with high water-vapour permeability and selectivity of water-vapour over contaminants, such as those used by dPoint Technologies [6]. They are anisotropic membranes made up of a porous substrate for structural support ($\sim 100\mu\text{m}$) and a dense selective coating ($0.1\text{-}10\mu\text{m}$), shown in Figure 1.3 [7]. The coating is made sufficiently thin to maintain high water-vapour permeability. The overall thickness of the membrane is small enough so that its sensible heat transfer resistance is negligible. Overall, the membrane water vapour transfer resistance is several orders of magnitude larger than the thermal resistance, making the behavior of the two processes different [8, pp. 12–14].

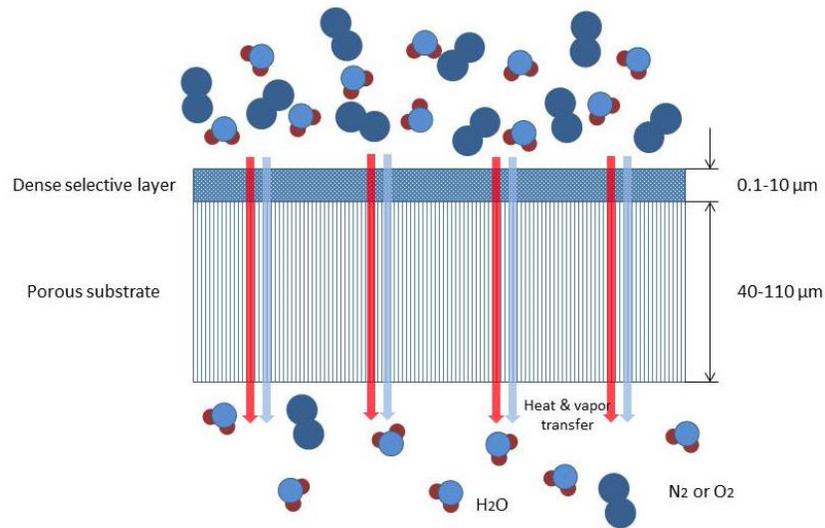


Figure 1.3 – Composite polymer membrane construction [7].

ERVs are commonly made as parallel plate exchangers in either a cross- or counter-flow arrangement [8, pp. 93–120]. They are made by stacking multiple layers of membrane material separated by spacers. Both cross- and counter-flow arrangements are shown in Figure 1.4. Cross-flow cores are easier to fabricate, but have lower effectiveness [9, pp. 336–347]. When selecting a core for an application, there is a tradeoff between footprint, cost, and effectiveness. Traditionally, recovery effectiveness targets are met by varying the channel height, width and length. However, this requires more material and increases the ERV footprint. Another method of improving effectiveness is by using more permeable material, however highly-permeable and selective materials are typically costly.

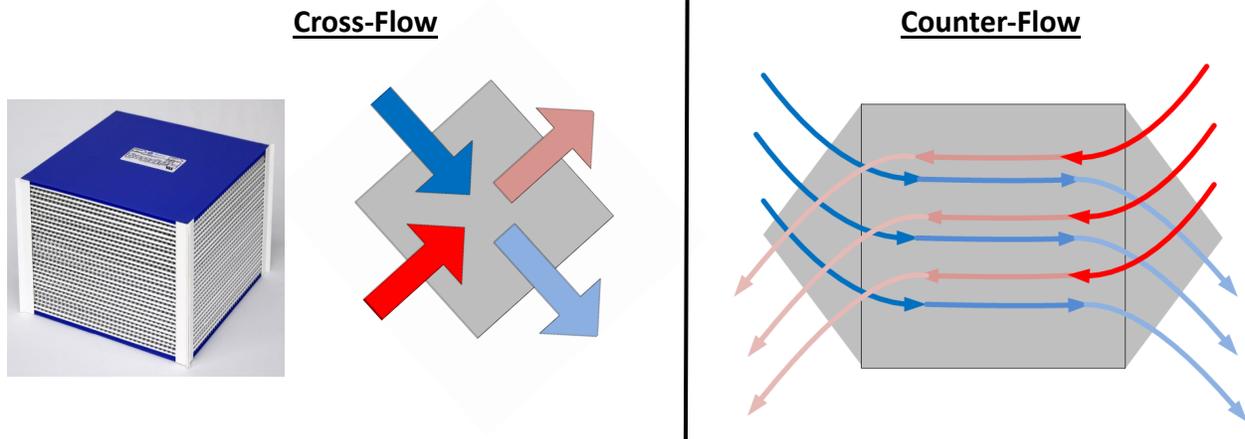


Figure 1.4 – ERV Construction. Cross-flow (left) and counter-flow (right).

An alternative method of increasing performance is through geometric mixing features within the channels themselves. As air moves along a smooth channel, the temperature and moisture concentration fields become less homogeneous, which reduces the effectiveness of the exchanger [8, pp. 120–123]. Certain variations to the flow passage shape can promote mixing and reduce the heterogeneity of the temperature and moisture distributions. This will be discussed in more detail in section 1.2.5.

1.1.3 Other Applications

While ERVs are the main focus of this work, heat and mass exchangers have other important applications which could also benefit from the ideas discussed here. One other application is for PEM fuel cells. Fuel cells require a controlled level of moisture at the inlet even though the fuel cell reactions produce water [10]. Membrane-based fuel cell humidifiers offer a method of recovering the produced moisture and using it for inlet humidification. They are similar to ERVs, though the temperature and humidity is typically higher and the footprint must be smaller [11]. A commercial cross-flow fuel cell humidifier is shown in Figure 1.5.

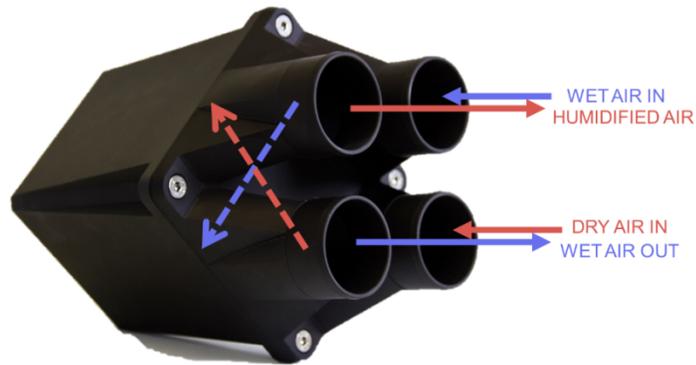


Figure 1.5 - dPoint fuel cell humidifier

Other applications include gas separation with specialized membranes that are selective to a particular species of interest. These are typically large-scale industrial processes used to obtain a pure gas from a mixture. Examples include separating nitrogen from air, hydrogen from ammonia plant purge-gas streams, carbon dioxide from methane in natural gas, and dehydration of air [12, pp. 325–326].

1.2 Theory

In order to maintain generality in the current study, the fluid in the simulations is considered Newtonian with constant properties. This means that properties such as density and thermal conductivity do not change with pressure, temperature, or humidity.

1.2.1 Heat and Mass Transport

Under steady state conditions, the governing equations of the fluid are [9]

Mass

$$\nabla \cdot \vec{v} = 0 \quad (1)$$

Momentum

$$\rho \vec{v} \cdot \nabla \vec{v} = \nabla^2 \vec{v} - \nabla p \quad (2)$$

Energy (Assuming viscous dissipation is negligible)

$$\rho c_p \vec{v} \cdot \nabla T = \lambda \nabla^2 T \quad (3)$$

Humidity

$$\vec{v} \cdot \nabla Y = D \nabla^2 Y \quad (4)$$

Where ρ is the density, c_p is the specific heat, λ is the thermal conductivity, $Y = c/\rho$ is the mass fraction of water vapour (kg water/ kg mixture), and D is the diffusivity of water vapour in air.

The behavior of the flow can be described by the Reynolds Number, which is the ratio of inertial to viscous forces.

$$Re = \frac{\rho v D_h}{\mu} \quad (5)$$

Where μ is the fluid viscosity and $D_h = 4A/P$ is the hydraulic diameter of the flow passage, where A and P are the cross-section area and perimeter, respectively. The thermal behavior is

described by the Prandtl Number, which is the ratio of momentum diffusivity to thermal diffusivity

$$Pr = \frac{\mu c_p}{\lambda} \quad (6)$$

Similarly, the Schmidt Number is the ratio of momentum diffusivity to mass diffusivity

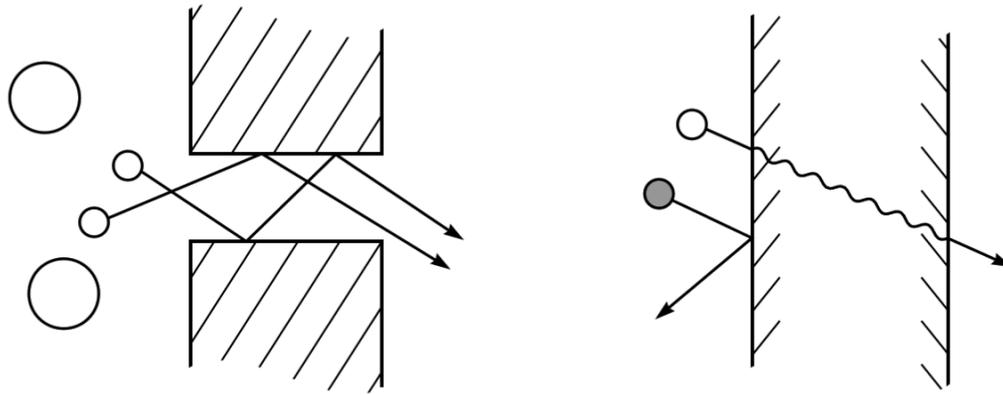
$$Sc = \frac{\mu}{\rho D} \quad (7)$$

For this study, $Pr = 0.7$ and $Sc = 0.6$, which are typical values for air. The fact that $Pr > Sc$ means that thermal energy diffuses through the fluid more quickly than mass (humidity).

1.2.2 Membrane Transport

As mentioned in section 1.1.2, anisotropic polymer membranes are effective ERV materials possessing high water vapour permeability and selectivity over contaminants. As shown in Figure 1.3, they are made up of a porous substrate, used for structural support, with dense coating on one side that provides selectivity of water vapour over contaminants.

Transport theory in these two media has been described by Baker [12, pp. 15–92]. The porous substrate has permanent pores through which fluid flows driven by a pressure gradient through the material, known as the pore-flow model. Some molecules are filtered out due to their size, but this effect is not significant in ERV membranes. The dense coating, on the other hand, has no well-defined pores. Instead, species dissolve into the membrane material and diffuse through it driven by a concentration gradient, known as the solution-diffusion model. The inability of some molecules to dissolve into the dense coating is the source of membrane selectivity in ERVs.



Pore-flow model: molecules flow through permeant pores. Some are filtered out due to their size.

Solution-diffusion model: molecules dissolve into the membrane material. Some are separated out due to an inability to dissolve.

Figure 1.6 – Membrane permeation/separation modes [12, p. 16].

The flux of species i according to the pore-flow model is

$$J_i = -K'c_i \frac{dp}{dx} \quad (8)$$

Where c_i is the concentration of species i , and K' is a constant depending on the membrane material and fluid. The flux for the solution-diffusion model is

$$J_i = -D_i \frac{dc_i}{dx} \quad (9)$$

Where D_i is the diffusion coefficient, which describes the ability of the species to adsorb to the membrane surface, diffuse through it, and desorb from the other side. For composite membranes, a similar relation is used

$$J''_i = -P \frac{dc_i}{dx} = -P \frac{c_i^{(s)} - c_i^{(f)}}{l} \quad (10)$$

Where (s) and (f) refer to the sweep and feed sides of the membrane, respectively, l is the membrane thickness, and P is the membrane permeability, which is typically measured experimentally and depends on the membrane material, temperature, and humidity.

Sensible heat moves through the membrane according to the conduction equation

$$Q'' = -\lambda_m \frac{dT}{dx} = -\lambda_m \frac{T^{(s)} - T^{(f)}}{l} \quad (11)$$

The membrane is more permeable to heat than to mass, this can be quantitatively seen through the ratios of membrane flux to air-side diffusion to given by

$$\tau = \frac{P/l}{D/H} \quad (12)$$

$$\sigma = \frac{\lambda_m/l}{\lambda/H} \quad (13)$$

These values depend on the membrane material but, typically, $\tau \sim 1$, so the membrane mass transfer resistance is the same order of magnitude as the air-side diffusion resistance. On the hand, $\sigma \sim 100$, so the membrane heat transfer resistance is negligible compared to air-side diffusion [6], [13], [14].

1.2.3 Performance Metrics

Heat transfer performance can be characterized by the Nusselt number, which is the ratio of convective to conductive heat transfer

$$Nu_z = \frac{D_h h}{\lambda} \quad (14)$$

Where D_h is the hydraulic diameter of the channel and h is the heat transfer coefficient, defined for the air-side convection transfer as

$$h = \frac{q''}{(T_w - T_b)} \quad (15)$$

Where q'' and T_w are averaged over the periphery of the membrane. One can also define the overall heat transfer coefficient between the two channels as

$$H = \frac{q''}{(T_b^{(f)} - T_b^{(s)})} \quad (16)$$

In this form the heat transfer coefficient accounts for the air-side resistance of both channels as well as the membrane resistance. One can analogously define the local Sherwood number, which gives the ratio of convective to diffusive species (e.g. moisture) transfer

$$Sh_z = \frac{D_h k}{D} \quad (17)$$

$$k = \frac{J''}{(c_w - c_b)} \quad (18)$$

$$K = \frac{J''}{(c_b^{(f)} - c_b^{(s)})} \quad (19)$$

The average Nusselt and Sherwood numbers are defined as

$$Nu = \frac{1}{L} \int_0^L Nu_z dz \quad (20)$$

$$Sh = \frac{1}{L} \int_0^L Sh_z dz \quad (21)$$

Where the integrals are taken over the length of the channel. The overall recovery efficiency of the exchanger is given by

$$\eta_T = \frac{T_o^{(s)} - T_i^{(s)}}{T_i^{(f)} - T_i^{(s)}} \quad (22)$$

$$\eta_c = \frac{Y_o^{(s)} - Y_i^{(s)}}{Y_i^{(f)} - Y_i^{(s)}} \quad (23)$$

Because the fluid has (by assumption) constant properties with equal flow rates on each side of the counter-flow exchanger, the bulk temperature and humidity difference is constant along the channel. This gives simplified equations for the LMTD method [9]

$$q = HA\Delta T \quad (24)$$

$$J = KA\rho\Delta Y \quad (25)$$

Where A is the smooth-channel membrane area in a periodic section. Combining these with the definitions of efficiency, it can be shown that

$$\eta_T = \frac{1}{1 + \frac{mc}{HA_T}} \quad (26)$$

$$\eta_c = \frac{1}{1 + \frac{\dot{m}}{\rho KA_T}} \quad (27)$$

Where A_T is the total smooth-channel membrane area over the entire channel length. The heat and mass transfer coefficients can be calculated in a single periodic section of the duct and used with the above equations to predict the effectiveness for the entire duct. The benefit of increasing heat and mass transfer must be weighed against a consequential increase in frictional losses. This can be expressed through the friction factor, defined locally as

$$f_z = \frac{1}{D_h} \frac{dp/dz}{\rho u} \quad (28)$$

and globally as

$$f = \frac{1}{L} \int_0^L f_z dz \quad (29)$$

The power consumed to force air through the duct is then

$$P_{fan} = Q\Delta p\eta_{fan} = Q(fLD_h\rho u)\eta_{fan} \quad (30)$$

Where Q is the volumetric flow rate through the duct and $\eta_{fan} < 1$ is the fan efficiency.

1.2.4 Periodic Boundary Conditions

The hydrodynamically developing region is an important aspect of ERV performance. It is characterized by the section of channel downstream of the entrance where the flow fields change from homogeneous to a ‘developed’ distribution defined by the channel boundary conditions. Figure 1.7 shows a developing velocity profile in a 2D duct, in which the velocity field changes from uniform to a parabola due to the no-slip ($\vec{v} = 0$) boundary condition at the walls. Similar developing profiles exist for the temperature and humidity fields. This entrance region is characterized by high values of Nu , Sh , and f values that decrease asymptotically to lower developed values. This region can take up a large fraction of the channel length and account for a significant portion of the ERV performance [8, pp. 120–124].

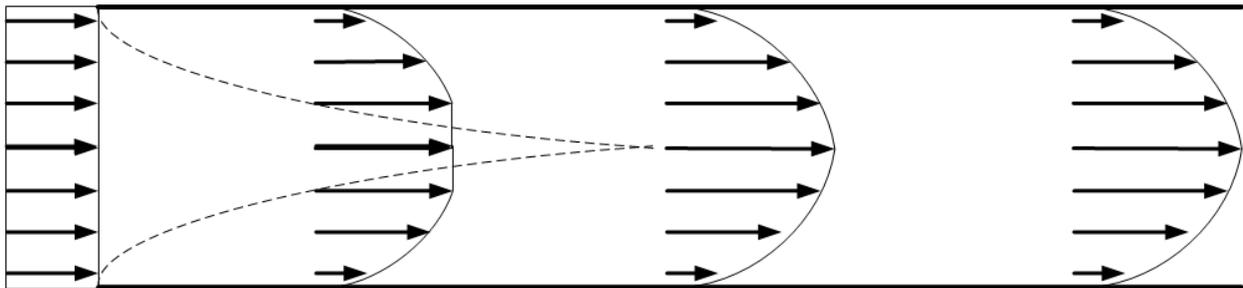


Figure 1.7 - Hydrodynamically developing velocity profile

Unfortunately, the developing region is so long that it can incur considerable computational expense to simulate. For this reason, and in order to highlight the effects of the mixing features, the channel entrance region is ignored in the current study. This means that the effectiveness predictions are lower than they would be in reality. However, because the developing region

would increase the effectiveness of both ribbed and smooth channels, the differences in performance between the two would be roughly the same. Future work could be done to investigate the magnitude of this effect in different circumstances.

The entrance region is ignored by assuming periodic boundary conditions between the inlet and outlet. This is typically only valid when the boundary conditions are uniform temperature (concentration) or uniform wall heat (mass) flux. While this case is neither, the total heat (mass) flux is the same for each periodic section, so the analysis of Patankar et al. can be applied [15].

They found the velocity field and the dimensionless quantities

$$\theta = \frac{T - T_w}{T_b - T_w} \quad (31)$$

$$\zeta = \frac{Y - Y_w}{Y_b - Y_w} \quad (32)$$

repeat between the inlet and outlet. Where the wall values are averages over the cross-section periphery on the membrane. This gives the inlet conditions as

$$T^{(i)} = T_w^{(i)} + (T_b^{(i)} - T_w^{(i)}) \frac{T^{(o)} - T_w^{(o)}}{T_b^{(o)} - T_w^{(o)}} \quad (33)$$

$$Y^{(i)} = Y_w^{(i)} + (Y_b^{(i)} - Y_w^{(i)}) \frac{Y^{(o)} - Y_w^{(o)}}{Y_b^{(o)} - Y_w^{(o)}} \quad (34)$$

$$\vec{v}^{(i)} = \vec{v}^{(o)} \quad (35)$$

Where $T_b^{(i)}$ and $Y_b^{(i)}$ and the mass flow rate are simulation inputs and the other terms come from the previous simulation iteration. Each simulation was initialized with uniform velocity,

temperature, and humidity profiles, and then the above equations were used after each iteration until simulation convergence.

1.2.5 Geometric Enhancement

As discussed in section 1.1.2, geometric features that promote mixing can be an attractive method of increasing ERV performance without requiring additional materials or increase the core footprint. Others have demonstrated significant performance boosts through such variations. Zhong et al. found that adding air deflectors and spreader plates to a cross-flow ERV can improve sensible and latent effectiveness by up to 17.4% and 7.8%, respectively [16]. Koester et al. investigated the performance of an ERV with netting installed between membrane layers. They found the new configuration had higher effectiveness and pressure drop, which overall could result in significant energy savings depending on the cost of energy and climate conditions [17]. While these studies demonstrate that certain mixing features can improve overall core performance, they did not investigate the local performance within the individual channels. A goal of the present study was to investigate how mixing features can alter local flow fields improve individual channel performance.

Recovery effectiveness between a pair of channels can be described in terms of a resistance in series model. Within each channel there is an air-side resistance due to the time it takes for heat/mass to diffuse to the membrane surface. The membrane itself also has a resistance as the heat/mass must diffuse through the membrane. Finally, there is another air-side resistance in the opposite channel as the heat/mass must diffuse away from the membrane surface and move along the channel.

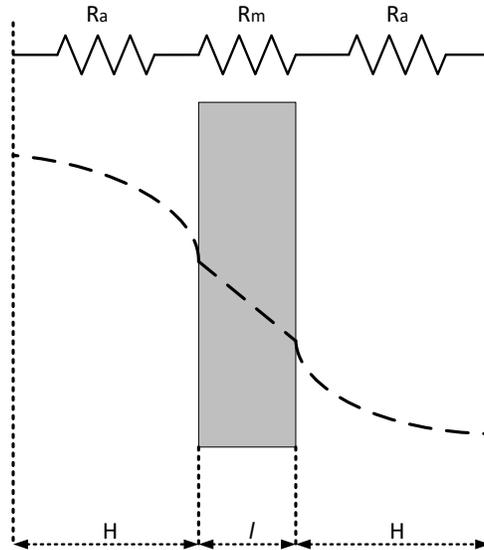


Figure 1.8 - Resistance in series model

These resistances add together to oppose the flux from one channel to another, much the same as electrical resistors oppose the flow of current in a circuit. It is the work of material scientists to try and create the most permeable membrane (ie. lowest resistance) without sacrificing selectivity of the membrane. This work, on the other hand, aims to improve performance by lowering the air-side resistance. This is done by mixing the flow in such a way to reduce the distance for diffusion towards the membrane.

Many different mixing features have been investigated to enhance sensible heat transfer in ducts. Figure 1.9 shows some of the most popular features: delta winglets [18], [19], wire-coil inserts [20], and angled ribs. A summary of studies investigating angled rib features is in Table 1.1.

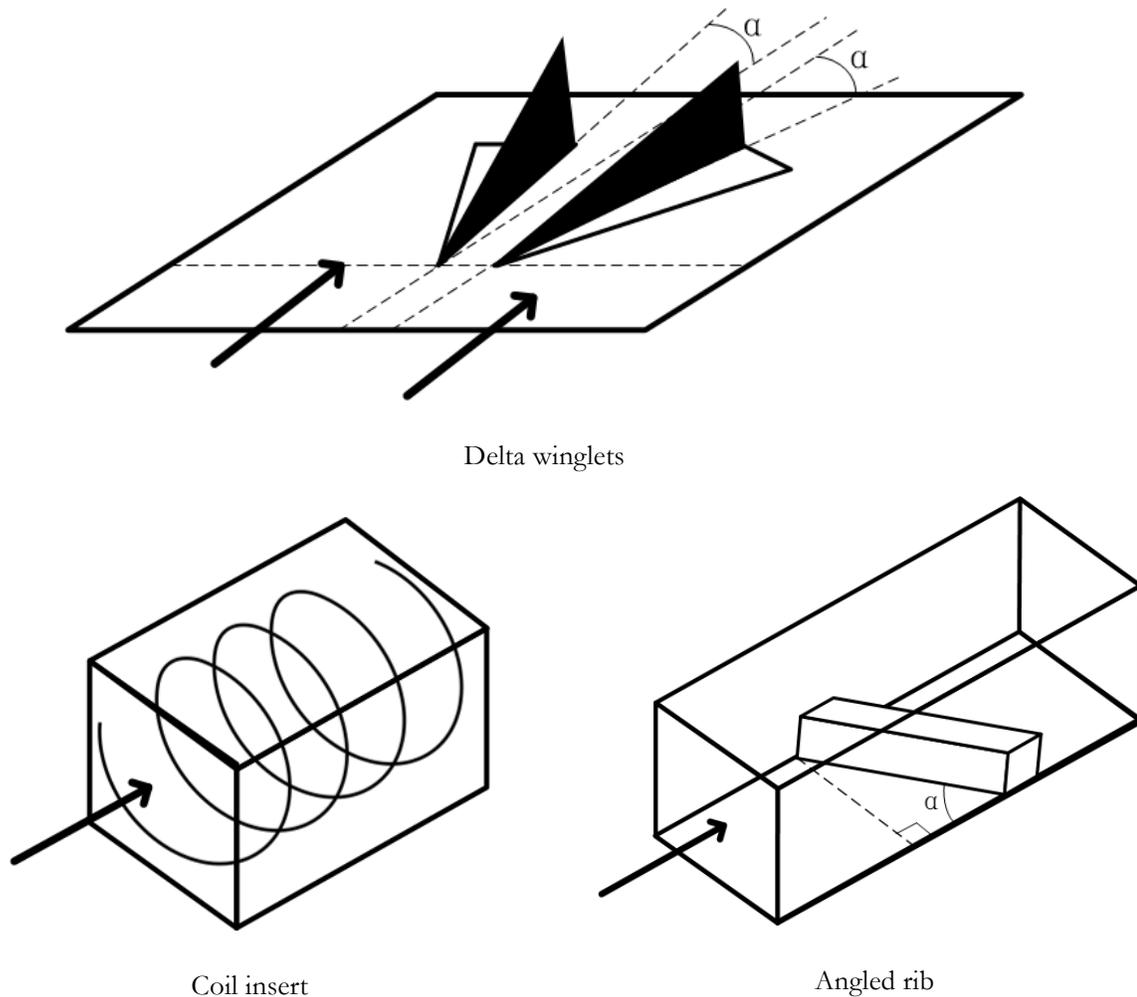


Figure 1.9 - Common enhancement geometries

Table 1.1 Previous studies investigating heat transfer enhancement in ducts with angled ribs

Study	Regime	Description
Zheng et al. (2016) - Numerical investigations of the thermal-hydraulic performance in a rib-grooved heat exchanger tube based on entropy generation analysis	Turbulent	Studied a circular tube with discrete ribs covering the inside surface and constant wall temperature. Nusselt number increased by a factor 1.58-2.46 while friction factor increased by 1.82-5.03. This resulted in an overall increase in thermal performance of 1.19-1.68.

Olsson and Sundén (1997) - Experimental study of flow and heat transfer in rib-roughened rectangular channels [21]	Laminar and Turbulent	Experimental study of ribs on opposite sides of a rectangular channel at constant wall temperature. Looked at parallel ribs, cross ribs, cross V-ribs, parallel V-ribs, and multiple V-ribs. They found all cases increased the heat transfer coefficient for a given pumping power per heat transfer area.
Tanda and Abram (2009) - Forced convection and heat transfer in channels with rib turbulators inclined at 45 deg [22]	Turbulent	Experimental study of angled ribs through a rectangular channel with assumed constant wall heat flux. Nusselt number increase by up to 50% for fixed pumping power.
Kim et al. (2009) - Optimal design of angled rib turbulators in a cooling channel [23]	Turbulent	Numerical study of a square channel with ribs on opposite walls and constant wall temperature. Improved thermal performance by up to 60% by optimizing rib angle of attack and height/spacing ratio
Nonino and Comini (2002) – Convective heat transfer in ribbed, square channels [24]	Laminar	Numerical study of a square channel with angled ribs and uniform wall temperature on surfaces. Angled ribs increase Nusselt number by up to 20% more than the corresponding increase in friction factor
Saha (2010) - Thermal and friction characteristics of laminar flow through rectangular and square ducts with transvers ribs and wire coil inserts [20]	Laminar	Experimental study combining angled ribs and coil inserts in rectangular and circular channels. Found that, for equal pumping power, the combination of ribs and wires can increase heat recovery by up to 50% compared to ribs or wires alone.

These studies all observe a significant heat transfer enhancement with ribs and, provided the various design parameters are tuned correctly, the increased heat transfer outweighs the corresponding increase in frictional pressure drop. This makes angled ribs an attractive feature for ERVs. However, the data from these studies cannot be applied to ERVs directly because, for one, the analysis was restricted to either constant wall temperature or constant wall heat flux boundary conditions. The boundary conditions in an ERV are neither constant wall temperature

(concentration) nor constant wall heat (mass) flux. Rather, they are defined by the coupling between adjacent flow passages through the membrane.

Another issue with applying this data to ERVs is that mass transfer was not investigated. Because the fundamental equations of heat and mass transfer are very similar, it is often possible to convert heat transfer data to mass transfer data with the empirical Chilton-Colburn j-factor analogy, which states

$$\frac{h}{\rho c_p} (Pr)^{1/3} = k(Sc)^{1/3} \quad (36)$$

Unfortunately, in recirculating flows, such as those introduced by internal channel ribs, the Chilton-Colburn j-factor analogy breaks down [8, p. 16], [9, pp. 538–539]. Therefore, a full counter-flow exchanger model considering both heat and mass transfer is required.

1.3 Current Study Objectives

This study investigates full conjugate heat and mass transfer in an ERV with angled ribs in the channels. The model geometry consists of two rectangular ducts in a counter-flow arrangement, separated by a porous polymer membrane. In order to compare the effect of ribs to simply changing the channel dimensions or flow rate, channels are considered for various aspect ratios and flow rates, with and without ribs. The ribs are placed on the membrane surface and are themselves permeable to heat and moisture.

Chapter 2: Methods

Instead of investigating the performance of a full ERV directly, this study investigates the performance of a relatively short pair of channels in a counter-flow arrangement. This is done to reduce computational and experimental complexity. The results from this short section are then used to estimate the performance of full length channels as well as a cross-flow ERV. Validation experiments are performed to investigate the mass transfer and pressure drop predictions of the simulations.

2.1 Simulation

2.1.1 Problem Description

The geometry of the counter-flow ERV model is shown in Figure 2.1. Two square ducts are in a counter-flow arrangement separated by a polymer membrane that is permeable to water vapour. All other walls are adiabatic. Feed air (building exhaust) enters at point 1 and leaves through point 2. Sweep air (building intake) enters point 3 and leaves through point 4. The inlets at 1 and 3 are modeled as mass-flow inlets (wherein the inlet pressure is varied to give a desired mass flow) and outlets 2 and 4 are pressure outlets at a gauge pressure of zero. Periodic boundary conditions are applied between the inlet and outlet of each duct as described in Section 1.2.4. The user define function (UDF) for implementing the periodic boundaries is included in Appendix A.2.

The channel width is constant for all cases. The height is varied to give an aspect ratio (height/width) of 0.6, 1, or 1.4. Simulations are performed with and without ribs with ribs

spaced 16 channel widths apart. The ribs themselves are $1/3$ channel height tall and $1/3$ of the channel width wide.

The fluid is considered Newtonian with constant properties ($Pr = 0.7$ and $Sc = 0.6$). Three different flow rates are considered, yielding a Reynolds number in the range of $200 \leq Re \leq 900$, which is typical for a commercial-scale ERV. This range of Re was studied for a similar geometry [24] and found to be in the laminar regime, so laminar flow is assumed for all cases in this study.

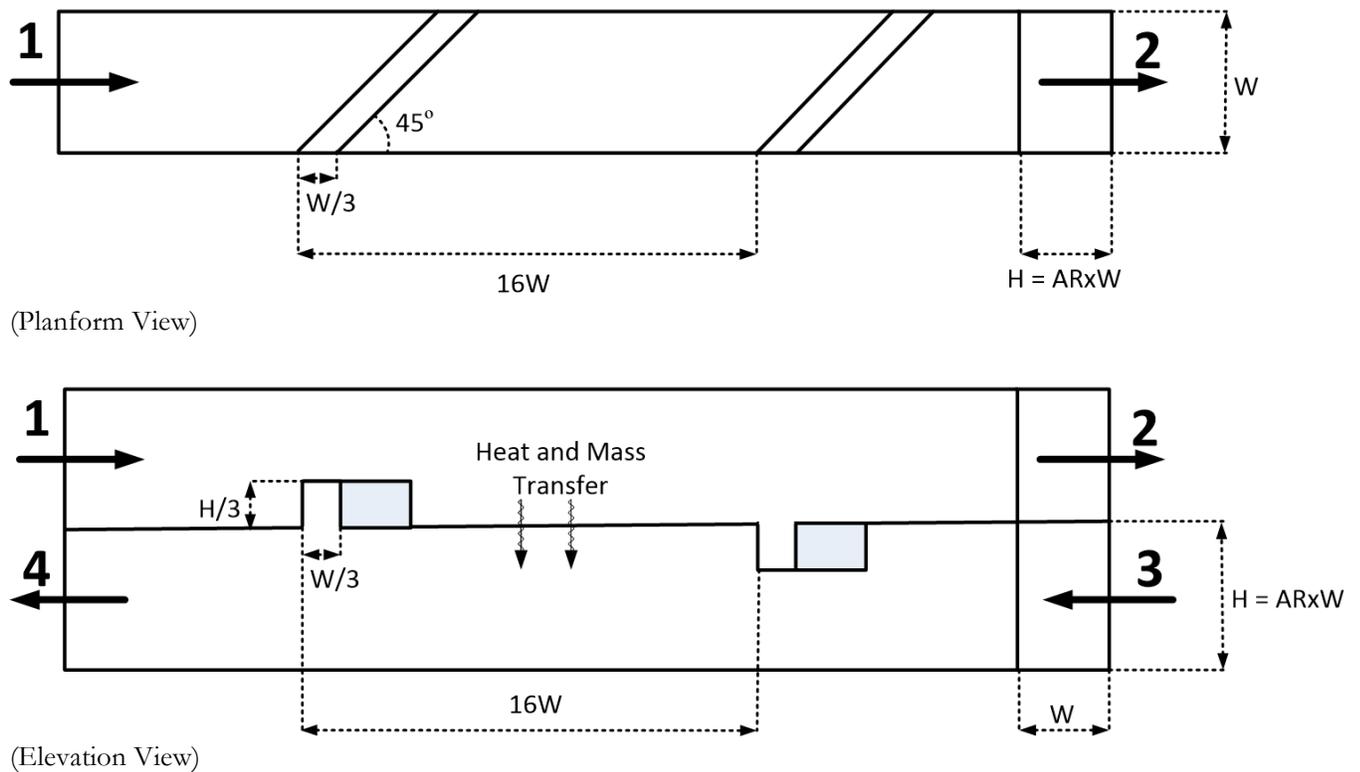


Figure 2.1 – Schematic of the simulation domain

2.1.2 Simulation Parameters

Simulations were run on the finite-volume solver, Fluent (version 16.1). Meshing was done on Ansys Meshing (version 16.1). The flow was solved in the 3d laminar, pressure-based solver with a 2nd order upwind discretization scheme. SIMPLE pressure-velocity coupling was used with a least squares cell-based gradient calculation.

2.1.3 Membrane Modelling Approach

Fluent has built in functionality to implement equation (11) for heat transfer across the membrane, however, there is no such functionality to directly implement equation (10) for mass transfer. Instead, a custom User-Defined Function (UDF) must be written to implement equation (10). Among the many uses of UDFs is they allow the user to script custom profiles and source terms over mesh elements during the simulation [25]. Previous studies [26]–[28] have used UDFs to define custom sink/source terms on the mesh elements adjacent to the membrane surface of the form

$$\frac{dm_i}{dt} = -P \frac{c_{w+\Delta}^{(s)} - c_{w+\Delta}^{(f)}}{l} \frac{V_{cell}}{A_{cell}} \quad (37)$$

Where $c_{w+\Delta}$ is the humidity in the cell adjacent to the membrane surface and V_{cell} and A_{cell} are the volume and area of the cell on which the source term is implemented. The drawback of this method is that when attempting to implement it in this study, it was found to depend heavily on the size of mesh elements adjacent to the membrane surface.

Instead, an alternate method was used in which the membrane surface concentration was varied in order to get a desired concentration gradient. This method has been used in at least one other

study [29] and can be described as follows. Mass-flux through the membrane into the sweep stream can be described by equation (10) and the air-side flux into the sweep stream is given by

$$J'' = -D\nabla c^{(s)} = -D(c_w^{(s)} - c_{w+\Delta}^{(s)})/\Delta y \quad (38)$$

Where D is the diffusivity of water-vapour in air. Rearranging the above equation gives the membrane surface concentration on the sweep side as

$$c_w^{(s)} = c_{w+\Delta}^{(s)} + \frac{P/l}{D/\Delta y} (c_w^{(s)} - c_w^{(f)}) \quad (39)$$

After each simulation iteration, the membrane surface concentration was varied according to equation (39) to produce the membrane flux predicted by equation (10). The terms on the right hand side of equation (39) were the data from the previous iteration. The UDF for implementing this is included in Appendix A.1.

2.1.4 Convergence Criteria

Simulations were considered converged after all residuals dropped below 1e-6 and the sweep side outlet humidity mass fraction, outlet temperature, and inlet pressure remained unchanged to the 5th significant figure for 25 iterations. For smooth channels this was achieved within 1000 iterations. For ribbed channels up to 3000 iterations were required.

2.1.5 Meshing and Mesh Validation

Meshing was performed in Ansys Meshing (v16.1). The mesh is shown in and Figure 2.3. It is a pure hexahedral structured mesh with a higher concentration of cells near the walls, membrane and ribs. In order to maintain a structured mesh with refinement near the ribs, block meshing was used with separate blocks around the ribs. Non-conformal interfaces were used at the interfaces

between the rib blocks and the rest of the mesh. The mesh for the smooth channel (no ribs) was a grid of $28 \times 28 \times 180$ (H×W×L) mesh elements with finer elements near the walls. With this mesh size a typical simulation run would take roughly six hours to complete.

Richardson extrapolation was used to find an error estimate on the final result as described in [30]. In short, the numerical solution data is assumed to have a series expansion in the grid spacing, h

$$f = f[\text{exact}] + g_1 h + g_2 h^2 + \dots \quad (40)$$

The error estimator then takes the form

$$E = \frac{f_2 - f_1}{1 - r^p} \quad (41)$$

Where $r = h_2/h_1 > 1$, p is the order of accuracy of the algorithm, and f_i is the solution obtained for from using grid spacing h_i . Usually the value of p is not known a priori, however it can be calculated by using three successive mesh refinements with equal values of r . The order of convergence is then

$$p = \ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) / \ln(r) \quad (42)$$

This method was applied to the current work, with three different meshes where the grid spacing was successively reduced by 20% (ie. $r = 1.25$). The resulting variation in Nu , Sh , and fRe is shown in Figure 2.4. Equations (41) and (42) were used to find the error estimator for each value. The error estimator is usually multiplied by a safety factor, which, for high quality structured

meshes such as this one, can be taken as 1.25. This gave a final error estimate of 0.5% for Sh , 1% for Nu , and 2% for f .

As a further validation, the smooth channel simulations were run with the membrane replaced with a constant temperature boundary condition. This gave $Nu = 2.414$ and $fRe = 14.168$, which agree within 1% of the values of $Nu = 2.437$ and $fRe = 14.227$ reported by [31].

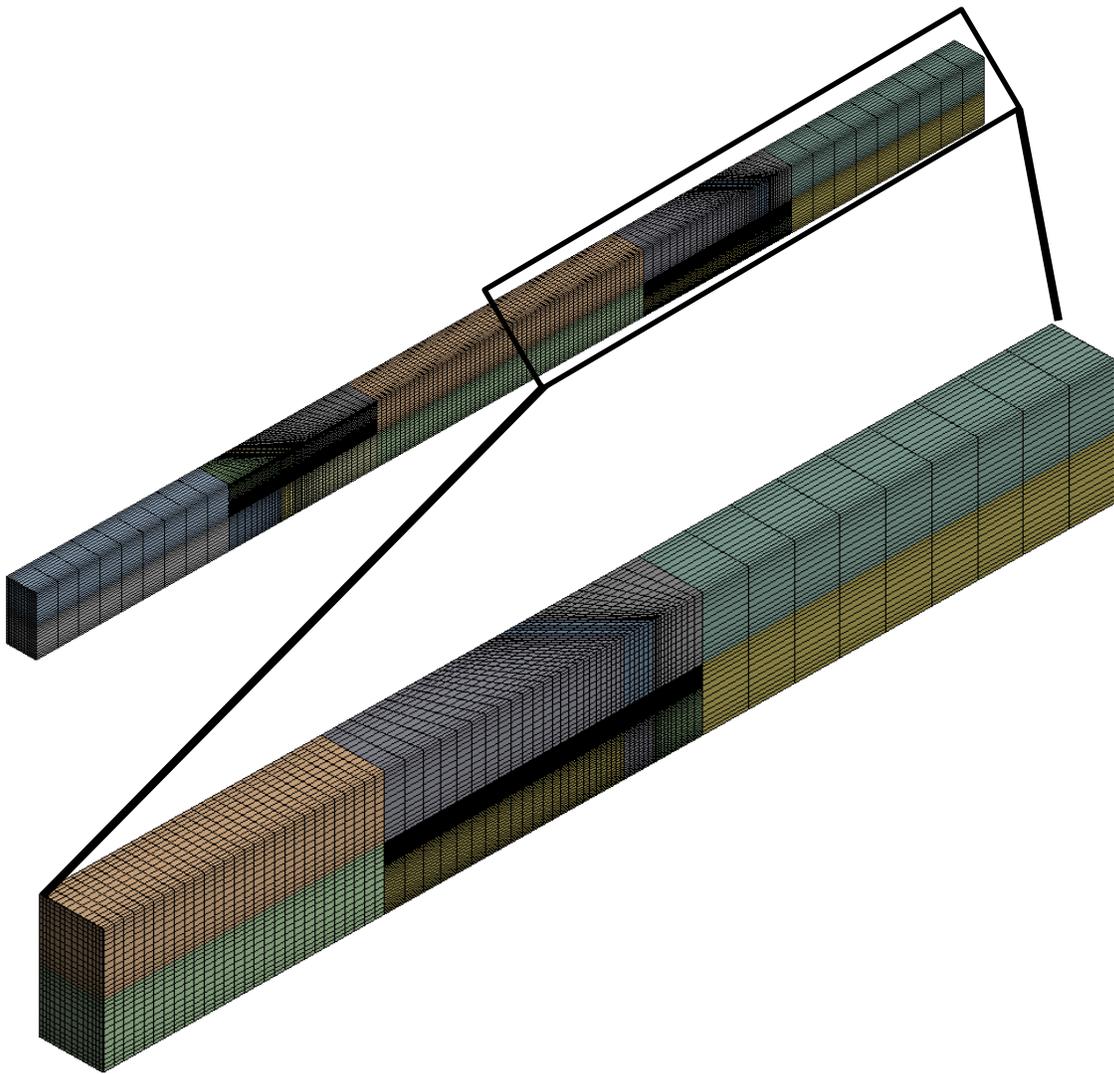


Figure 2.2 – Isometric view of the mesh, mesh spacing has been doubled in each dimension for clarity

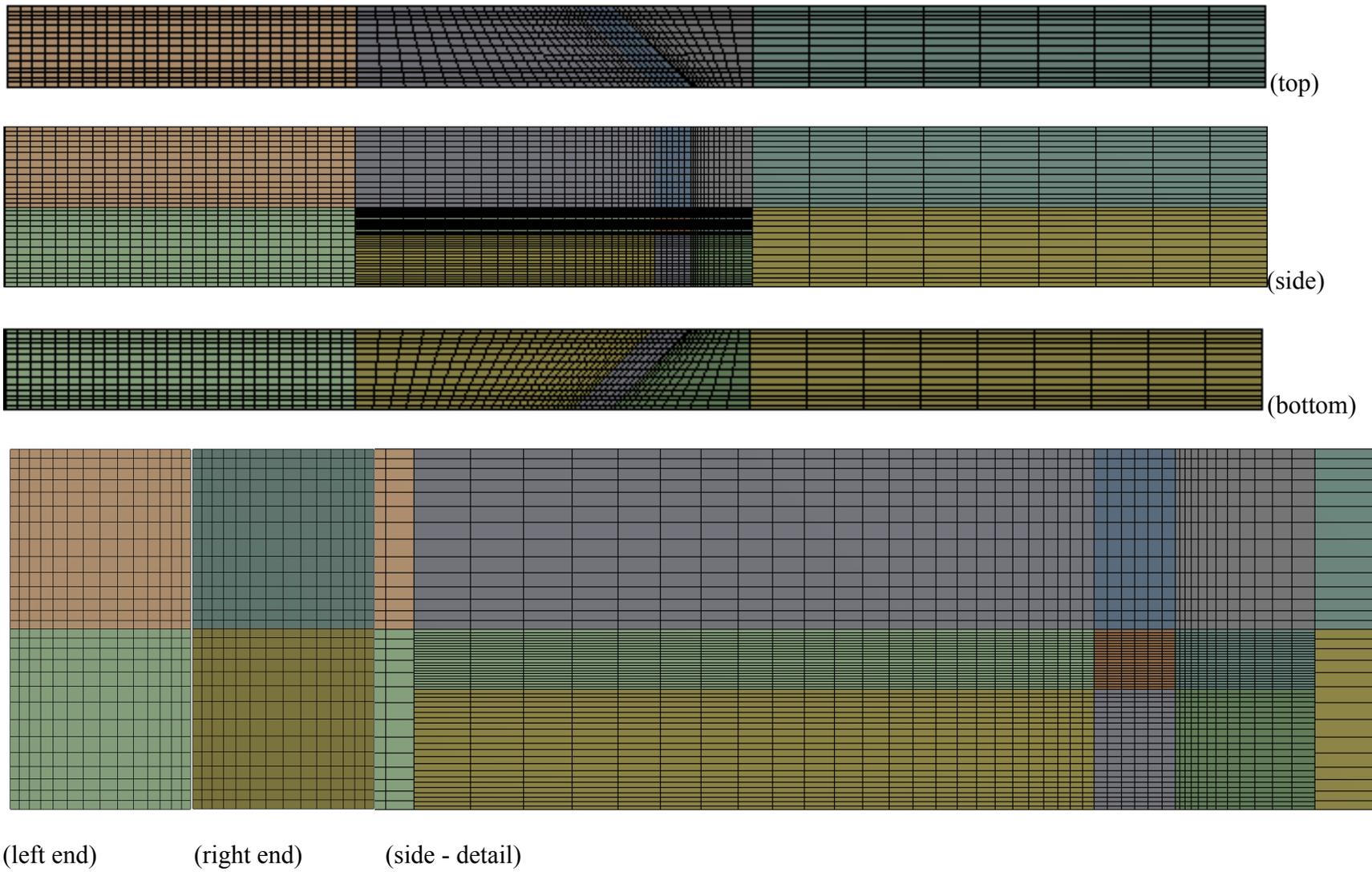


Figure 2.3 – Various views of the mesh. Mesh spacing has been doubled in each dimension for clarity

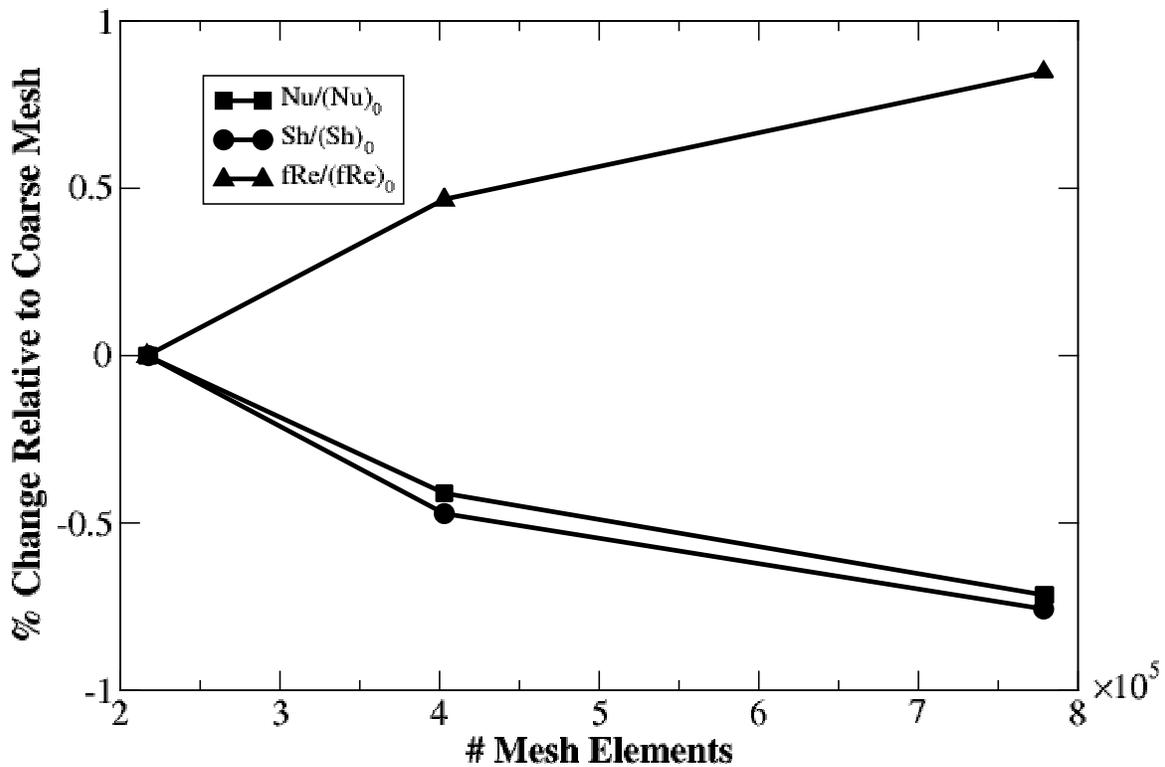


Figure 2.4 – Mesh sensitivity study

2.1.6 Post Processing

Post-processing was required to extract local air-side Nu , Sh , and f values. This was accomplished with a UDF, which is included in Appendix A.1. In brief, the simulation domain was divided up into small, equal-sized, sections along the length of the channel (20 for Nu and Sh , 60 for f). Average bulk and wall values were calculated on each section and used to calculate the local coefficients according to equations (15), (18) and (28). The average air-side coefficients for the entire channel were then calculated by averaging the coefficients in all the bins.

2.2 Experimental

The experimental portion of this work was used to validate the results from the simulations. A test station was designed and built to measure water vapour flux through the membrane under isothermal conditions in a module that closely resembled the simulation domain in Figure 2.1. Tests were done for various channel aspect ratios with and without ribs and compared to the predictions of the simulation data.

2.2.1 Water Vapour Transport Station

A diagram of the station used for testing is shown in Figure 2.5. Air enters the system after passing through a silica gel dehumidifier. One stream is at a controlled humidity (feed) and the other stays dry (sweep). They are passed over opposite sides of a membrane inside a test module. Analog voltage humidity sensors (Measurement Specialties, HTM2500) take measurements at RH1, RH2, and RH3 while a more accurate chilled mirror sensor (Edgetech DewMaster) measures humidity at RH4. Mass-flow controllers (Alicat MSC-10SLPM) at MFC-1 and MFC-2 dictate the flow rate of each airstream. All testing takes place in an oven which maintains isothermal conditions. The pressure drop along the sweep side of the module is measured between pressure taps with a digital pressure transducer (Setra Model 267).

Air is humidified using an automotive fuel cell humidifier (dPoint Technologies PX1). Water is circulated on one side of the humidifier and air passes through another side. Water moves to the airstream through a permeable membrane. Air leaves the humidifier fully saturated provided the water is at a high enough temperature, which is insured by preheating the water in a reservoir.

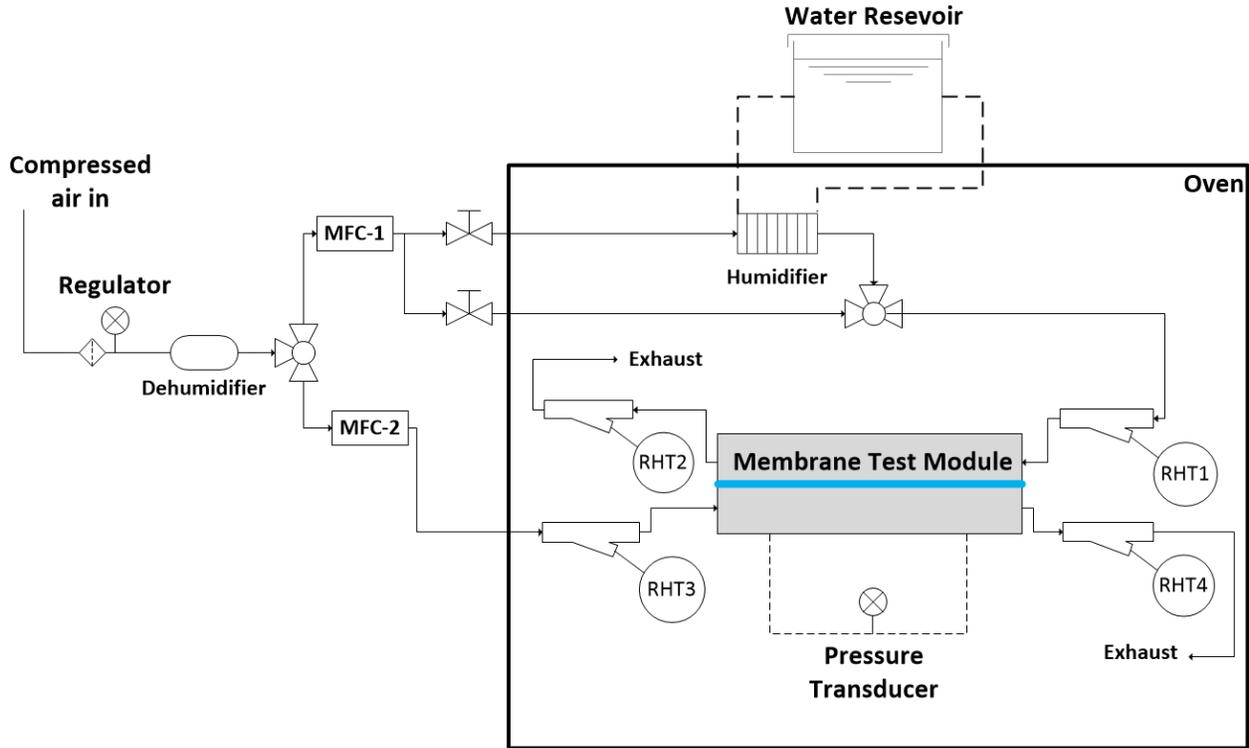


Figure 2.5 – Experimental test station

The recovery effectiveness of the module is

$$\eta_L = \frac{\dot{m}_s(x_4 - x_3)}{\dot{m}_{min}(x_1 - x_3)} \quad (43)$$

Here $x_{1,2,3,4}$ are the humidity ratios at the feed inlet, feed outlet, sweep inlet, and sweep outlet, respectively and $\dot{m}_{f,s}$ is the mass flow rate of dry air into the feed and sweep streams, respectively and where \dot{m}_{min} is the lower of the two.

2.2.2 Testing Module

The membrane flow module is shown in Figure 2.6. It consists of four aluminum plates fastened together with bolts. Membrane material is placed between the two middle plates. Gaskets between each set of plates to create a seal. Feed and sweep streams enter from opposite sides and

opposite ends of the module to create a counter-flow arrangement as shown in Figure 2.7. Pressure taps on the sweep side of the module allow measuring the pressure drop through the passage.

Table 2.1 gives the dimensions of the slots in the aluminum housing. The slot height represents the height, including gasket material, after the module has been fastened together (which compresses the gaskets slightly). Knowing these dimensions, it was possible to 3D print flow channel inserts to replicate the geometries studied in the simulations. The flow passages were created with a Polyjet 3D printer (Stratasys Object 30) using VeroBlue printing material. The inserts are discussed in section 3.3.

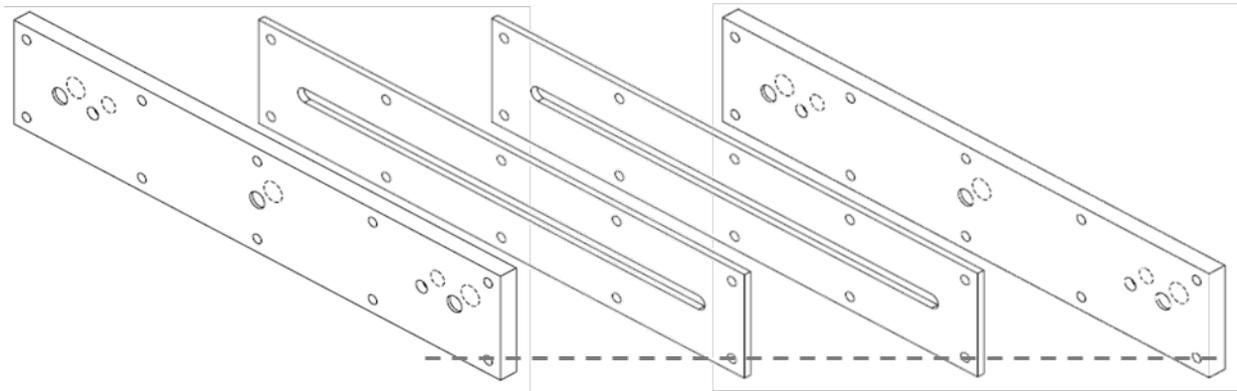


Figure 2.6 – Aluminum housing of testing module

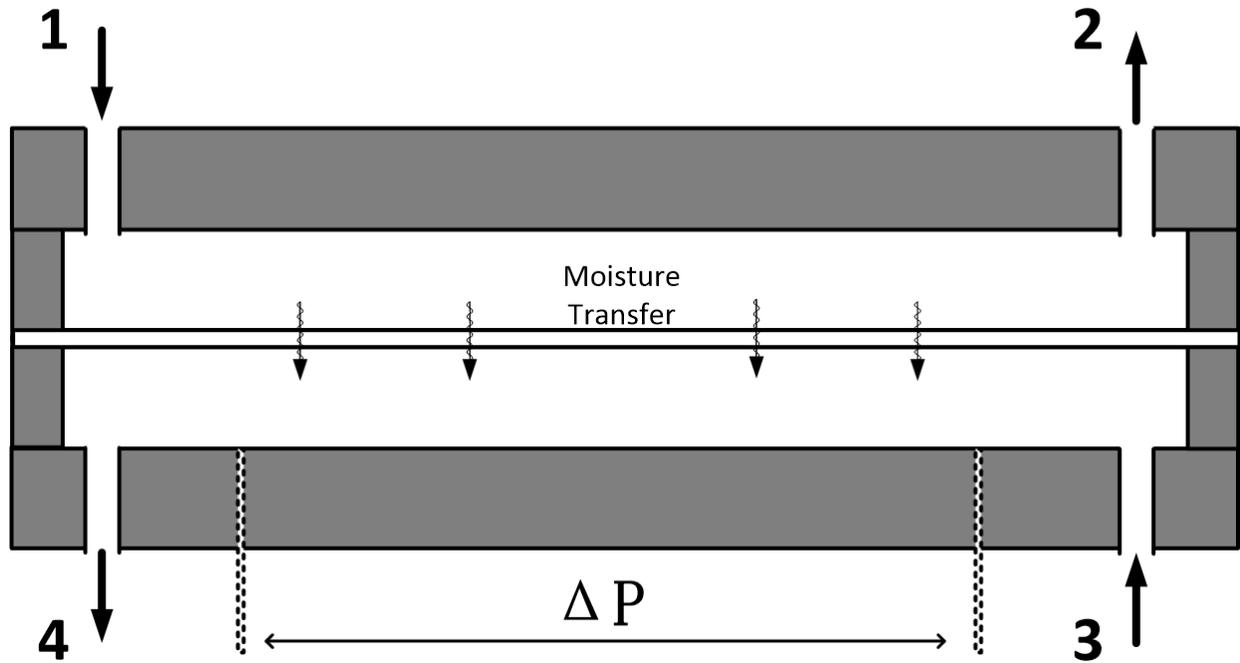


Figure 2.7 – Flow configuration in the membrane housing module

Table 2.1 Dimensions of the housing slot

Parameter	Value (mm)
Housing slot length	300
Housing slot width	10
Housing slot height	6.1

2.2.3 Membrane Forming

The most direct method of validating the simulations in this work is to form rib features into the membrane surface. A two-part aluminum mould was used to emboss rib features into membrane samples. The mould pattern is shown in Figure 2.8. A pressure of 300 psi over the surface of the mould for 10 seconds at room temperature was adequate to create permanent features.

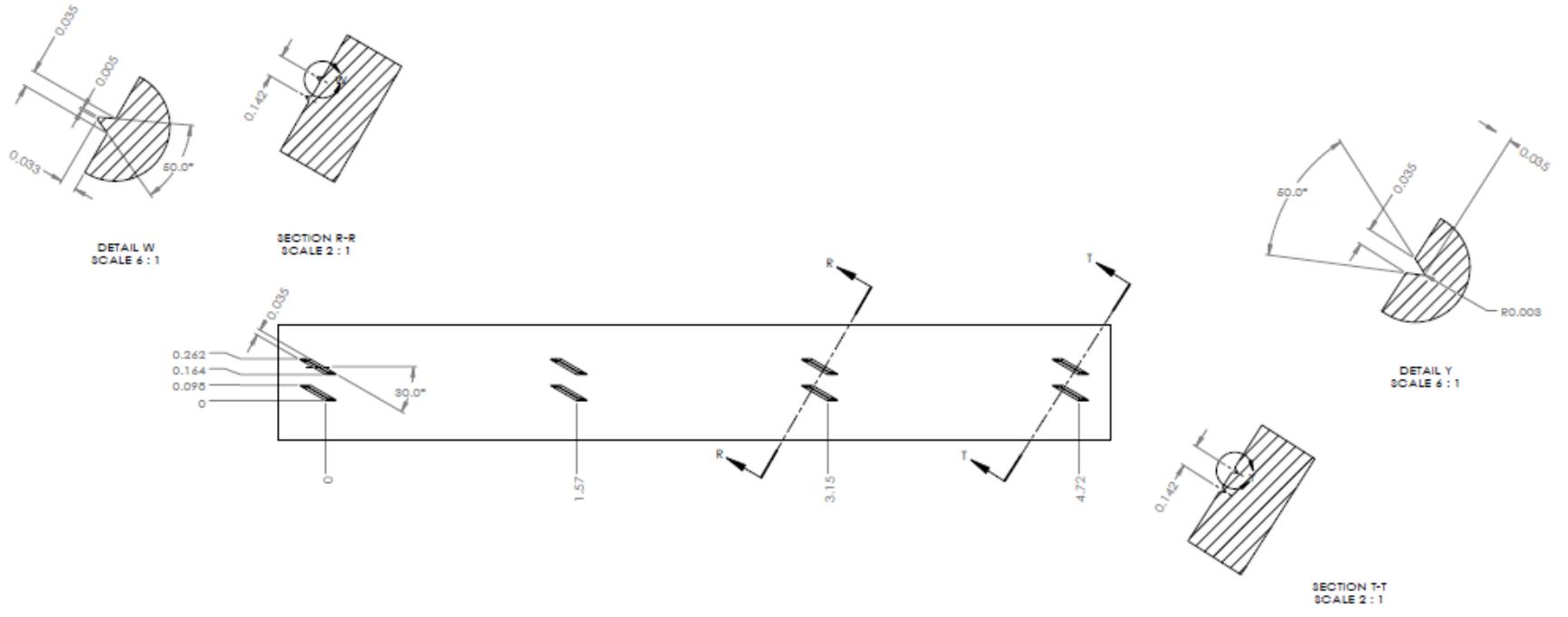


Figure 2.8 - Membrane forming template. Dimension are in inches.

Chapter 3: Results and Discussion

3.1 Simulations

3.1.1 Membrane Ribs

Figure 3.1 shows the transverse velocity vectors 1, 5, and 8 step widths downstream of the rib for three different aspect ratios. These transverse flow patterns move the flow from the top of the channel towards the membrane and reduce the temperature (humidity) heterogeneity, resulting in a higher membrane flux. The recirculation pattern starts out closest to the membrane and moves upward towards the channel centre as the flow moves farther from the rib. As noted by [24], if the top of the channel is also transferring heat (mass), it is better to stagger the ribs on the top and bottom of the channel as this tends to move fluid from the centre of the channel to the sides.

The main vortex is largest and most intense for the $AR = 1$; though, it occupies the largest fraction of the channel cross-section for $AR = 0.6$. Overall, the vortex appears least stable for the $AR = 1.4$ case, indicating that ribs are not as beneficial for aspect ratios higher than 1. There is a smaller, secondary vortex on the opposite side of the main one (far right corner) that may also aid the channel performance.

Figure 3.2 shows how the qualitative behavior of the flow changes with Re . It shows the same types of velocity vectors as Figure 3.1, but with $AR = 1$ for all cases and three different values of Re . All three cases display the same main vortex pattern downstream of the rib, however the vortex intensity increases with increasing Re .

Figure 3.3 shows the fluid speed from the elevation perspective (see Figure 2.1) for the $AR = 1$, $Re = 468$ case. The fluid reaches its maximum speed in the constriction created by the rib. There is a stagnant zone directly upstream of the rib (which is generally poor for heat and mass transfer) and a recirculating zone downstream of the rib. This highlights the importance of picking the correct rib geometry for the channel and flow rate of interest. In this particular case, the recirculation regions are an order of magnitude longer than the stagnant regions, suggesting that overall the effectiveness will increase. If the rib spacing were to be reduced to the point where the rib spacing was similar to the rib height (or width), the stagnant regions could become large enough that the channel effectiveness would actually decrease below the smooth channel value.

Figure 3.4 shows contour plots of humidity and temperature on the membrane surface looking down on the feed side. There are large changes in both variables downstream of the rib. The large mass-transfer resistance of the membrane (relative to heat transfer resistance) can be seen in these plots as the humidity is highly asymmetric between the left and the right compared to the temperature. Note the periodic boundaries in both Figure 3.3 and Figure 3.4.

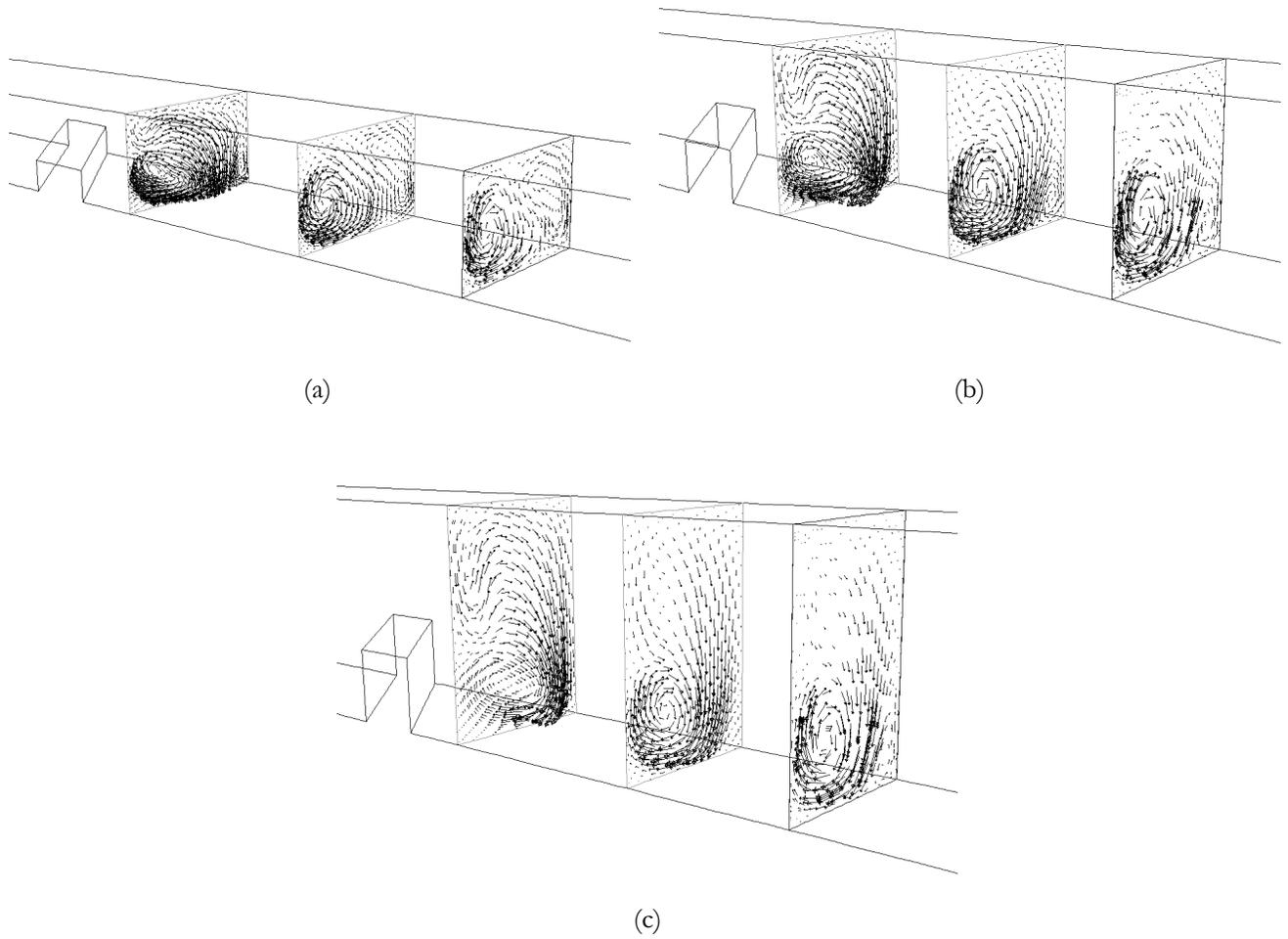
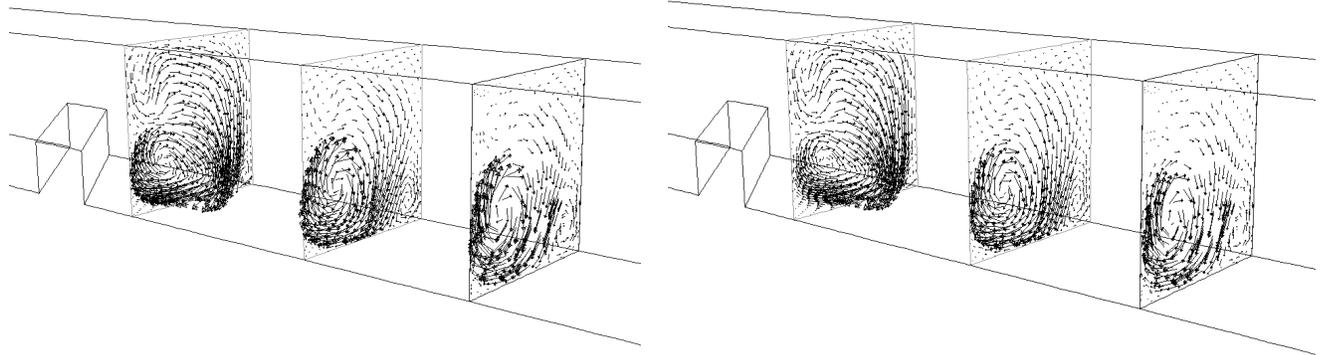
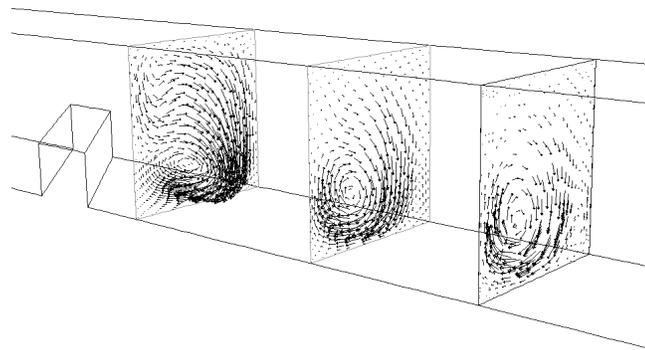


Figure 3.1 - Transverse velocity fields 1, 5, and 8 step widths downstream of the rib for the middle of the three flow rates. (a) $AR = 0.6$, $Re = 390$. (b) $AR = 1$, $Re = 468$. (c) $AR = 1.4$, $Re = 585$.



(a)

(b)



(c)

Figure 3.2 - Transverse velocity fields 1, 5, and 8 step widths downstream of the rib for AR = 1. (a) Re = 703.

(b) Re = 468. (c) Re = 234.

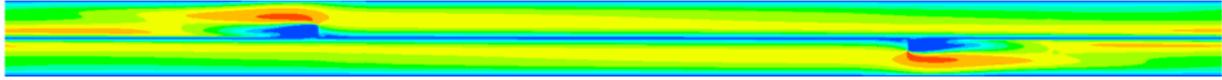


Figure 3.3 – Speed contour plot for the AR = 1, Re = 468 case

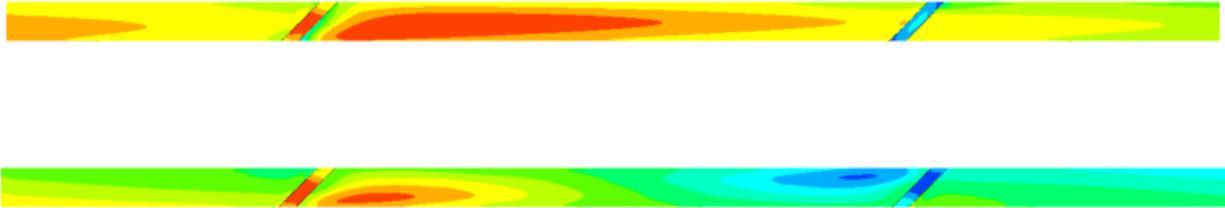


Figure 3.4 – Contour plots of humidity (top) and temperature (bottom) on the membrane surface for the AR = 1, Re = 468 case. Looking down to on the feed side. Red (blue) indicates highest (lowest) humidity and temperature.

The performance of the ribbed channels can be understood in terms of an increase in heat and mass transfer compared to the corresponding increase in pressure drop. These can be interpreted non-dimensionally in terms of Nu, Sh, and fRe .

These performance data are presented locally for the feed side channel in Figure 3.5 for AR = 1 and Re = 468 (Figure 3.1(b) case). The friction value (top) mostly has the same value for the smooth case, except in the immediate vicinity of the ribs. The rib at $\frac{1}{4}$ of the channel length protrudes into the main flow and has the largest effect on fRe , while the second rib protrudes into the opposite channel and has a lesser effect. fRe has a shear component, caused by the shear stress tangential to the main flow, and a normal stress, caused by the components of the rib surface perpendicular to the main flow. A large part of the additional pressure drop in the ribbed channels is caused by this normal component. A notable feature of the friction plot is the

negative spikes directly downstream of the rib. These are caused by the flow reversing direction in the rib wake. The shear component of the friction factor also increases due to the flow speeding up in the constriction created by the rib.

The local Nusselt and Sherwood numbers (bottom) behave differently than the friction coefficient. Both values stay above the smooth channel value over the entire channel length, with the largest increase in the 50% of the channel downstream of the first, protruding rib. There is then a dip in performance as the flow passes over the negative rib, which contains a pocket of stagnant air. Nu and Sh are increased by the flow recirculation shown in Figure 3.1 and, to a lesser extent, by the added membrane surface area on the rib. Nu tends to be higher than Sh due to the higher value of Pr than Sc (ie. the thermal diffusivity is larger than the mass diffusivity).

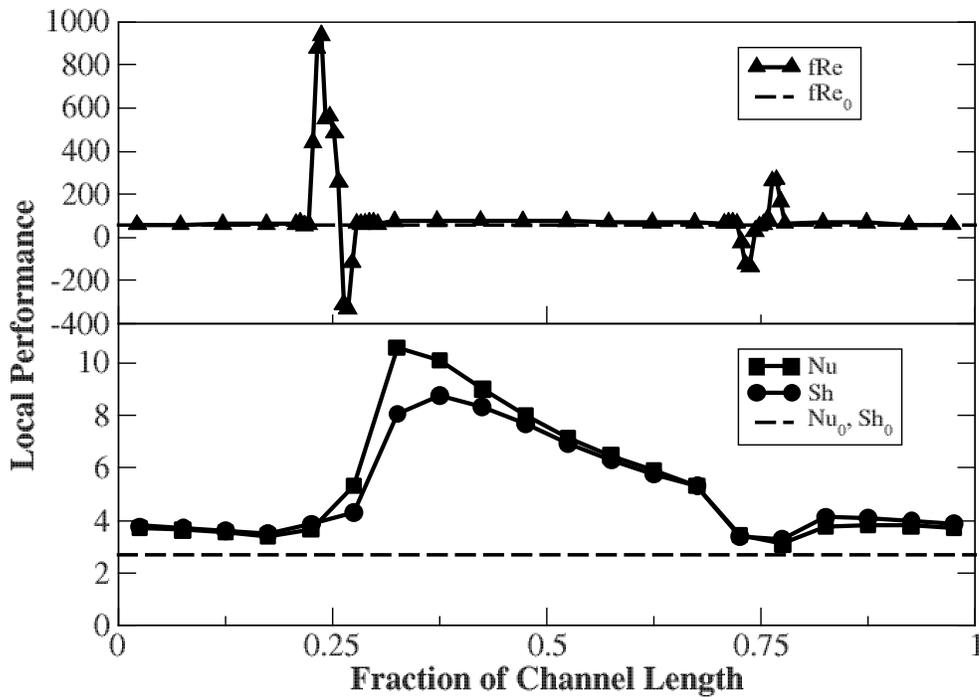


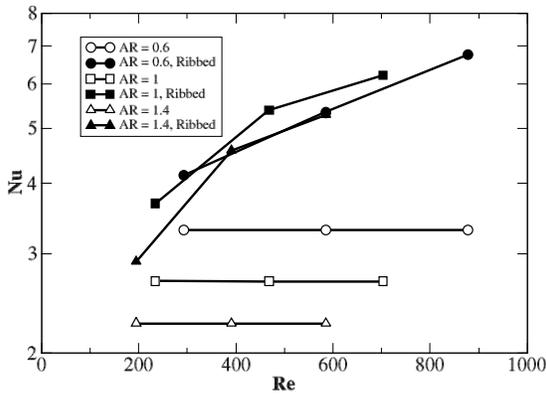
Figure 3.5 – Local performance data along the channel for AR = 1 and Re = 468.

Figure 3.6 shows the overall performance data for all cases studied. For smooth channels, Nu and Sh are independent of Re and closely approximate the case of constant wall heat flux ($Nu, Sh = 2.712$ for $AR = 1$ [31]). Sh and Nu are equal for smooth channels of the same aspect ratio, which is a consequence of the heat and mass transfer analogy being fully satisfied. As the aspect ratio decreases, Nu and Sh increase because the fluid is constrained to be closer to the membrane and the distance required for diffusion to the membrane is shorter. On the other hand, fRe is at a minimum for $AR = 1$ and increases with either an increase or decrease in aspect ratio.

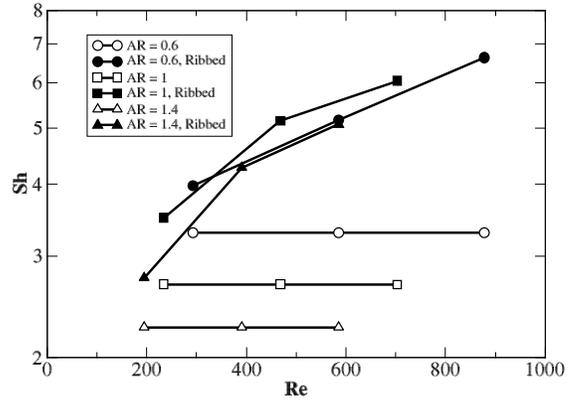
The recirculating flows in the ribbed channels break the heat and mass transfer analogy and Sh , Nu and fRe all increase with Re . Nu and Sh follow a similar pattern as each other but Nu is always slightly higher because Pr is higher than Sc . Ribbed channels with $AR = 1$ tend to produce the highest Nu and Sh . This case is likely most favorable because it allows the flow to circulate more freely around the channel axis (i.e., vortices of aspect ratio one are more stable than higher aspect ratio vortical structures) as shown in Figure 3.1. fRe follows a different pattern and consistently increases with AR , partially because the rib height also increases with AR .

When comparing the ribbed channels to the smooth channels of the same aspect ratio, the increase in Sh and Nu is larger than the corresponding increase in fRe . The relationship is most favorable in the $AR = 1$ ribbed case, shown in Figure 3.6(d). In the range studied, increasing Re tends to increase the benefit gained from ribs.

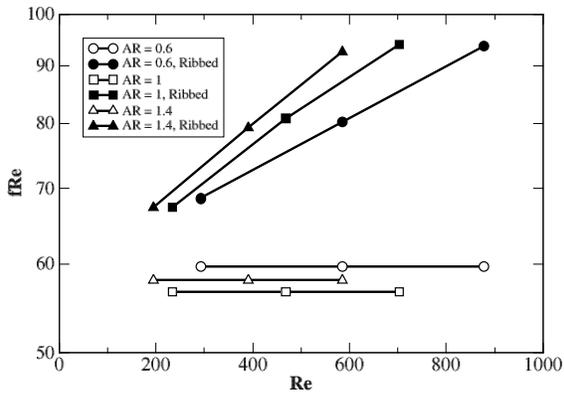
The benefits from ribs shown here could be improved by optimizing the geometric parameters of the ribs. Namely, the rib height, width, spacing, and angle, and shape. For a specific ERV use-case, the optimum values would depend on several factors, such as the flow rate, footprint, and pressure drop requirements. Given that each simulation takes roughly six hours to complete, and using five values for each parameter, an optimization study is estimated to take one month to complete, allowing for time to manually mesh certain configurations. Once the effect of these parameters is better understood, any future optimization studies would be significantly shorter.



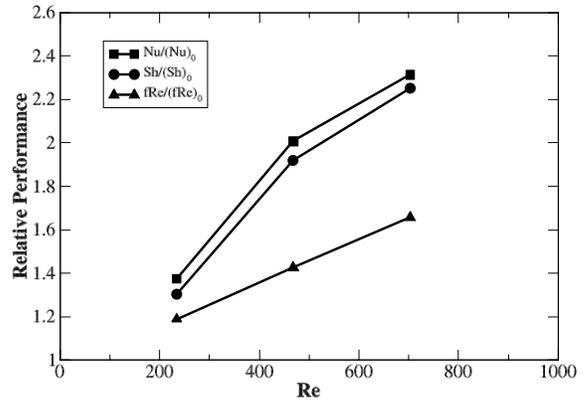
(a)



(b)



(c)



(d)

Figure 3.6 – Performance data of the ribbed and smooth channels for three different flow rates. (a) Nusselt number. (b) Sherwood number. (c) Friction factor times Reynolds number. (d) Relative performance of ribbed and smooth channels for the AR=1 case.

3.1.2 Ribs on Walls

Placing ribs on the top and bottom walls of the modules could be a simpler way of achieving improvement. This geometry is obtained by moving the rib from the membrane surface to the opposite wall, shown in Figure 3.7. The ribs are adiabatic so do not increase the membrane surface area compared to the smooth case, but the recirculating flow pattern is similar.

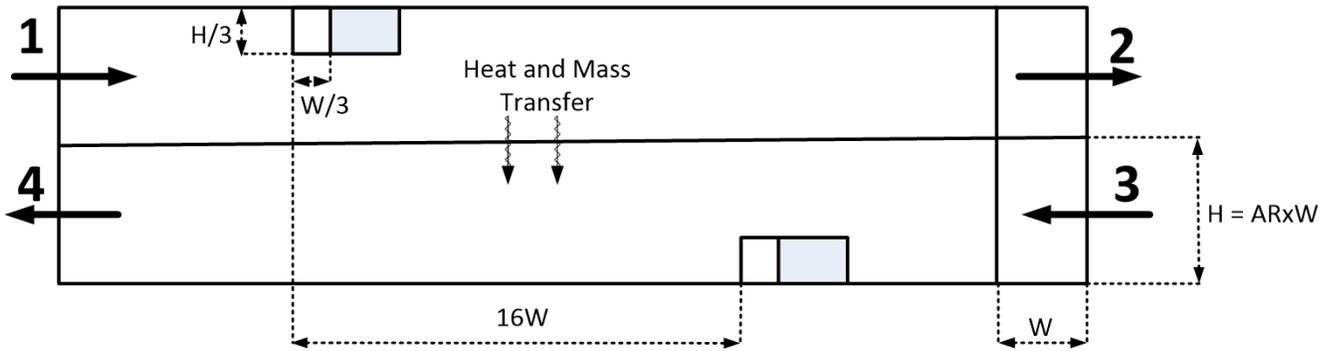


Figure 3.7 - Diagram of ribs on walls

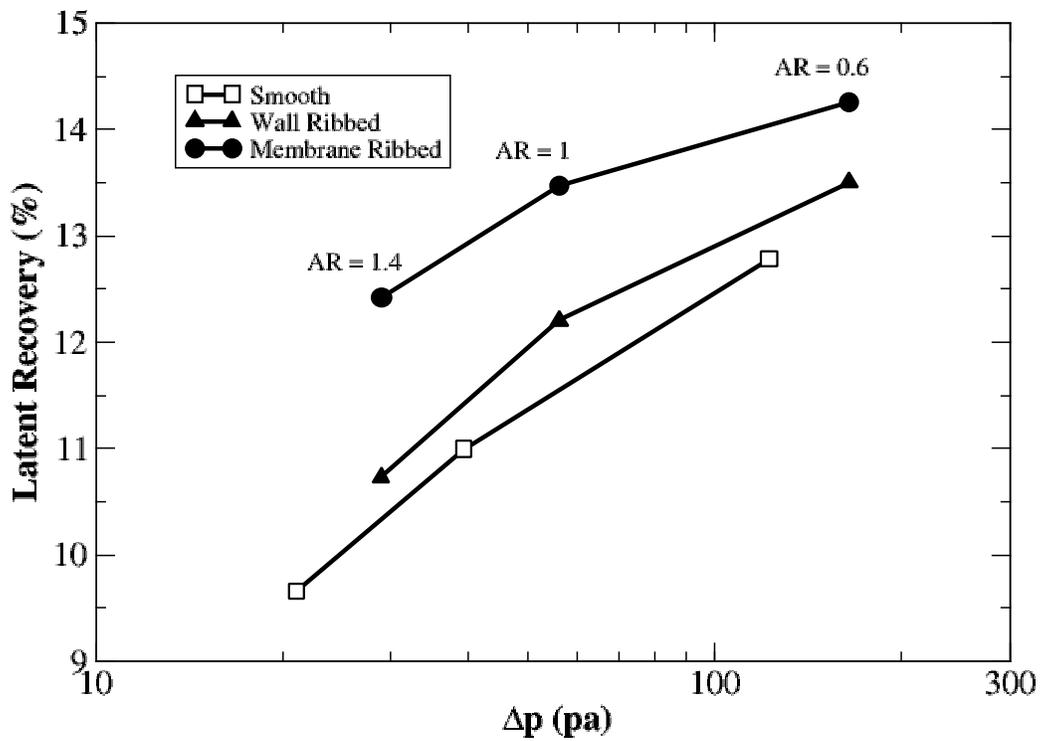


Figure 3.8 – Performance of smooth, membrane ribs, and wall ribs

For lower aspect ratio (height), wall ribs perform similarly to membrane ribs because, as shown in Figure 3.1(a), the downstream vortex covers most of the whole channel. For higher aspect ratios (heights) the vortex is concentrated on the same side as the rib, as shown in Figure 3.1(c).

It is better for performance to have recirculation closer to the membrane surface so membrane ribs perform better in these cases.

As expected, the pressure drop for wall ribs and membrane ribs is identical, because the flow obstruction is the same in both cases. Because membrane ribs incur the same pressure drop but improved recovery, they are the focus of this work.

3.2 Effectiveness Predictions

3.2.1 Counter-Flow Channel

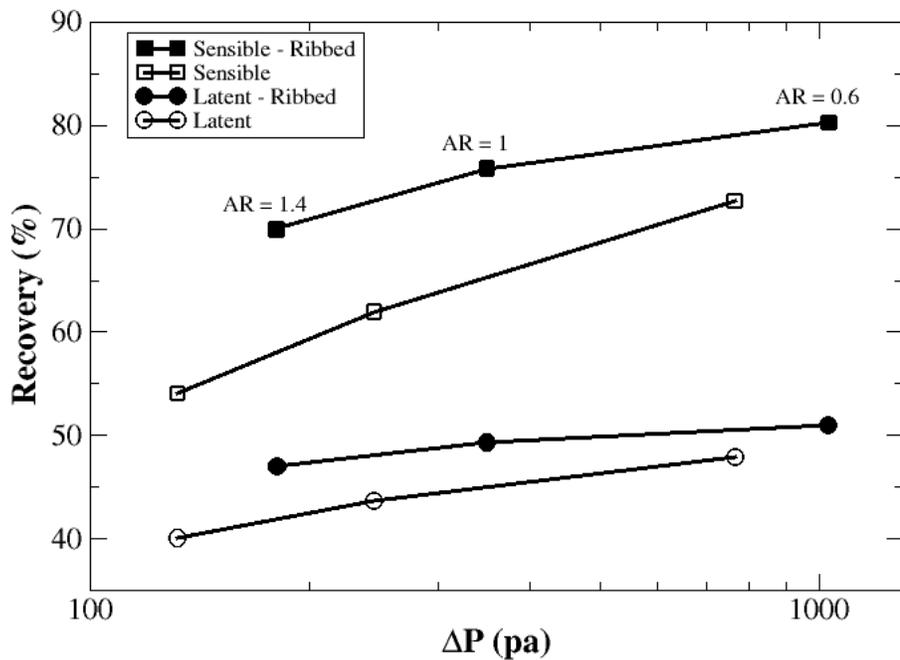


Figure 3.9 – Performance data of a 1m long counter-flow channel

The data in Figure 3.6 can be used with equations (24) and (25) to give the effectiveness of a counter-flow channel of any given length. Figure 3.9 shows these data for a 1m long counter-flow channel with a 2.5mm width and a 1 LPM flow rate (the middle of the three flow rates

considered). This is typical for channels in a commercial-scale ERV. The data are presented for channels with and without ribs for various aspect ratios. For a given pressure drop, the ribbed channel always has a higher recovery effectiveness. This means that the recovery can be increased for an equal pressure drop by adding ribs and increasing the channel height.

The sensible recovery in Figure 3.9 is consistently higher than the latent recovery. This is primarily because the membrane heat transfer resistance is lower than its mass transfer resistance but also because the air-side value of k increases more than h with ribs. This means that the rib enhancement is more apparent for heat transfer.

Figure 3.10 shows the recovered power for a single channel in AHRI summer conditions (Feed stream: 35.0°C dry bulb and 25.6°C wet bulb, Sweep stream: 23.9°C dry bulb and 17.2°C wet bulb) [32]. The fan power was calculated with a fan efficiency of 0.56. It is notable that the recovered power is 2-3 orders of magnitude larger than the fan power, which is indicative of the overall energy effectiveness of the ERV.

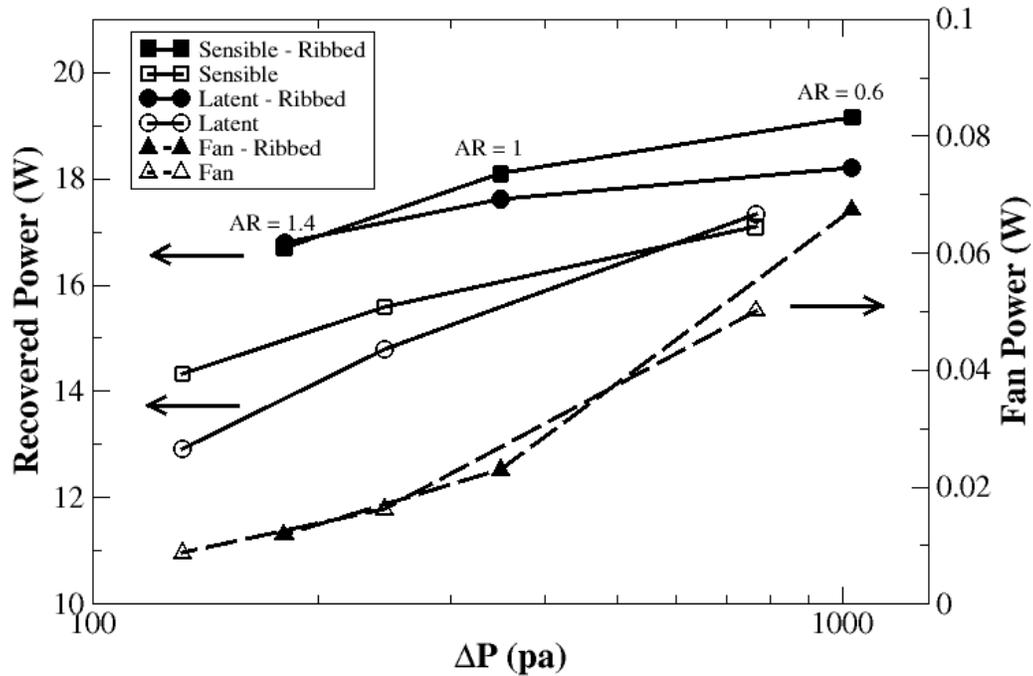


Figure 3.10 - Recovered power and fan power for AHRI standard summer operating conditions.

3.2.2 Cross-Flow

Cross-flow cores are the most popular arrangement due to their relative ease of construction, even though they are not as effective as a counter-flow arrangement. A typical cross flow arrangement is shown in Figure 3.11. It is constructed by stacking layers in alternate orthogonal directions separated by membrane material. Each layer is supported by a spacer that separates the flow into multiple parallel channels. The end result is that there is no heat or mass transfer on the side walls of each channel, but there is transfer through the top and bottom membrane surfaces.

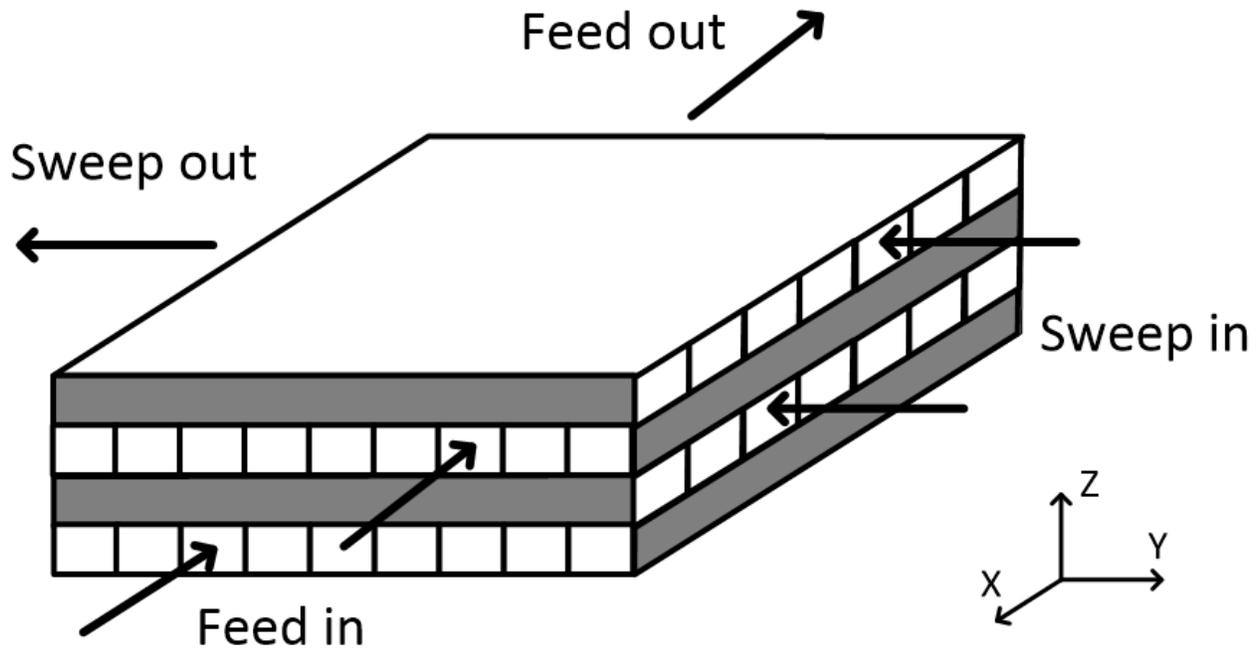


Figure 3.11 - Cross-flow ERV configuration

The boundary conditions in a cross-flow arrangement are different than the counter-flow arrangement presented in Section 3.1.1. For one, there is flux through both the top and bottom of the channel (the other walls remain adiabatic). The top and bottom walls can be approximated as constant wall temperature (humidity). To see why, consider that the sweep side channels are all identical and so will all have the same temperature (humidity) distribution for any given position along the y-axis (neglecting edge effects). Now consider a channel on the feed side which flows along the x-axis at a constant value of y. Because the channels above and below are all sweep-side channels at a set value of y, they will all have the same temperature (humidity) distribution, making these fields approximately constant on the membrane surface. The same argument can be made for any channel on the sweep side. The simulation presented thus far can be modified to consider this case by considering only one of the two channels and making the top and bottom

walls (including ribs) a constant wall temperature (humidity) boundary condition. Note that in this configuration every second layer of membrane has ribbed features while the others are flat.

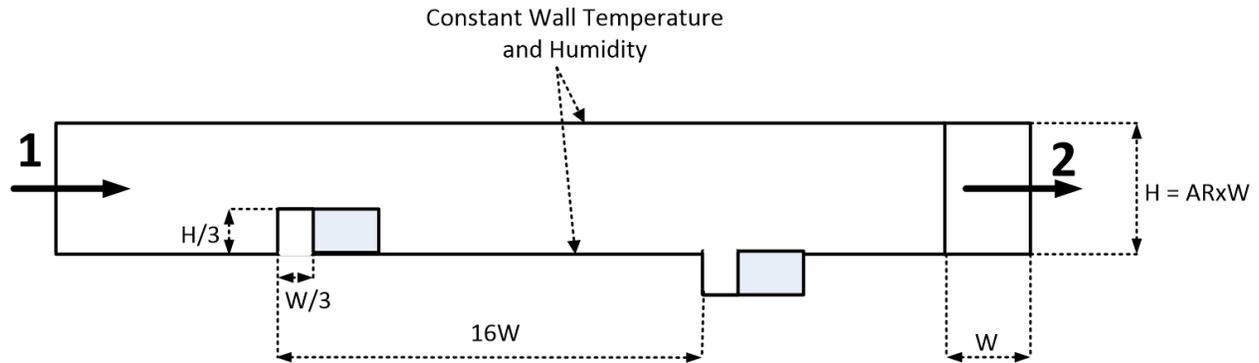


Figure 3.12– Cross flow core simulation domain

Because the channel shape and fluid properties are the same as the counter-flow arrangement, the local values of f are the same as that case. On the other hand, the local air-side Nu and Sh values are different and are shown in Figure 3.13. These values behave qualitatively similar to the counter-flow case, in that there is a positive spike downstream of the membrane and then a gradual return to the smooth-channel values. Also as in the counter-flow case, there is an overall increase in Nu and Sh with increasing Re (which corresponds to increasing flow rate in this case). However, because the smooth-channel performance is higher in this arrangement – due to both the top and bottom walls transferring heat and mass – the performance increase from ribs is not as high as it is for the counter-flow case.

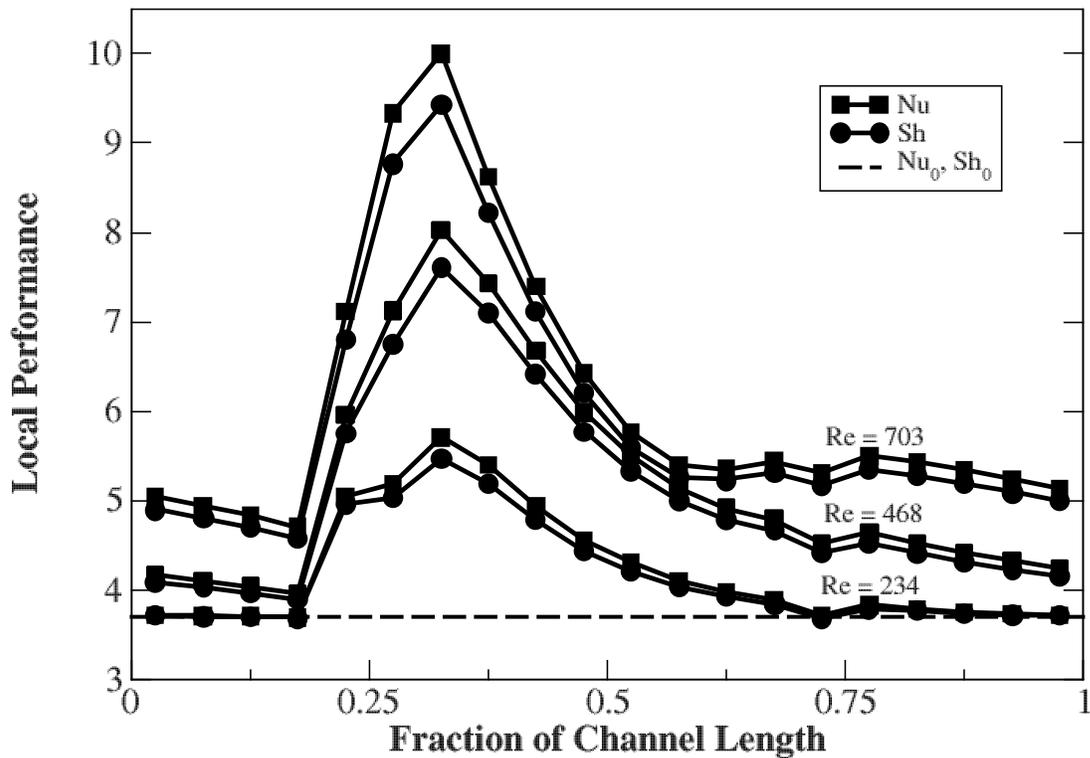
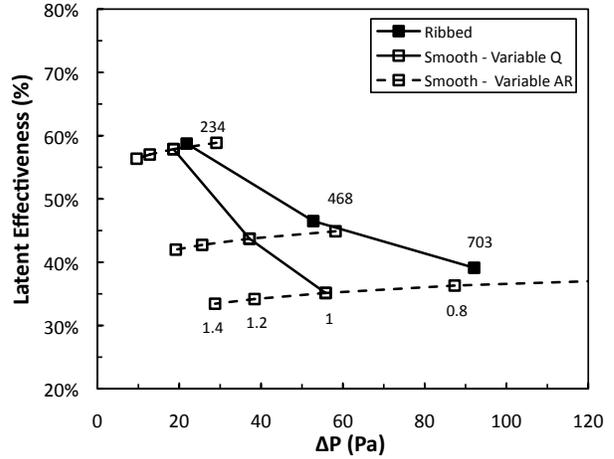
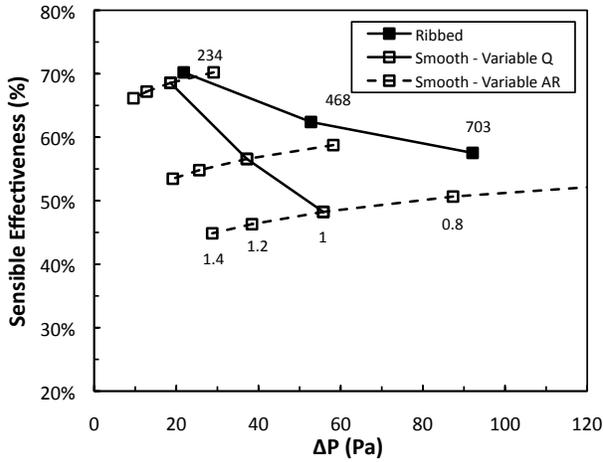


Figure 3.13– Cross flow local Nu and Sh values for various values of Re

To find the predicted performance of a cross-flow core, the local values of Nu, Sh, and fRe were used as inputs to a MATLAB model which has been described in [33]. For this section of the work the core geometry was made up of 60 layers with 64 channels per layer. Each layer was of 160mmx160mm. The results are shown in Figure 3.14. As was case with the counter-flow predictions, the ribbed channels consistently have higher performance than the smooth channels. The solid lines give the performance of $AR = 1$ channels for three different flow rates. The dashed lines show the change in smooth channel performance as the channel height is decreased for a constant flow rate. For a given flow rate and pressure drop, the ribbed channel always has a higher effectiveness than the smooth channel. In the range of Re studied, the sensible improvement is between 1.1% to 6.7% and the latent improvement is between 0.6% to

2.7%. The sensible effectiveness improves more than the latent effectiveness, largely because the membrane mass transfer resistance is so high that the improvement in air-side resistance is less significant. Another reason for this is that Nu increases more than Sh with ribs, as shown in Figure 3.13. A final notable feature of these plots is that as the flow rate (and Re) increases, the overall effectiveness decreases for both smooth and ribbed cases because the residence time of an air parcel in the channel decreases. Despite this, the difference in effectiveness between ribbed and smooth channels increases because of the larger Nu and Sh in the ribbed channels. Physically, the larger Re increases the mixing in the ribbed channels and partially compensates for the decreased residence time.

By assuming AHRI Summer operating conditions (Feed stream: 35.0°C dry bulb and 25.6°C wet bulb, Sweep stream: 23.9°C dry bulb and 17.2°C wet bulb [32]), the sensible and latent effectiveness can be converted to total recovered power, shown in Figure 3.15. Here the recovered power increases with increasing flow rate because, even though the effectiveness is decreasing, the total available heat moving through the ERV is larger. Again, it can be seen that ribbed channels can significantly increase recovery for an equal pressure drop compared to smooth channels. For the $Re = 703$ case, the ribbed core recovers nearly 100W more than the smooth channel with an equal pressure drop.



(a)

(b)

Figure 3.14. - Variation of the effectiveness of enthalpy exchangers with smooth and ribbed channels for three different operating flowrates. (a) Sensible effectiveness. (b) Latent effectiveness. Dashed lines indicate effectiveness variation for smooth channels with variable AR of 1.4, 1.2, 1, and 0.8 at constant flowrate.

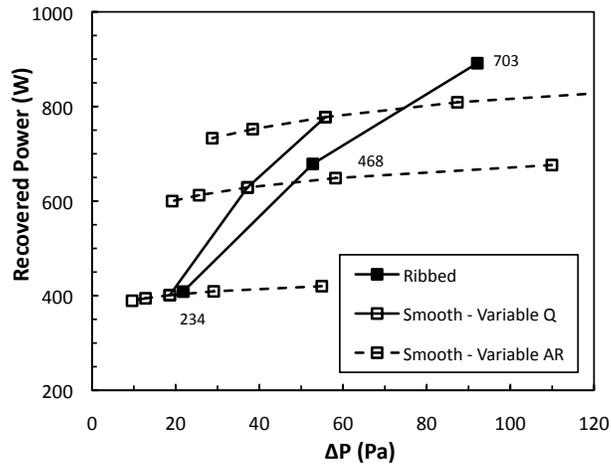


Figure 3.15. - Total recovered power of enthalpy exchangers with smooth and ribbed channels for three different operating flowrates in AHRI Summer (cooling) operating conditions. Dashed lines indicate effectiveness variation for smooth channels with variable AR of 1.4, 1.2, 1, and 0.8 from left to right.

3.3 Experimental Validation

The CFD model was validated against experiments conducted in the test stand and counter-flow module described in section 2.2. Tests were performed isothermally at 50°C with the feed stream at 50%RH (40.4g/kg) and the sweep stream kept dry (<0.3%RH, 0.2g/kg). Tests were run with a flow of 2.15 SLPM on the feed and sweep side. Because the sweep inlet was dehumidified ($x_3 = 0$) and the flow rates were equal on each stream, the recovery effectiveness simplifies to

$$\eta_L = \frac{x_4}{x_1} \quad (44)$$

The flow passages were 3D printed to replicate the configurations from the simulations, that is, smooth channels and ribbed channels of various aspect ratios. Each insert consists of a pair of parallel channels so that the overall flow rate could be increased to a nominal level for the mass flow controllers (1-3 SLPM) and to increase the amount of active membrane area.

Two methods were considered for creating the ribbed features. In the first, the ribs were 3D printed as extensions of the channels. This method has the advantage of being able to create rectangular angled rib features with a high degree of accuracy, however the ribs cannot be permeable extensions of the membrane as they were considered in the simulations. Instead, they are impermeable ribs on top of the membrane. Figure 3.16 shows an example of one of these ribbed channels. The channel shown is for the AR = 1 case and has 2.5mmx2.5mm channel dimensions.

Another method of creating the rib features is to form them into the membrane surface, as was considered in the simulations. Because a rectangular rib feature would tear the membrane on

forming, a triangular rib feature was selected instead. Also, the ribs were at an angle of 30 degrees with respect to the main flow direction as this geometry left less space at the rib edges for the flow to go around the rib. These modifications were added to the simulation so they could be compared directly to the experiments. Ribs were formed with a mould as described in section 2.2.3. The final formed rib features are shown in Figure 3.17.

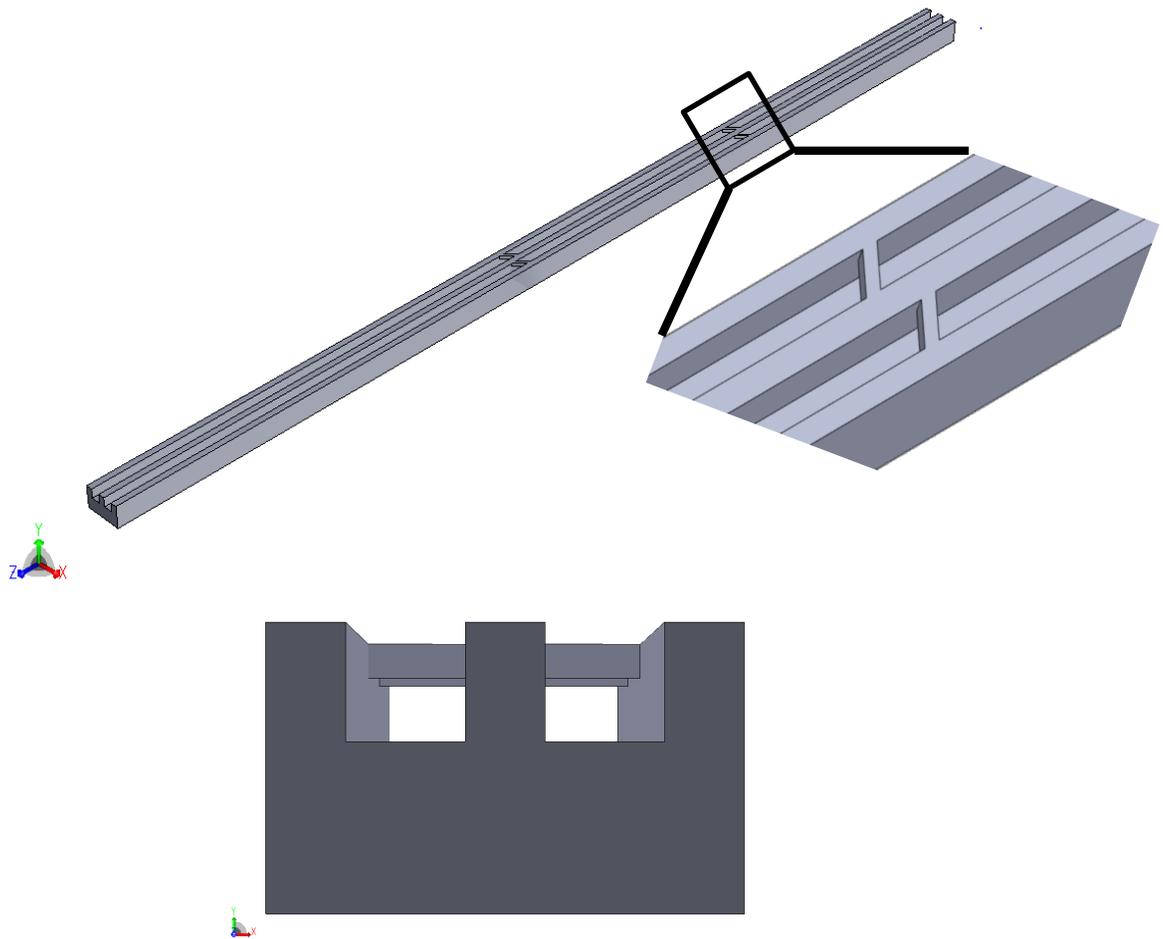


Figure 3.16 –3D-Printed Flow channel insert. Case shown is 2.5mmx2.5mm channel with plastic ribs near the membrane surface. Isometric view (top) and end view (bottom).

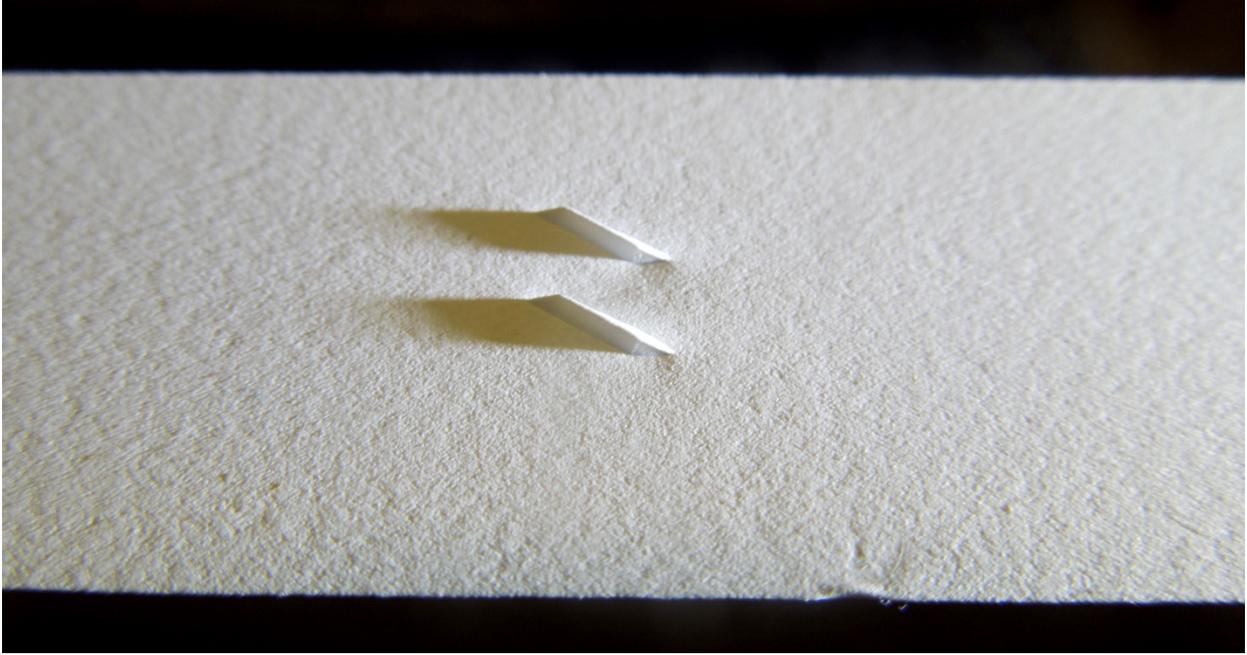


Figure 3.17- Rib features formed into the membrane

Each channel insert was 289mm long with a 62mm entrance length on each leaving 165mm length for active transfer across the membrane. The entrance regions were covered in impermeable tape so that no humidity transport could occur across the membrane. They were required to create a known velocity distribution at the start of the active membrane area. The length, L_e , required for a fully developed velocity distribution in the laminar regime is given by [9, p. 180]

$$L_e/D_h = 0.0575Re \quad (45)$$

The velocity profile for a rectangular channel has the analytical solution [34]

$$u = -\frac{16c_1a^2}{\pi^3} \sum_{n=1,3,\dots}^{\infty} \frac{1}{n^3} (-1)^{(n-1)/2} \left[1 - \frac{\cosh\left(\frac{n\pi y}{2a}\right)}{\cosh\left(\frac{n\pi b}{2a}\right)} \right] \cos\left(\frac{n\pi x}{2a}\right) \quad (46)$$

$$u_m = -\frac{c_1 a^2}{3} \left[1 - \frac{192}{\pi^5} \left(\frac{a}{b}\right) \sum_{n=1,3,\dots}^{\infty} \frac{1}{n^5} \tanh\left(\frac{n\pi b}{2a}\right) \right] \quad (47)$$

Where a and b are $\frac{1}{2}$ the channel width and height ($AR = 2b/2a$), and the coordinate system origin is at the cross-section centre, as shown in Figure 3.18. The UDF for implementing this profile is included in Appendix A.3.

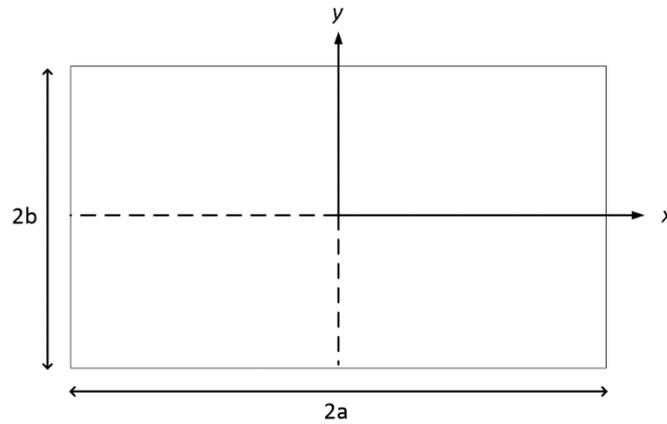


Figure 3.18 - Rectangular channel cross-section and dimensions

Two channel aspect ratios were considered for experiments, 2.5mmx2.5mm ($AR=1$) and 1.5mmx2.5mm ($AR = 0.6$). However, because the 3D printer used (Stratasys Object 30) had a reported accuracy of 0.1mm, the channel cross-section dimensions were measured with gauge blocks to find the actual channel dimensions. It was possible to measure the dimension to within 0.01mm this way.

In order to make the simulations and experiments comparable, the simulations geometry was matched to the dimensions in Table 3.1. The fluid properties, listed in Table 3.3, were also used to match the test conditions of 50°C and atmospheric pressure. The velocity profile at the inlets

was assumed to be fully developed and given by equation (46). The humidity, on the other hand, was considered to be well mixed so the profile was uniform. The flow rate on each side was 2.15 SLPM, the feed inlet humidity was 40.4 g/kg and the sweep inlet humidity was 0 g/kg (the same as the experiment conditions).

Table 3.2 Measured dimensions of the flow channel inserts

Channel	Width (mm)	Height (mm)
AR = 1	2.38	2.43
AR = 1, Ribbed	2.31	2.41
AR = 0.6	2.28	1.43

Table 3.3 Air properties assumed at 50°C and 1atm

Property	Value
Density, ρ	1.086 kg/m ³ [9]
Viscosity, μ	1.962 × 10 ⁻⁵ kg/m/s [9]
Mass Diffusivity, D	2.90 × 10 ⁻⁵ m ² /s [35]

Figure 3.19 shows the experimental results and corresponding simulations predictions. Results for smooth channels, a channel with 3D printed ribs, and a channel with formed membrane ribs are shown. Error bars on the experimental data represents the 95% confidence interval based on the sample standard deviation. Error bars on the WRR of the simulations data represents the uncertainty on the membrane permeability used. Quality control data from dPoint technologies indicates that the reported permeability of a given roll of membrane samples has an uncertainty

of +/- 6%. The experimental and simulation data agree within this uncertainty. As expected the ribbed channels outperform the smooth channels.

Another feature is that because the membrane ribs are permeable, the membrane ribs outperform the printed ribs. Note, however, that this effect has been exaggerated by the dimensions of the channels, listed in Table 3.2. In particular, the printed AR = 1, ribbed channel has a smaller width than its smooth AR = 1 counterpart, which tends to decrease the WRR performance. Therefore, the experimental data should be taken mainly as an indicator of the simulation accuracy.

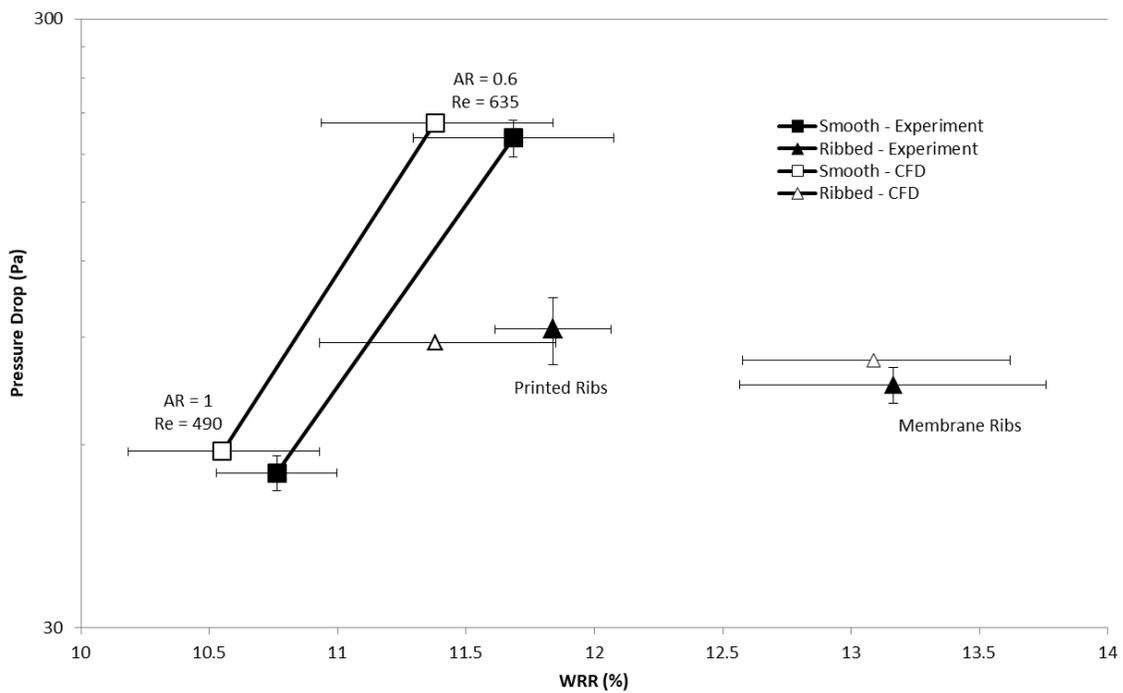


Figure 3.19 – Experimental performance of smooth and ribbed channels with corresponding simulation predictions.

Conclusions

Angled rib features can be an effective method for increasing the effectiveness of membrane-based energy recovery ventilators. This study showed that adding ribs to a channel can increase the heat and mass transfer coefficients enough to outweigh the corresponding increase in friction factor. The magnitude of the heat transfer enhancement is consistent with previous studies. Furthermore, for realistic values for the membrane mass transfer resistance, the latent heat effectiveness of the ERV is also substantially improved by the addition of ribs. For smooth channels, Nu , Sh , and fRe are independent of Reynolds number and equal to the constant-wall-heat-flux case. For ribbed channels, they are strong functions of Re . The influence of channel aspect ratio on effectiveness was also studied. An aspect ratio of 1 generally gave the most improvement with ribs, as that aspect ratio yielded the most robust vortical structure downstream of the rib.

Adding ribs and slightly increasing the channel height can increase recovery effectiveness by over 10% for an equal pressure drop. Even with constant channel dimensions, the ribbed channel is energetically preferable to a smooth channel because the increase in recovered energy is 2-3 orders of magnitude larger than the corresponding increase in fan power (during typical summer conditions). That is, under realistic conditions, the energy benefit of the ribs far outweighs the energy penalty due to increased friction.

Experiments were conducted to validate the simulations. Ribs were either 3D printed and rested on the membrane surface (and so were impermeable) or formed into the membrane surface. Both configurations were found to be consistent with experimental predictions.

Traditionally, ERVs meet their effectiveness requirements by using more permeable membrane materials, decreasing the channel dimensions, or increasing the channel length. The data reported here show that ribs embossed on the membrane surface could be used instead, yielding a better enhancement/pressure relation than simply changing the channel dimensions. Also, embossing the membrane to create ribs would not require additional materials and would not entail changing the footprint of the core, which are both advantages relative to changing the channel dimensions.

Future work in this area should focus on testing rib features in a full-sized ERV core. The current study showed that the benefit of ribs increases with Re , in the range studied, so a larger Re would make validation simpler. The rib geometry (height, width, angle, and spacing) could be optimized for the channel dimensions and Reynolds number chosen in order to maximize the sensible and latent improvement. In addition to increasing the net benefit of ribs, this would make the improvements easier to measure.

References

- [1] Natural Resources Canada, “Energy Use Data Handbook Tables,” 15-Dec-2004. [Online]. Available: <http://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/menus/trends/handbook/tables.cfm>. [Accessed: 02-Oct-2015].
- [2] “Buildings Energy Data Book,” US Department of Energy, 2011, p. 2.
- [3] ASHRAE, “ASHRAE handbook fundamentals,” 2009, p. 9.12.
- [4] ASHRAE, “ASHRAE Standard 62.1-2016 - Ventilation for Acceptable Indoor Air Quality.” 2016.
- [5] L.-Z. Zhang, *Total heat recovery heat and moisture recovery from ventilation air*. New York: Nova Science Publishers, 2008.
- [6] R. Huizing, W. Mérida, and F. Ko, “Impregnated electrospun nanofibrous membranes for water vapour transport applications,” *J. Membr. Sci.*, vol. 461, pp. 146–160, Jul. 2014.
- [7] A. Engarnevis, “Amin Engarnevis PhD Proposal - The Influence of Humidity, Temperature, and Particulate Fouling on Permeation Properties of Dense Polymer Membranes.” 18-May-2015.
- [8] L.-Z. Zhang, *Conjugate Heat and Mass Transfer in Heat Mass Exchanger Ducts*. Academic Press, 2013.
- [9] J. R. Welty, Ed., *Fundamentals of momentum, heat, and mass transfer*, 5th ed. Danver, MA: Wiley, 2008.
- [10] J. Larminie and A. Dicks, “Fuel cell systems explained,” 2nd ed., Chichester, West Sussex: J. Wiley, 2003, pp. 80–90.
- [11] R. Huizing, M. Fowler, W. Mérida, and J. Dean, “Design methodology for membrane-based plate-and-frame fuel cell humidifiers,” *J. Power Sources*, vol. 180, no. 1, pp. 265–275, May 2008.
- [12] R. W. Baker, *Membrane Technology and Applications*. Academic Press, 2012.
- [13] L.-Z. Zhang, “Heat and mass transfer in a quasi-counter flow membrane-based total heat exchanger,” *Int. J. Heat Mass Transf.*, vol. 53, no. 23–24, pp. 5478–5486, Nov. 2010.
- [14] S. Metz, W. Vandeven, J. Potreck, M. Mulder, and M. Wessling, “Transport of water vapor and inert gas mixtures through highly selective and highly permeable polymer membranes,” *J. Membr. Sci.*, vol. 251, no. 1–2, pp. 29–41, Apr. 2005.
- [15] S. V. Patankar, C. H. Liu, and E. M. Sparrow, “Fully developed flow and heat transfer in ducts having streamwise-periodic variations of cross-sectional area,” *ASME J Heat Transf.*, vol. 99, no. 2, pp. 180–186, 1977.
- [16] L. Y. Qiu Zhong, “An optimized crossflow plate-fin membrane-based total heat exchanger,” *Energy Build.*, vol. 86, pp. 550–556, 2015.
- [17] S. Koester, A. Klasen, J. Lölsberg, and M. Wessling, “Spacer enhanced heat and mass transfer in membrane-based enthalpy exchangers,” *J. Membr. Sci.*, vol. 520, pp. 566–573, Dec. 2016.
- [18] M. Fiebig, T. Güntermann, and N. K. Mitra, “Numerical Analysis of Heat Transfer and Flow Loss in a Parallel Plate Heat Exchanger Element With Longitudinal Vortex Generators as Fins,” *J. Heat Transf.*, vol. 117, no. 4, pp. 1064–1067, Nov. 1995.

- [19] P. Deb, G. Biswas, and N. K. Mitra, "Heat transfer and flow structure in laminar and turbulent flows in a rectangular channel with longitudinal vortices," *Int. J. Heat Mass Transf.*, vol. 38, no. 13, pp. 2427–2444, 1995.
- [20] S. K. Saha, "Thermal and friction characteristics of laminar flow through rectangular and square ducts with transverse ribs and wire coil inserts," *Exp. Therm. Fluid Sci.*, vol. 34, no. 1, pp. 63–72, Jan. 2010.
- [21] C.-O. Olsson and B. Sunden, "Experimental study of flow and heat transfer in rib-roughened rectangular channels," *Exp. Therm. Fluid Sci.*, vol. 16, pp. 349–365, 1998.
- [22] G. Tanda and R. Abram, "Forced Convection Heat Transfer in Channels With Rib Turbulators Inclined at 45 deg," *J. Turbomach.*, vol. 131, no. 2, p. 21012, 2009.
- [23] K. M. Kim, H. Lee, B. S. Kim, S. Shin, D. H. Lee, and H. H. Cho, "Optimal design of angled rib turbulators in a cooling channel," *Heat Mass Transf.*, vol. 45, no. 12, pp. 1617–1625, Oct. 2009.
- [24] C. Nonino and G. Comini, "Convective heat transfer in ribbed square channels," *Int. J. Numer. Methods Heat Fluid Flow*, vol. 12, no. 5, pp. 610–628, Aug. 2002.
- [25] "FLUENT 6.3 UDF Manual." [Online]. Available: https://www.sharcnet.ca/Software/Fluent6/html/udf/main_pre.htm. [Accessed: 30-May-2017].
- [26] M. Staudacher, M. Harasek, T. Brinkmann, W. Hilgendorff, and A. Friedl, "CFD-simulation of mass transfer effects in gas and vapour permeation modules," *Desalination*, vol. 146, no. 1, pp. 237–241, Sep. 2002.
- [27] M. Coroneo, G. Montante, J. Catalano, and A. Paglianti, "Modelling the effect of operating conditions on hydrodynamics and mass transfer in a Pd–Ag membrane module for H₂ purification," *J. Membr. Sci.*, vol. 343, no. 1–2, pp. 34–41, Nov. 2009.
- [28] Y. Benguerba, J. Amer, and B. Ernst, "CFD modeling of the H₂/N₂ separation with a nickel/ α -alumina microporous membrane," *Chem. Eng. Sci.*, vol. 123, pp. 527–535, Feb. 2015.
- [29] R. Al-Waked, M. S. Nasif, G. Morrison, and M. Behnia, "CFD simulation of air to air enthalpy heat exchanger," *Energy Convers. Manag.*, vol. 74, pp. 377–385, Oct. 2013.
- [30] P. J. Roache, "Quantification of uncertainty in computational fluid dynamics," *Annu. Rev. Fluid Mech.*, vol. 29, no. 1, pp. 123–160, 1997.
- [31] A. L. London, Ed., "Advances in HEAT TRANSFER A2 - Shah, R.K.," in *Laminar Flow Forced Convection in Ducts*, Academic Press, 1978, p. ii.
- [32] AHRI, "2013 Standard for Performance Rating of Air-to-Air Exchangers for Energy Recovery Ventilation Equipment." 2013.
- [33] A. Engarnevis, "Conjugate heat and mass transfer modeling in membrane-based enthalpy exchangers," *Unpublished Manusc.*, 2017.
- [34] R. K. Shah and A. L. London, *Laminar Flow Forced Convection in Ducts: A Source Book for Compact Heat Exchanger Analytical Data*. Academic Press, 2014.
- [35] G. Nellis and S. Klein, *Heat Transfer*. Cambridge University Press, 2009.

Appendix A – User Defined Functions

The functions below are compiled within fluent and used to run the simulation and post-process the results

A.1 Membrane transport, Nusselt number, Sherwood number, and Friction Factor

```
#include "udf.h"
#define del 0.000105 /*thickness of membrane in m*/
#define R 8.314 /*universal gas constant J/K/mol*/
#define L 0.160 //length in m
#define Dia 0.00143//hydraulic diameter in m
#define n 40 //divide channel into this many bins
#define nm 400 //n for cf calculation
#define k_air 0.0262 //thermal conductivity of air in w/m/K (assuming 30C)
#define D_air_h20 2.6e-5 //from fundamental of heat and mass transfer book
#define Tw_i 303 //initial wall temperature (for constant wall temperature cases)
#define T_film 298.15 //temperature to convert concentration to partial pressure
#define WALL_ID 3 //ID of the dry wall surface
#define WALL_ID_WET 5 //ID of the wet wall surface
#define DRY_ID 7 //cell zone id of dry
#define WET_ID 6 //cell zone id of wet
#define WALL_ID_2 9//adiabatic dry wall
#define inD "real-1"
real D;

/*prepares cells for implementation of the source term (so only wall cells are
affected)*/
DEFINE_INIT(initialize,d)
{
face_t f;
cell_t c, c0, c1;
real NV_VEC(farea);
real area;

Thread *t, *t0, *t1, *ct;

d = Get_Domain(1);

/* Initialize Cells */
/* this loops over all cells and lets the UDM = 0 */

thread_loop_c(ct, d)
{
begin_c_loop(c,ct)
{
C_UDMI(c,ct,0) = 0;
}
end_c_loop(c,ct)
}

/* Loop over all faces on wall */
t = Lookup_Thread(d, WALL_ID);
```

```

begin_f_loop(f,t)
{
/* c0 and t0 identify the adjacent dry cell, c1 and t1 identifies the corresponding humid
cell */
c0 = F_C0(f, t);
c1 = F_C1(f,t);
t0 = THREAD_T0(t);
t1 = THREAD_T1(t);
/* this loops over all cells adjacent to wall and lets the UDM = 1.0 for dry, and -1 for
humid*/
C_UDMI(c0, t0, 0) = 1;
C_UDMI(c1, t1, 0) = -1;
C_UDMI(c0, t0, 3) = 1;
C_UDMI(c1, t1, 3) = -1;

C_UDMI(c0, t0, 1) = c1; /*sets partner for dry cell*/
C_UDMI(c1, t1, 1) = c0; /*sets partner for humid cell*/

C_UDMI(c0, t0, 4) = F_YI(f,t,0); //initializes the stored wall concentration for cases
when we have a uniform wall concentration
C_UDMI(c1, t1, 4) = F_YI(f,t,0);

C_UDMI(c1, t1, 5) = Tw_i; // temporarily hard coding in the wall temp
C_UDMI(c0, t0, 5) = Tw_i; // temporarily hard coding in the wall temp

/* this is still in loop, and stores the face area of each cell on the membrane. They are
assumed to be the same for each side */
F_AREA(farea,f,t);
area = NV_MAG(farea);

C_UDMI(c0, t0, 2) = area; /*sets face area for dry cell*/
C_UDMI(c1, t1, 2) = area; /*sets face area humid cell*/

}
end_f_loop(f, t)

}

/*****
/* UDF for specifying h2O mass source */
/* */
*****/
DEFINE_SOURCE(m_source, c, t, dS, eqn)
{
    Domain *d;
    real area, vol, source = 0;
    Thread *t1, *tfd, *tfh;
    cell_t c1;
    D = RP_Get_Input_Parameter(ind); /*permiability of membrane in mol-m/m^2/s/pa,
from paramaeters list */

    d = Get_Domain(1);
    tfd = Lookup_Thread(d, WALL_ID); /* gives the faces thread of membrane-shadow*/
    tfh = Lookup_Thread(d, WALL_ID_WET); /* gives the faces thread of membrane*/

    area = C_UDMI(c, t, 2); /* area of cell face on membrane*/

```

```

    vol = C_VOLUME(c, t); /* volume of cell*/

    if (C_UDMI(c, t, 0) == 1) /*dry cell*/
    {
        t1 = THREAD_T0(tfh); /*identifies the humid stream cell thread using the
faces thread of membrane*/
    }

    else if (C_UDMI(c, t, 0) == -1) /*humid cell*/
    {
        t1 = THREAD_T0(tfd); /*identifies the dry stream cell thread using the
faces thread of membrane shadow*/
    }

    else
    {
        source = 0;
        dS[eqn] = 0;
        return source;
    }

    c1 = C_UDMI(c, t, 1); /*identifies the partner of the cell, taken from results of
initilization*/

    source = D*R*T_film*(C_R(c1, t1)*C_YI(c1, t1, 0) - C_R(c, t)*C_YI(c, t, 0))*area /
(vol*del);
    dS[eqn] = -D*R*C_R(c, t)*T_film*area/(vol*del);

    return source;
}

/*****
/* UDF for specifying an h2O fraction on membrane */
/* */
*****/
DEFINE_PROFILE(membrane_conc, tf0, i)
{
    cell_t c0, c1;
    Thread *t0, *t1, *tf1;
    real x0[ND_ND], xf0[ND_ND], A[ND_ND]; /* this will hold the
position vector */
    real y, faceYI;
    face_t f, f0, f1;
    float dy, dr;
    D = RP_Get_Input_Parameter(inD);

    begin_f_loop(f0, tf0)
    {
        /* c0 and t0 identify the adjacent cell, c1 and t1 identify the opposite
cell */
        c0 = F_C0(f0, tf0);
        t0 = THREAD_T0(tf0);
        c1 = F_C1(f0, tf0);
        t1 = THREAD_T1(tf0);

        f1 = F_SHADOW(f0, tf0);

```

```

        tf1 = THREAD_SHADOW(tf0);

        C_CENTROID(x0, c0, t0);
        F_CENTROID(xf0, f0, tf0);

        dy = sqrt(pow(x0[0] - xf0[0], 2) + pow(x0[1] - xf0[1], 2) + pow(x0[2] -
xf0[2], 2)); //distance between wall and cenn center

        if (N_ITER>0) { //only do this after initialization
            faceYI = C_YI(c0, t0, 0) - (D*R / (D_air_h20*del))*T_film*(F_YI(f0,
tf0, 0) - F_YI(f1, tf1, 0))*C_R(c1, t1) / C_R(c0, t0))*dy;
            F_PROFILE(f0, tf0, i) = faceYI; //set wall concentration to create
the desired gradient
            C_UDMI(c0, t0, 4) = faceYI;
            C_UDMI(c0, t0, 5) = F_T(f0,tf0);
        }
        else {
            F_PROFILE(f0, tf0, i) = 0;
        }
    }
end_f_loop(f0, tf0)
}

/*****
/* UDF for getting Sherwood number
/*
/*
*****/

DEFINE_OUTPUT_PARAMETER(Sh, m, parlist)
{
    Domain *d; //declare domain variable, since it is not passed by the function
    Thread *t, *tw, *tc;
    cell_t c;
    face_t f;
    real x[ND_ND], x0[ND_ND], area[ND_ND]; //for position and area
    int k, i;
    real dy;

    int num[n] = { 0 }; //hold the number of cells in each bin
    real T_sum[n] = { 0 }, T_norm[n] = { 0 }, T_b[n] = { 0 };
    real H_sum[n] = { 0 }, H_norm[n] = { 0 }, H_b[n] = { 0 };
    int numf[n] = { 0 }; //hold the number of faces in each bin
    real Tf_sum[n] = { 0 }, Tf_norm[n] = { 0 }, T_w[n] = { 0 };
    real Hf_sum[n] = { 0 }, Hf_norm[n] = { 0 }, H_w[n] = { 0 };
    real Qf_sum[n] = { 0 }, Qf_norm[n] = { 0 }, Q_w[n] = { 0 };
    real Jf_sum[n] = { 0 }, Jf_norm[n] = { 0 }, J_w[n] = { 0 };
    real Nu[n] = { 0 }, Sh[n] = { 0 }; //nusseIt number at bin n
    real Nu_av = 0, Sh_av = 0;
    d = Get_Domain(1);
    real a = parlist[0], b = parlist[1];

    real Dh = 4*a*b / (a + b); //compute the hydraulic diameter. a is 1/2 the hight, b
is 1/2 the width

    //bin cells by z position and calculate bulk values

```

```

tc = Lookup_Thread(d, DRY_ID);
begin_c_loop(c, tc)
{
    C_CENTROID(x, c, tc); //get the position of the cell
    C_UDMI(c, tc, 0) = floor(x[2] * n / L); //set memory of cell to its bin
number
    k = C_UDMI(c, tc, 0); //get the bin number in integer form
    if (k < n && k >= 0)
    {
        T_sum[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc)*C_T(c, tc);
//integrate to get bulk temperature
        T_norm[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc);
//normalization factor

        H_sum[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc)*C_R(c,
tc)*C_YI(c, tc, 0); //integrate to get bulk concentration
        H_norm[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc);

        num[k]++; //increment the count for that bin
    }
}
end_c_loop(c, tc)

//bin faces by z position
tw = Lookup_Thread(d, WALL_ID);
begin_f_loop(f, tw)
{
    c = F_C0(f, tw); //get cell index of face neighbor
    tc = THREAD_T0(tw);
    F_CENTROID(x, f, tw); //get the position of the face
    C_CENTROID(x0, c, tc); //get the position of the neighbor cell
    F_AREA(area, f, tw); //get face area
    F_UDMI(f, tw, 1) = floor(x[2] * n / L); //set memory of face to its bin
number
    k = F_UDMI(f, tw, 1); //get the bin number in integer form
    dy = sqrt(pow(x[0] - x0[0], 2) + pow(x[1] - x0[1], 2) + pow(x[2] - x0[2],
2)); //distance between wall and cell center

    Tf_sum[k] += NV_MAG(area)*F_T(f, tw); //integrate the wall temperature
    Tf_norm[k] += NV_MAG(area); //normalization factor, area including rib
    Hf_sum[k] += NV_MAG(area)*C_R(c, tc)*F_YI(f, tw, 0); //integrate to get
wall h20 concentration
    Hf_norm[k] += NV_MAG(area); //normalization factor, area including rib
    Qf_sum[k] += NV_MAG(area)*k_air*(F_T(f, tw) - C_T(c, tc)) /
dy; //BOUNDARY_HEAT_FLUX(f,tw);
    Qf_norm[k] += fabs(area[1]); //normalization factor, area not including rib
    Jf_sum[k] += NV_MAG(area)*D_air_h20*C_R(c, tc)*(F_YI(f, tw, 0) - C_YI(c,
tc, 0)) / dy; //boundary H20 flux
    Jf_norm[k] += fabs(area[1]); //normalization factor, area not including rib
    numf[k]++; //increment the count for that bin
}
end_f_loop(f, tw)

//get the bulk temperature for each bin

```

```

for (int i = 0; i < n; i++)
{
    T_b[i] = T_sum[i] / T_norm[i];
    T_w[i] = Tf_sum[i] / Tf_norm[i];
    Q_w[i] = Qf_sum[i] / Qf_norm[i];

    H_b[i] = H_sum[i] / H_norm[i];
    H_w[i] = Hf_sum[i] / Hf_norm[i];
    J_w[i] = Jf_sum[i] / Jf_norm[i];

    Nu[i] = Dh*Q_w[i] / (k_air*(T_w[i] - T_b[i])); //calculate nusselt number
in each bin
    Sh[i] = Dh*J_w[i] / (D_air_h20*(H_w[i] - H_b[i])); //calculate sherwood
number in each bin

    Message("Bin %d: Nu = %g, Sh = %g , H_b = %g, H_w = %g, J_w = %g \n", i,
Nu[i], Sh[i], H_b[i], H_w[i], J_w[i]);
}

for (int i = n/4; i <= 3*n/4-1; i++) {
    Nu_av += Nu[i] / (n/2); //average over the middle of tube
    Sh_av += Sh[i] / (n/2);
}
//Message("Average Nu at end: %g, Average Sh at end: %g \n", Nu_av, Sh_av);
return Sh_av;
}

/*****
/* UDF for getting Nusselt number */
/* */
*****/

DEFINE_OUTPUT_PARAMETER(Nu, m, parlist)
{
    Domain *d; //declare domain variable, since it is not passed by the function
    Thread *t, *tw, *tc;
    cell_t c;
    face_t f;
    real x[ND_ND], x0[ND_ND], area[ND_ND]; //for position and area
    int k, i;
    real dy;

    int num[n] = { 0 }; //hold the number of cells in each bin
    real T_sum[n] = { 0 }, T_norm[n] = { 0 }, T_b[n] = { 0 };
    real H_sum[n] = { 0 }, H_norm[n] = { 0 }, H_b[n] = { 0 };
    int numf[n] = { 0 }; //hold the number of faces in each bin
    real Tf_sum[n] = { 0 }, Tf_norm[n] = { 0 }, T_w[n] = { 0 };
    real Hf_sum[n] = { 0 }, Hf_norm[n] = { 0 }, H_w[n] = { 0 };
    real Qf_sum[n] = { 0 }, Qf_norm[n] = { 0 }, Q_w[n] = { 0 };
    real Jf_sum[n] = { 0 }, Jf_norm[n] = { 0 }, J_w[n] = { 0 };
    real Nu[n] = { 0 }, Sh[n] = { 0 }; //nusselt number at bin n
    real Nu_av = 0, Sh_av = 0;
    d = Get_Domain(1);
    real a = parlist[0], b = parlist[1];

```

```

    real Dh = 4 * a*b / (a + b); //compute the hydraulic diameter. a is 1/2 the hight,
    b is 1/2 the width

                                                                    //bin cells by z position
    tc = Lookup_Thread(d, DRY_ID);
    begin_c_loop(c, tc)
    {
        C_CENTROID(x, c, tc); //get the position of the cell
        C_UDMI(c, tc, 0) = floor(x[2] * n / L); //set memory of cell to its bin
number
        k = C_UDMI(c, tc, 0); //get the bin number in integer form
        if (k < n && k >= 0)
        {
            T_sum[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc)*C_T(c, tc);
//integrate to get bulk temperature
            T_norm[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc);
//normalization factor

            H_sum[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc)*C_R(c,
tc)*C_YI(c, tc, 0); //integrate to get bulk concentration
            H_norm[k] += fabs(C_W(c, tc))*C_VOLUME(c, tc)*C_R(c, tc);

            num[k]++; //increment the count for that bin
        }
    }
    end_c_loop(c, tc)

    //bin faces by z position
    tw = Lookup_Thread(d, WALL_ID);
    begin_f_loop(f, tw)
    {
        c = F_C0(f, tw); //get cell index of face neighbor
        tc = THREAD_T0(tw);
        F_CENTROID(x, f, tw); //get the position of the face
        C_CENTROID(x0, c, tc); //get the position of the neighbor cell
        F_AREA(area, f, tw); //get face area
        F_UDMI(f, tw, 1) = floor(x[2] * n / L); //set memory of face to its bin
number
        k = F_UDMI(f, tw, 1); //get the bin number in integer form
        dy = sqrt(pow(x[0] - x0[0], 2) + pow(x[1] - x0[1], 2) + pow(x[2] - x0[2],
2)); //distance between wall and cell center

        Tf_sum[k] += NV_MAG(area)*F_T(f, tw); //integrate the wall temperature
        Tf_norm[k] += NV_MAG(area); //normalization factor, area including rib
        Hf_sum[k] += NV_MAG(area)*C_R(c, tc)*F_YI(f, tw, 0); //integrate to get
wall h20 concentration
        Hf_norm[k] += NV_MAG(area); //normalization factor, area including rib
        Qf_sum[k] += NV_MAG(area)*k_air*(F_T(f, tw) - C_T(c, tc)) /
dy; //BOUNDARY_HEAT_FLUX(f,tw);
        Qf_norm[k] += fabs(area[1]); //normalization factor, area not including rib
        Jf_sum[k] += NV_MAG(area)*D_air_h20*C_R(c, tc)*(F_YI(f, tw, 0) - C_YI(c,
tc, 0)) / dy; //boundary H20 flux
        Jf_norm[k] += fabs(area[1]); //normalization factor, area not including rib
        numf[k]++; //increment the count for that bin

```

```

}
end_f_loop(f, tw)

//get the bulk temperature for each bin
for (int i = 0; i < n; i++)
{
    T_b[i] = T_sum[i] / T_norm[i];
    T_w[i] = Tf_sum[i] / Tf_norm[i];
    Q_w[i] = Qf_sum[i] / Qf_norm[i];

    H_b[i] = H_sum[i] / H_norm[i];
    H_w[i] = Hf_sum[i] / Hf_norm[i];
    J_w[i] = Jf_sum[i] / Jf_norm[i];

    Nu[i] = Dh*Q_w[i] / (k_air*(T_w[i] - T_b[i])); //calculate nusselt
number in each bin
    Sh[i] = Dh*J_w[i] / (D_air_h20*(H_w[i] - H_b[i])); //calculate
sherwood number in each bin

    Message("Bin %d: Nu = %g, Sh = %g , T_b = %g, T_w = %g, Q_w = %g
\n", i, Nu[i], Sh[i], T_b[i], T_w[i], Q_w[i]);
}

for (int i = n / 4; i <= 3 * n / 4 - 1; i++) {
    Nu_av += Nu[i] / (n/2); //average over the end of tube
    Sh_av += Sh[i] / (n/2);
}
Message("Average Nu in middle: %g, Average Sh in middle: %g \n", Nu_av, Sh_av);
return Nu_av;
}

/*****
/* UDF for getting wall stress (for friction factor) */
/* */
*****/

DEFINE_OUTPUT_PARAMETER(cf, m, parlist)
{
    Domain *d; //declare domain variable, since it is not passed by the function
    Thread *t, *tw, *tc;
    cell_t c;
    face_t f;
    real wallshear[ND_ND], x[ND_ND], x0[ND_ND], area[ND_ND]; //for position, area, and
wall shear
    int k, i;

    real F_w[nm] = { 0 };
    real Tau[nm] = { 0 };
    real Tau_av = 0;
    real F_tot = 0;
    int walls[2] = { WALL_ID, WALL_ID_2 };
    d = Get_Domain(1);
    real a = parlist[0], b = parlist[1];

```

```

    real Dh = 4 * a*b / (a + b); //compute the hydraulic diameter. a is 1/2 the hight,
    b is 1/2 the width
    real A_part = L * 4 * (a + b)/nm; //area of each bin

    //bin faces by z position
    for (int i = 0; i < 2; i++)
    {
        tw = Lookup_Thread(d, walls[i]);
        begin_f_loop(f, tw)
        {
            F_AREA(area, f, tw); //get face area
            F_CENTROID(x, f, tw); //get the position of the face
            k = floor(x[2] * nm / L); //set memory of face to its bin number

            NV_V(wallshear, =, C_STORAGE_R_NV(f, tw, SV_WALL_SHEAR)); //get wall
shear stress
            F_w[k] += F_P(f, tw)*area[2] - wallshear[2]; //total force in z-
direction is the sum of shear and pressure components
        }
        end_f_loop(f, tw)
    }

    //get the bulk temperature for each bin
    for (int i = 0; i < nm; i++)
    {
        Tau[i] = F_w[i] / A_part; //total wall stress in each bin
        F_tot += F_w[i];

        Message("Bin %d: Tau = %g\n", i, Tau[i]);
    }

    for (int i = nm / 4; i <= 3 * nm / 4 - 1; i++)
    {
        Tau_av += Tau[i] / (nm / 2); //average over the end of tube
    }
    Message("Average Tau in middle: %g, F_tot: %g \n", Tau_av, F_tot);
    return Tau_av;
}

```

A.2 Periodic Boundary Conditions

```
#include "udf.h"
#define OUT_ID_DRY 11 //ID of the dry wall surface
#define OUT_ID_WET 12 //ID of the wet wall surface
#define IN_ID_DRY 13 //cell zone id of dry
#define IN_ID_WET 10 //cell zone id of wet
#define N 1000
#define Rho 1.184 //density of dry air
#define T_wet 308.15//T of dry air
#define T_dry 297.0388889//T of wet air
#define X_wet 0.01677 //this is actually Y, the humidity mass fraction
#define X_dry 0.00751 //this is actually Y, the humidity mass fraction
#define ina "real-2" //1/2 width of channel cross-section
#define inb "real-3" //1/2 height of channel cross-section
#define N_START 5 //number of iterations to wait before starting periodic

DEFINE_PROFILE(periodic_velocity, tf0, i)
{
    Domain *d;
    Thread *tf1, *t;
    cell_t c;
    real x0[ND_ND], x1[ND_ND], vel[N][N];
    face_t f0, f1;
    int xn, yn;
    real a, b;

    a = RP_Get_Input_Parameter(ina)*2; //get channel dimensions from parameter list
    b = RP_Get_Input_Parameter(inb)*2;

    d = Get_Domain(1);

    if (tf0 == Lookup_Thread(d, IN_ID_DRY)) //match up dry inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_DRY);
    }
    else if (tf0 == Lookup_Thread(d, IN_ID_WET)) //match up wet inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_WET);
    }

    begin_f_loop(f1, tf1) //store the velocity field of the onlet in the array
    {
        if (N_ITER > 0) { //only do this after first iteration
            F_CENTROID(x1, f1, tf1);
            xn = floor(fabs(x1[0] * N / a));
            yn = floor(fabs(x1[1] * N / b));
            vel[xn][yn] = pow(pow(F_U(f1, tf1), 2)+ pow(F_V(f1, tf1), 2)+
pow(F_W(f1, tf1), 2), 0.5);
        }
    }
    end_f_loop(f1,tf1)

    begin_f_loop(f0, tf0) //apply the velocity field of the outlet to the inlet
    {
```

```

        F_CENTROID(x0, f0, tf0);
        xn = floor(fabs(x0[0] * N / a));
        yn = floor(fabs(x0[1] * N / b));

        if (N_ITER>N_START) {
            F_PROFILE(f0, tf0, i) = fabs(vel[xn][yn]);
        }
        else {
            F_PROFILE(f0, tf0, i) = 1;
        }
    }
end_f_loop(f0, tf0)
}
DEFINE_PROFILE(periodic_u, tf0, i)
{
    Domain *d;
    Thread *tf1, *t;
    cell_t c;
    real x0[ND_ND], x1[ND_ND], vel[N][N];
    face_t f0, f1;
    int xn, yn;
    real a, b;

    a = RP_Get_Input_Parameter(ina) * 2; //get channel dimensions from parameter list
    b = RP_Get_Input_Parameter(inb) * 2;
    d = Get_Domain(1);

    if (tf0 == Lookup_Thread(d, IN_ID_DRY)) //match up dry inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_DRY);
    }
    else if (tf0 == Lookup_Thread(d, IN_ID_WET)) //match up wet inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_WET);
    }

    begin_f_loop(f1, tf1) //store the velocity field of the outlet in the array
    {
        if (N_ITER > 0) { //only do this after first iteration
            F_CENTROID(x1, f1, tf1);
            xn = floor(fabs(x1[0] * N / a));
            yn = floor(fabs(x1[1] * N / b));
            vel[xn][yn] = F_U(f1, tf1);
        }
    }
end_f_loop(f1, tf1)

    begin_f_loop(f0, tf0) //apply the velocity field of the outlet to the inlet
    {
        F_CENTROID(x0, f0, tf0);
        xn = floor(fabs(x0[0] * N / a));
        yn = floor(fabs(x0[1] * N / b));

        if (N_ITER>N_START) {

```

```

        F_PROFILE(f0, tf0, i) = vel[xn][yn];
    }
    else {
        F_PROFILE(f0, tf0, i) = 0;
    }
}
end_f_loop(f0, tf0)
}
DEFINE_PROFILE(periodic_v, tf0, i)
{
    Domain *d;
    Thread *tf1, *t;
    cell_t c;
    real x0[ND_ND], x1[ND_ND], vel[N][N];
    face_t f0, f1;
    int xn, yn;
    real a, b;

    a = RP_Get_Input_Parameter(ina) * 2; //get channel dimensions from parameter list
    b = RP_Get_Input_Parameter(inb) * 2;
    d = Get_Domain(1);

    if (tf0 == Lookup_Thread(d, IN_ID_DRY)) //match up dry inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_DRY);
    }
    else if (tf0 == Lookup_Thread(d, IN_ID_WET)) //match up wet inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_WET);
    }

    begin_f_loop(f1, tf1) //store the velocity field of the inlet in the array
    {
        if (N_ITER > 0) { //only do this after first iteration
            F_CENTROID(x1, f1, tf1);
            xn = floor(fabs(x1[0] * N / a));
            yn = floor(fabs(x1[1] * N / b));
            vel[xn][yn] = F_V(f1, tf1);

        }
    }
    end_f_loop(f1, tf1)

    begin_f_loop(f0, tf0) //apply the velocity field of the outlet to the inlet
    {
        F_CENTROID(x0, f0, tf0);
        xn = floor(fabs(x0[0] * N / a));
        yn = floor(fabs(x0[1] * N / b));

        if (N_ITER > N_START) {
            F_PROFILE(f0, tf0, i) = vel[xn][yn];
        }
        else {
            F_PROFILE(f0, tf0, i) = 0;
        }
    }
}

```

```

    }
    end_f_loop(f0, tf0)
}
DEFINE_PROFILE(periodic_w, tf0, i)
{
    Domain *d;
    Thread *tf1, *t;
    cell_t c;
    real x0[ND_ND], x1[ND_ND], vel[N][N];
    face_t f0, f1;
    int xn, yn;
    real a, b, k=1;

    a = RP_Get_Input_Parameter(ina) * 2; //get channel dimensions from parameter list
    b = RP_Get_Input_Parameter(inb) * 2;
    d = Get_Domain(1);

    if (tf0 == Lookup_Thread(d, IN_ID_DRY)) //match up dry inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_DRY);
    }
    else if (tf0 == Lookup_Thread(d, IN_ID_WET)) //match up wet inlet and outlet
    {
        tf1 = Lookup_Thread(d, OUT_ID_WET);
        k = -1;
    }

    begin_f_loop(f1, tf1) //store the velocity field of the inlet in the array
    {
        if (N_ITER > 0) { //only do this after first iteration
            F_CENTROID(x1, f1, tf1);
            xn = floor(fabs(x1[0] * N / a));
            yn = floor(fabs(x1[1] * N / b));
            vel[xn][yn] = F_W(f1, tf1);
        }
    }
    end_f_loop(f1, tf1)

    begin_f_loop(f0, tf0) //apply the velocity field of the outlet to the inlet
    {
        F_CENTROID(x0, f0, tf0);
        xn = floor(fabs(x0[0] * N / a));
        yn = floor(fabs(x0[1] * N / b));

        if (vel[xn][yn] == 0) {
            F_PROFILE(f0, tf0, i) = k;
        }
        else if (N_ITER > N_START) {
            F_PROFILE(f0, tf0, i) = vel[xn][yn];
        }
        else {
            F_PROFILE(f0, tf0, i) = k;
        }
    }
}

```

```

        end_f_loop(f0, tf0)
    }

DEFINE_PROFILE(periodic_species, tf0, i)
{
    Domain *d;
    Thread *tf1, *t;
    cell_t c;
    real x0[ND_ND], x1[ND_ND], area[ND_ND], Y[N][N];
    face_t f0, f1;
    int xn, yn;
    real a, b;
    real H_sum = 0, H_norm = 0, H_b0 = 0, X_b0 = 0.005, H_b1 = 0;
    real Hf_sum = 0, Hf_norm = 0, H_w0 = 0, H_w1 = 0;

    a = RP_Get_Input_Parameter(ina) * 2; //get channel dimensions from parameter list
    b = RP_Get_Input_Parameter(inb) * 2;

    d = Get_Domain(1);

    if (N_ITER > N_START)
    {
        if (tf0 == Lookup_Thread(d, IN_ID_DRY)) //match up dry inlet and outlet
        {
            tf1 = Lookup_Thread(d, OUT_ID_DRY);
            X_b0 = X_dry;
        }
        else if (tf0 == Lookup_Thread(d, IN_ID_WET)) //match up wet inlet and outlet
        {
            tf1 = Lookup_Thread(d, OUT_ID_WET);
            X_b0 = X_wet;
        }

        //calculate and store the species field of the outlet
        begin_f_loop(f1, tf1)
        {
            if (N_ITER > 0) { //only do this after first iteration
                F_CENTROID(x1, f1, tf1); //get the position of the cell
                xn = floor(fabs(x1[0] * N / a)); //x bin
                yn = floor(fabs(x1[1] * N / b)); //y bin
                F_AREA(area, f1, tf1);
                Y[xn][yn] = F_YI(f1, tf1, 0)*F_R(f1, tf1); //store the species
                concentraion on the outlet

                H_sum += fabs(F_W(f1, tf1))*NV_MAG(area)*F_R(f1, tf1)*F_R(f1,
                tf1)*F_YI(f1, tf1, 0); //integrate to get bulk concentration
                H_norm += fabs(F_W(f1, tf1))*NV_MAG(area)*F_R(f1, tf1);
            } //normalization factor for bulk concentration

            c = F_C0(f1, tf1); //get cell adjacent to face
            t = THREAD_T0(tf1); // get thread for above cell
        }
    }
}

```

```

        if (fabs(C_UDMI(c,t,3)) == 1) //check if the cell is adjacent to the
membrane (+1 for dry, -1 for humid)
        {
            Hf_sum += C_VOLUME(c,t)*C_R(c, t)*C_UDMI(c,t,4); //integrate
to get wall h20 concentration. the wall mass fraction has been stored in UDMI 4 by the
fick's law macro
            Hf_norm += C_VOLUME(c, t); //normalization factor
        }
    }
end_f_loop(f1, tf1)

H_b1 = H_sum / H_norm; //compute bulk and wall average at the outlet
H_w1 = Hf_sum / Hf_norm;

H_sum = 0; //reset the running averages
H_norm = 0;
Hf_sum = 0;
Hf_norm = 0;

//calculate inlet bulk and wall fields
begin_f_loop(f0, tf0)
{
    F_CENTROID(x0, f0, tf0); //get the position of the cell
    F_AREA(area, f0, tf0);
    H_sum += fabs(F_W(f0, tf0))*NV_MAG(area)*F_R(f0, tf0)*F_R(f0, tf0)*F_YI(f0,
tf0, 0); //integrate to get bulk concentration
    H_norm += fabs(F_W(f0, tf0))*NV_MAG(area)*F_R(f0, tf0); //normalization
factor for bulk concentration

    c = F_C0(f0, tf0); //get cell adjacent to face
    t = THREAD_T0(tf0); // get thread for above cell

    if (fabs(C_UDMI(c, t, 3)) == 1) //check if the cell is adjacent to the
membrane (+1 for dry, -1 for humid)
    {
        Hf_sum += C_VOLUME(c, t)*C_R(c, t)*C_UDMI(c, t, 4); //integrate to
get wall h20 concentration. the wall mass fraction has been stored in UDMI 4 by the
fick's law macro
        Hf_norm += C_VOLUME(c, t); //normalization factor
    }
}
end_f_loop(f0, tf0)

//compute bulk and wall concentrations at the inlet
H_w0 = Hf_sum / Hf_norm;
H_b0 = X_b0*Rho; //calculate bulk inlet concentration using inlet humidity ratio
and dry air density

//apply the field of the outlet to the inlet
begin_f_loop(f0, tf0)
{
    F_CENTROID(x0, f0, tf0);
    xn = floor(fabs(x0[0] * N / a));
    yn = floor(fabs(x0[1] * N / b));

```

```

        F_PROFILE(f0, tf0, i) = ((Y[xn][yn] - H_w1) / (H_b1 - H_w1)*(H_b0 - H_w0) +
H_w0) / F_R(f0, tf0);
    }
    end_f_loop(f0, tf0)
}

    else
    {
        begin_f_loop(f0, tf0)
        {
            F_PROFILE(f0, tf0, i) = X_b0;
        }
        end_f_loop(f0, tf0)
    }

    //Message("Hb = %g, Hw = %g, Hb0 = %g, Hw0 = %g \n", H_b1, H_w1, H_b0, H_w0);
}

DEFINE_PROFILE(periodic_temperature, tf0, i)
{
    Domain *d;
    Thread *tf1, *t;
    cell_t c;
    real x0[ND_ND], x1[ND_ND], area[ND_ND], Y[N][N];
    face_t f0, f1;
    int xn, yn;
    real a, b;
    real T_sum = 0, T_norm = 0, T_norm_1 = 0, T_b0 = 300, T_b1 = 0, T_b_inlet = 0;
    real Tf_sum = 0, Tf_norm = 0, T_w0 = 0, T_w1 = 0;

    a = RP_Get_Input_Parameter(ina) * 2; //get channel dimensions from parameter list
    b = RP_Get_Input_Parameter(inb) * 2;

    d = Get_Domain(1);

    if (N_ITER>N_START*3)
    {
        if (tf0 == Lookup_Thread(d, IN_ID_DRY)) //match up dry inlet and outlet
        {
            tf1 = Lookup_Thread(d, OUT_ID_DRY);
            T_b0 = T_dry;
        }
        else if (tf0 == Lookup_Thread(d, IN_ID_WET)) //match up wet inlet and
outlet
        {
            tf1 = Lookup_Thread(d, OUT_ID_WET);
            T_b0 = T_wet;
        }

        //calculate and store the temperature field of the outlet
        begin_f_loop(f1, tf1)
        {

```

```

        F_CENTROID(x1, f1, tf1); //get the position of the cell
        xn = floor(fabs(x1[0] * N / a)); //x bin
        yn = floor(fabs(x1[1] * N / b)); //y bin
        F_AREA(area, f1, tf1);

        T_sum += fabs(F_W(f1, tf1))*NV_MAG(area)*F_R(f1, tf1)*F_T(f1,
tf1); //integrate to get bulk temperature
        T_norm_1 += fabs(F_W(f1, tf1))*NV_MAG(area)*F_R(f1, tf1);
//normalization factor for bulk concentration

        Y[xn][yn] = F_T(f1, tf1); //store the temperature of the
outlet

        c = F_C0(f1, tf1); //get cell adjacent to face
        t = THREAD_T0(tf1); // get thread for cell adjacent to face

        if (fabs(C_UDMI(c, t, 3)) == 1) //check if the cell is
adjacent to the membrane (+1 for dry, -1 for humid)
        {
            Tf_sum += C_VOLUME(c, t)*C_UDMI(c, t, 5); //integrate
to get wall temperature.
            Tf_norm += C_VOLUME(c, t); //normalization factor
        }
    }
end_f_loop(f1, tf1)

T_b1 = T_sum / T_norm_1; //compute bulk and wall average at the outlet
T_w1 = Tf_sum / Tf_norm;

T_sum = 0; //reset the running averages
Tf_sum = 0;
Tf_norm = 0;

//calculate inlet bulk and wall fields
begin_f_loop(f0, tf0)
{
    F_CENTROID(x0, f0, tf0); //get the position of the cell
    F_AREA(area, f0, tf0);
    T_sum += fabs(F_W(f0, tf0))*NV_MAG(area)*F_R(f0, tf0)*F_T(f0, tf0);
//integrate to get bulk concentration
    T_norm += fabs(F_W(f0, tf0))*NV_MAG(area)*F_R(f0, tf0);
//normalization factor for bulk concentration

    c = F_C0(f0, tf0); //get cell adjacent to face
    t = THREAD_T0(tf0); // get thread for above cell

    if (fabs(C_UDMI(c, t, 3)) == 1) //check if the cell is adjacent to
the membrane (+1 for dry, -1 for humid)
    {
        Tf_sum += C_VOLUME(c, t)*C_UDMI(c, t, 5); //integrate to get
wall temperature.
        Tf_norm += C_VOLUME(c, t); //normalization factor
    }
}
end_f_loop(f0, tf0)

```

```

//compute bulk and wall temperatures at the inlet
T_w0 = Tf_sum / Tf_norm;
T_b_inlet = T_sum / T_norm;

//apply the field of the outlet to the inlet
begin_f_loop(f0, tf0)
{
    F_CENTROID(x0, f0, tf0);
    xn = floor(fabs(x0[0] * N / a));
    yn = floor(fabs(x0[1] * N / b));

    F_PROFILE(f0, tf0, i) = ((Y[xn][yn] - T_w1) / (T_b1 - T_w1)*(T_b0 -
T_w0) + T_w0); //define inlet profile attempting to take into account changes in density
}
end_f_loop(f0, tf0)
}
else
{
    begin_f_loop(f0, tf0)
    {
        F_PROFILE(f0, tf0, i) = T_b0;
    }
    end_f_loop(f0, tf0)
}

//Message("Tb = %g, Tw = %g, Tb0 = %g, Tw0 = %g \n", T_b1, T_w1, T_b_inlet, T_w0);
}

```

A.3 Rectangular Channel Fully Developed Velocity Profile

```
#include "udf.h"
#define pi 3.1415926535

DEFINE_PROFILE(square_profile, t, i)
{
    real r[ND_ND]; //holds the position of the face cells
    real y, x; //coordinates of the face cell
    face_t f;
    real a = RP_Get_Input_Parameter("real-2"); //half the inlet width from parameter
list
    real b = RP_Get_Input_Parameter("real-3"); //half the inlet heigh from parameter
list

    begin_f_loop(f, t)
    {
        F_CENTROID(r, f, t); //get face position
        y = r[1]; //y coordinate
        x = r[0]; //x coordinate
        real prof = 0; //initialise the profile vale for the face
        int m = 1;

        if (y < 0)
        {
            m = -1; //this modifier allows for use in teh dry zone which is
below which is shifted by 2b
        }

        for (int n = 1; n < 100; n = n + 2)
        {
            prof += 1/pow(n,3)*pow(-1, (n - 1) / 2)*(1 - cosh(n * pi*(y - m*b) /
2 / a) / cosh(n * pi*b / 2 / a))*cos(n * pi*(x-a) / 2 / a);
        }
        F_PROFILE(f, t, i) = prof;
    }
    end_f_loop(f,t)
}
```