Adsorptive Thermochemical Energy Storage and it's Impacts on Space Heating

Griffin Worboy

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Applied Science degree in Chemical Engineering

Department of Chemical and Biological Engineering
Faculty of Engineering
University of Ottawa

Abstract

Over the past 200 years, human activities have increased the carbon dioxide content in the atmosphere by 50% [1]. Currently, this corresponds to a 1.1ºC rise above 1850-1900 average global temperatures in 2011-2020 [2]. If the global average temperature continues to rise past 1.5ºC, the human population will face unprecedented climate-related risks and weather events [3]. In 2020, over half (53.5%) of the energy consumed by the space heating sector in Canada was supplied by fossil fuels [4]. To help reduce fossil fuel emissions and reach 2050 net zero goals a reduction in greenhouse gas emissions is required. To help address the problem, a heat storage technology is developed to store thermal energy for use in space heating applications.

The focus on this thesis is on the modelling and simulation of the thermal energy storage technology developed based on exothermic adsorption process and its validation with experimental results. The effect of column volume and length to diameter ratios on the performance of the system have been experimentally studied and simulated. The simulation was carried out by using gPROMS (general Process Modelling System) software. It incorporates single dimension heat and mass transfer equations where the radial changes along the column are neglected. The simulation uses the Guggenheim-Anderson-de Boer (GAB) isotherm model to describe the moisture equilibrium behaviour with the adsorbent sample. The modelling results show good agreement with the experimental data. Effects of length to diameter ratios show an increasing trend in energy storage density with decreasing length to diameter ratios. The fitted model was also coded in Fortran and implemented in a TRNSYS (Transient System Simulation Tool) simulation to study the performance of this energy storage technology in home heating scenarios. Using the results from the TRNSYS simulation the economic viability of this technology is explored.

Table of Contents

List of Figures

List of Tables

1. General Introduction

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2024/09/25

1.1. Introduction

In 2020, over half (53.5%) of the energy consumed by the space heating sector in Canada was supplied by fossil fuels [4]. This corresponds to 28.8 Mt of carbon dioxide (CO_2) released just in this sector in Canada in 2020 alone [4]. This additional CO_2 released continues to add to the already elevated CO_2 levels in our atmosphere from emissions over the past 200 years. Currently, the CO_2 concentration in the atmosphere has risen by 50% over the past 200 years [4]. This corresponds to a 1.1°C rise above 1850-1900 average global temperatures in 2011-2020 [2]. An average temperature rise past 1.5 °C will force the population to face unprecedented climate-related risks and weather events [3]. Therefore, a reduction of emissions is required to help stop/slow climate change. Currently, renewable energy sources are one of the most attractive options to help reduce climate change. However, because renewable energy is susceptible to environmental conditions such as wind speed, temperature, and solar radiation, energy storage devices are needed to store energy during peak supply times to fill the gaps during low energy supply times.

In this thesis, adsorptive thermochemical energy storage is explored as a potential technology to store thermal energy during peak supply times for later discharging in low energy supply times. Thermochemical energy storage is an attractive option to develop a thermal battery. Unlike sensible and latent heat storage, thermochemical energy storage is not heavily dependent on the insulation used to store heat for long periods. Because thermochemical energy storage utilizes a reversible chemical process to store heat as long as the components are kept separate, the energy is effectively stored. Because of its high storage efficiency, adsorptive thermochemical energy storage is a hot topic amongst researchers today. There are many papers published evaluating and discussing the energy storage density of many different solid adsorbents. Energy storage density (ESD) is used to describe the performance of the solid adsorbent indicating how much heat it is able to supply per m^3 or kg of adsorbent. Yang et al. (2023) evaluated the energy storage density of different salt hydrate adsorption materials showing a wide range of energy storage densities from 0.378 kWh/kg to 0.08 kWh/kg [5]. The highest ESD evaluated by Yang et al (2023) was magnesium sulphate heptahydrate and was evaluated to be 0.378 kWh/kg while the lowest one evaluated was strontium chloride hexahydrate with an ESD of 0.08 kWh/kg [5]. Yang et al (2024) investigated a silica gel-based adsorbent impregnated with magnesium sulphate and its potential to supply building heating [6]. They evaluated the ESD of this material to be 0.198 kWh/kg with

a maximum output power of 89.17 kW [6]. They also found that increasing the desorption temperature up to 110°C had a positive impact on the energy storage density allowing for more water to be desorbed and more energy to be recovered giving rise to a higher energy storage density. This finding is corroborated by Strong et al (2022) who studied and optimized the operating conditions of an open bulk-scale silica gel/water vapour adsorption energy storage system [7]. They found that past a regeneration temperature of 120°C no appreciable change in the energy storage density was found meaning 120°C is optimal to regenerate the studied adsorbents [7]. They also found a dependency on the inlet relative humidity during the adsorption phase, showing higher energy storage densities with increasing inlet relative humidity. This is because a higher inlet relative humidity allows for more moisture adsorption and therefore a higher heat recovery. Also studied by Strong et al (2022) was the effect of flowrate. Authors found that increasing the flowrate also had a positive effect on the reported energy storage density up to a point. They found that the energy storage density increased with increasing flowrate and then subsequently plateaued and no more increase in the energy storage density was observed. This was attributed to a trade off between increased velocity and pressure drop and decreased residence times [7]. Higher velocity will allow for a higher convective mass transfer coefficient enabling more water vapour adsorption [7]. However, these faster velocities also increased the heat transfer coefficient allowing for more heat to be transferred outside of the adsorption column [7]. Additionally, higher flowrates increased the pressure drop, and ultimately the pressure at the inlet of the column [7]. A higher pressure at the inlet of the column allowed for a higher absolute humidity and thus a higher driving force for mass transfer. On the other hand, higher velocities also decreased the residence time of water vapour in the column allowing less time for water to be adsorbed [7]. Based on these findings, an optimal inlet flowrate and relative humidity was chosen to be 80% relative humidity at 24 SLPM for this study. Shown in Figure 1.1 to Figure 1.3 are the effects of flow rate, inlet relative humidity, and regeneration temperature. Banaei et al (2021) studied the challenges of using adsorption as a heat storage medium in the residential sector. They found that increasing the volume of the adsorbent bed increased the energy storage density linearly [8]. However, in residential sectors where space is limited, increasing column volume is not desired. Banaei et al 2021 outlined that the main challenges faced by this technology were space restrictions and affordability stressing the importance of development into

efficient, compact adsorption column designs that increases water vapour-adsorbent contact area while reducing the size of the system [8].

While there are many papers discussing different adsorbents and their energy storage densities, little information can be found in regard to adsorption column geometry and its effects on reported energy storage densities. Additionally, to assess the viability of this technology in real home heating scenarios a simulation into this technology could give valuable insights to the potential of this technology. Therefore, the focus on this thesis is assessing the relationship between column geometry and reported energy storage densities by assessing column length to diameter ratios and column volume. In addition to assessing the effect of column geometry simulations of this technology in home heating scenarios can provide valuable insights to the performance of this technology in real home heating scenarios and is also studied using TRNSYS simulation software.

The work of this thesis was done in collaboration with a couple different people. Leonardo Rubalcava and Jeremy Simanjuntak assisted me with performing and analysing experimental results while studying the effects of column geometry. Dr. Cynthia Cruickshank from the Department of Mechanical Engineering at Carleton University provided support and insights into modelling this technology in TRNSYS simulation studio. Dr. F. Handan Tezel supported me in performing computational modelling as well as analysis of raw data collected in her lab at the University of Ottawa. Dr. Alexandra Mallett from the School of Public Policy and Administration at Carleton University provided different outlooks on the project from a policy perspective to aid in conclusions from different points of view. The work that I conducted on this project includes debugging and revitalizing the experimental setup used to collect experimental data, design of the adsorption columns used to assess column geometry, collect and analyse experimental data, develop and implement a mathematical model to model moisture vapour adsorption in gPROMS simulation software, coding the fitted mathematical model in Fortran for use in a TRNSYS simulation, developed a TRNSYS simulation to model the thermal battery in home heating scenarios, and the writeup of this thesis. This thesis is constructed with 4 chapters, Chapter 1 is a general introduction to the thesis, Chapter 2 is a paper on effects that column geometry has on reported energy storage density as well as modelling of this process, Chapter 3 summarizes modelling this thermal battery in TRNSYS and its impacts on home heating, and Chapter 4 is general conclusions for the thesis.

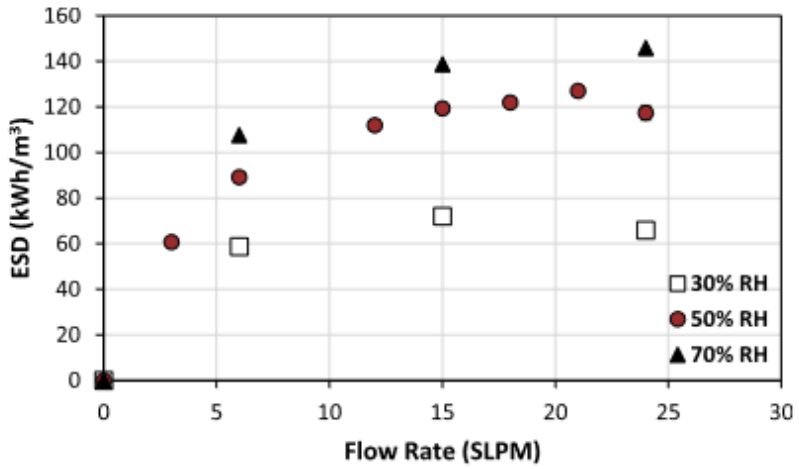


Figure 1.1: Effect of Flow Rate on Energy Storage Density (ESD) [7]

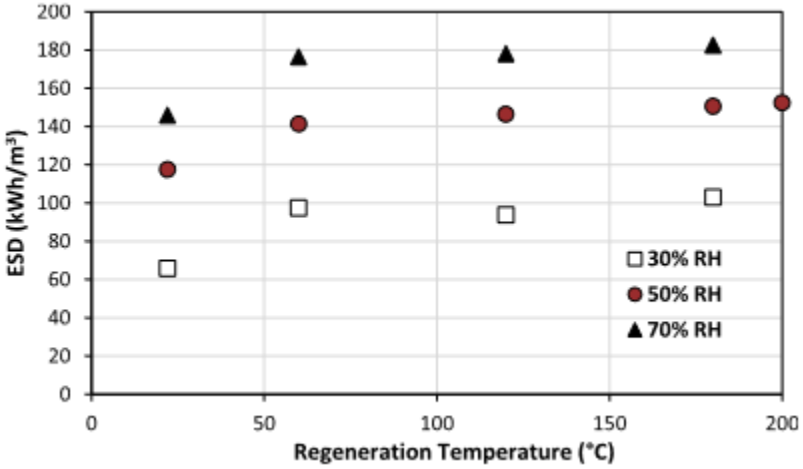


Figure 1.2: Effect of Regeneration Temperature on Energy Storage Density (ESD) [7]

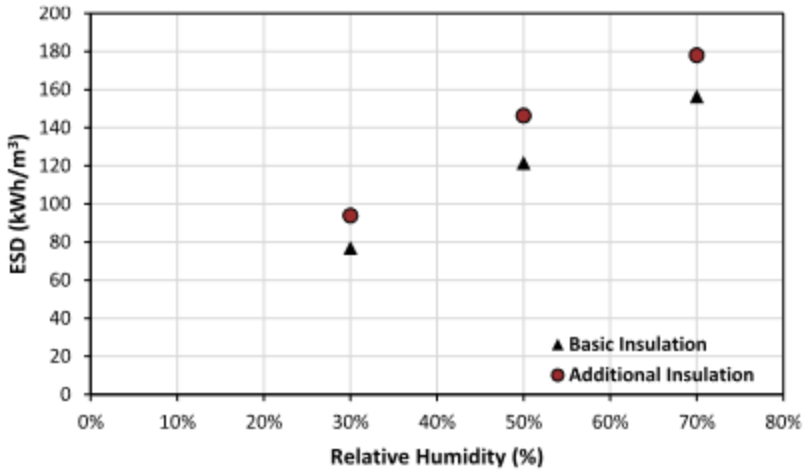


Figure 1.3: Effect of Relative Humidity on Energy Storage Density (ESD) [7]

1.2. References

[1] NASA, "Carbon Dioxide," NASA, February 2024. [Online]. Available: https://climate.nasa.gov/vital-signs/carbon-dioxide/?intent=121. [Accessed 04 04 2024].

[2] Core Writing Team, H. Lee and J. Romero, "Climate Change 2023: Synthesis Report," IPCC, Geneva, Switzerland, 2023.

[3] V. Masson-Delmotte, P. Zhai, H. Portner, D. Roberts, J. Skea, P. Shukla, A. Pirani, W. Moufouma-Okia, C. Peann, R. Pidcock, S. Connors, J. Matthews, Y. Chen, X. Zhou, M. Gomis, E. Lonnoy, T. Maycock, M. Tignor and T. Waterfields, "IPCC, 2018: Summary for Policymakers," Cambridge University Press, New York, 2018.

[4] Natural Resources Canada, "Natural Resources Canada," Government of Canada, 2020. [Online]. Available: https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/showTable.cfm?type=CP&sector=res&juris=ca&rn=7&year=2020&page=3. [Accessed 02 04 2024].

[5] H. Yang, C. Wang, L. Tong, S. Yin, L. Wang and Y. Ding, "Salt Hydrate Adsorption Material-Based Thermochemical Energy Storage for Space Heating Application: A Review," *Energies,* vol. 16, no. 6, p. 2875, 2023.

[6] H. Yang, C. Wang, Y. Zhang, B. Nie, L. Tong, S. Yin, L. Wang and Y. Ding, "Experimental Investigation of thermochemical energy storage system based on MgSO4-silica gel for building heating: adsorption/desorption performance testing and system optimization," *Energy Conversion and Management,* vol. 301, p. 118000, 2024.

[7] C. Strong, Y. Carrier and F. H. Tezel, "Experimental optimization of operating conditions for an open bulk-scale silica gel/water vapour adsorption energy storage system," *Applied Energy,* vol. 312, p. 118533, 2022.

[8] A. Banaei and A. Zanj, "A Review on the Challenges of Using Zeoloite 13X as Heat Storage Systems for the Residential Sector," *energies,* vol. 14, no. 23, p. 8062, 2021.

2. Modeling and Simulation of Adsorptive Thermal Energy Storage – Effect of Column Volume and Length to Diameter Ratio

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2024/09/25

Abstract

In 2020, over half (53.5%) of the energy consumed by the space heating sector in Canada was supplied by fossil fuels [1]. To help reduce fossil fuel emissions and reach 2050 net zero goals a reduction in greenhouse gas emissions is required. To help reduce emissions in the residential space heating sector adsorptive thermochemical energy storage is explored as a potential technology. In this paper, the effect that column geometries have on reported energy storage densities is explored, namely, length to diameter ratio and column volume. In addition to studying the effect of column geometries, a mathematical model is developed and fitted using gPROMS simulation software. The model shows good agreement with the experimental data set predicting the energy storage density with 85-92% accuracy. Effects observed from studying column geometry showed a decreasing trend in reported energy storage density with increasing length to diameter ratios, and an increasing trend of energy storage densities with increasing column volume.

Nomenclature

Symbol	Name	Units
a	Surface area to volume ratio	m^{-1}
$\boldsymbol{C}$	Gas phase concentration	*mol water/total mol*
$\hat{\boldsymbol{c}}_{\boldsymbol{f}}$	Fluid heat capacity	*J/kg moist air/K*
$\tilde{\boldsymbol{c}}_{\boldsymbol{f}}$	*Fluid heat capacity*	*J/mol moist air/K*
$\tilde{\boldsymbol{c}}_{\boldsymbol{i}}$	*Heat capacity of component i*	*J/mol i/K*
$\hat{\boldsymbol{c}}_{\boldsymbol{s}}$	*Solid heat capacity*	*J/kg solid/K*
$\boldsymbol{D}_{\boldsymbol{L}}$	*Axial diffusivity*	m^2/s
$\boldsymbol{D}_{\boldsymbol{m}}$	*Molecular diffusivity*	m^2/s
$\boldsymbol{d}_{\boldsymbol{p}}$	Particle diameter	*m*
h	Solid fluid heat transfer coefficient	$W/m^2/K$
$\boldsymbol{k}$	Boltzman constant	*J/K*
$\boldsymbol{k}_{\boldsymbol{f}}$	Film mass transfer coefficient	*m/s*
$\boldsymbol{k}_{\boldsymbol{g}}$	Fluid thermal conductivity	*W/m/K*
$\boldsymbol{k}_{\boldsymbol{i}}$	Thermal conductivity of component i	*W/m/K*
$\boldsymbol{k}_{\boldsymbol{m}}$	Overall mass transfer coefficient	*m/s*
$\boldsymbol{k}_{\boldsymbol{s}}$	Solid phase mass transfer coefficient	*m/s*
m	Slope of isotherm	-
$\boldsymbol{MW}_{\boldsymbol{a}}$	Molecular weight air	*kg/mol*
$\boldsymbol{MW}_{\boldsymbol{w}}$	Molecular weight water	*kg/mol*
$\boldsymbol{Nu}_{\boldsymbol{0}}$	Single particle Nusselt number	-
$\boldsymbol{Nu}_{\boldsymbol{p}}$	Particle bed Nusselt number	-
P	Total pressure	*Pa*
$\boldsymbol{p}_{\boldsymbol{H_2O}}$	Partial pressure of water	*Pa*
Pr	Prandtl number	-
$\overline{\boldsymbol{q}}$	Amount adsorbed	*wol /kg*
$\boldsymbol{q}^*$	Equilibrium amount adsorbed	*mol/kg*
R	Ideal gas constant	*J/mol/K*
Re	Reynolds number	-
Sc	Schmidt number	-
Sh	Sherwood number	-
t	Time	*s*
T	Temperature	*K*
$\boldsymbol{T}^*$	Dimensionless parameter	-
$\boldsymbol{T}_{\boldsymbol{f}}$	Fluid temperature	*K*
$\boldsymbol{T}_{\boldsymbol{s}}$	Solid temperature	*K*
v	Column interstitial velocity	*m/s*

z	Position in column	m
Z	Compressibility factor	-
$\Delta\widetilde{\boldsymbol{H}}$	Heat of adsorption	J/mol
$\boldsymbol{\varepsilon}$	Void fraction	-
$\frac{\boldsymbol{\varepsilon_a}}{\boldsymbol{k}}$	Lennard-Jones force constant for air	K
$\frac{\boldsymbol{\varepsilon_{aw}}}{\boldsymbol{k}}$	Lennard-Jones force constant for air water mixture	K
$\frac{\boldsymbol{\varepsilon_w}}{\boldsymbol{k}}$	Lennard-Jones force constant for water	K
$\boldsymbol{\mu_i}$	Viscosity of component i	$Pa\ s$
$\boldsymbol{\mu_{MA}}$	Viscosity of moist air	$Pa\ s$
$\widetilde{\boldsymbol{\rho}}_{\boldsymbol{f}}$	Fluid density	$mol\ moist\ air/m^3$
$\widehat{\boldsymbol{\rho}}_{\boldsymbol{f}}$	Fluid density	$kg\ moist\ air/m^3$
$\widehat{\boldsymbol{\rho}}_{\boldsymbol{s}}$	Solid density	$Kg\ solid/m^3$
$\boldsymbol{\sigma_a}$	Lennard-Jones potential air	$Å$
$\boldsymbol{\sigma_{aw}}$	Collision diameter	$Å$
$\boldsymbol{\sigma_w}$	Lennard-Jones potential water vapour	$Å$
$\boldsymbol{\Omega_{aw}}$	Collision integral	-

Abbreviations

Abbreviation	Meaning
$CaCl_2$	Calcium chloride
CO_2	Carbon dioxide
ESD	Energy storage density
GAB	Gugenheim Anderson De-Boer
gPROMS	General process modelling system
H_2O	Water
L/D	Length to Diameter
MFC	Mass flow controllers
NRCan	National Resources Canada
PFD	Process flow diagram
RH	Relative humidity
SG	Silica Gel
SLPM	Standard liters per minute
TES	Thermal energy storage
V	Volume

2.1. Introduction/Background

Over the past 200 years, human activities have increased the carbon dioxide content in the atmosphere by 50% [2]. Currently, this corresponds to a 1.1ºC rise above 1850-1900 average global temperatures in 2011-2020 [3]. If the global average temperature continues to rise past 1.5ºC, the human population will face unprecedented climate-related risks and weather events [4]. In 2020, over half (53.5%) of the energy consumed by the space heating sector in Canada was supplied by fossil fuels [1]. To help reduce fossil fuel emissions and reach 2050 net zero goals a reduction in greenhouse gas emissions is required. To do this, the world's energy systems are starting to "decentralize, decarbonize, and democratize" moving away from fossil fuels [5]. One of the most attractive options to do this is the use of renewable energy sources. However, because energy generation from renewable sources are susceptible to environmental conditions such as wind, solar radiation, and temperature, energy storage technologies must be developed to help with low energy supply times and ensure a reliable supply of energy to end users. To help address the problem, a thermal energy storage technology is explored for use in space heating applications reducing the need for fossil fuels and/or electricity demand on the grid. Currently, there are many types of thermal energy storage technologies such as latent heat storage, sensible heat storage, and thermochemical energy storage. Sensible and latent heat storage systems use the sensible and latent heat of the energy storage material to store the energy whereas thermochemical energy storage systems utilize the heat of a reversible reaction/process to store energy. The main advantage of thermochemical energy storage compared to sensible and latent heat storage is the storage efficiency. With latent and sensible heat storage, the energy stored can be susceptible to losses and inefficiencies if not properly insulated thus reducing the amount of heat that can be stored. On the other hand, thermochemical energy storage does not pose these same issues. The main benefit of thermochemical energy storage is the fact that heat is stored in reversible chemical processes. If the components are kept separate the heat is effectively stored without any losses. The technology explored in this paper is thermochemical energy storage.

The main mechanism behind this technology is the exothermic nature of adsorption. Adsorption is a mass transfer process in which matter from one phase adsorbs/sticks to a solid in another phase. This process is generally exothermic in nature, that is, when an adsorbate (matter that sticks to the solid) changes phase from the fluid phase to the adsorbed phase heat is released. This released heat is the principle of operation of this technology. In this paper the interaction is between water vapor and a solid adsorbent. A solid adsorbent is packed in an adsorption column and humid air is allowed to flow over this solid adsorbent resulting in water adsorption and a production of dry hot air at the column outlet that can be used in space heating applications. The focus of this chapter is to model and predict the water vapor adsorption process using mathematical models as well as study the effect that column geometry (length to diameter ratios) and column volume has on the reported results.

2.2. Experimental Setup

Outlined in Figure 2.1 is the process flow diagram (PFD) of the experimental setup used to collect data. Air is sourced from the building supply and enters at the left of the diagram. The building supply air enters the system at a pressure of 30.7psia. Two AliCat mass flow controllers (MFC) control the inlet flow of the dry and wet air streams to achieve a desired relative humidity at the column inlet. The pressure at the outlet of the MFC's can range between 18-25 psia. The dry air stream first enters R-100 which is an adsorptive air dehumidification column before mixing with the wet air stream. The adsorbent used to dry the dry air stream is drierite. The wet air stream bubbles through V-101 which is a humid air generation tank that is assisted by ultrasonic foggers to aid in generating humid air at higher flowrates. Immediately following the humidifying tank, the wet air stream enters V-102, a water knockout vessel. The purpose of this vessel is to remove excess water from the stream to ensure no condensation and/or excess water buildup down stream of the humidifying vessel. Once excess water is knocked out in V-102 the wet air stream mixes with the dry air stream in mixing chamber V-103 to achieve the desired relative humidity. V-103 is fitted with temperature, relative humidity sensors as well as a pressure gauge. These sensors allow the calculation of water concentration in the column inlet stream for modeling purposes. Following the mixing chamber the flow is either sent through the column or through the bypass for flow development depending on the progress of the experiment. The flow direction is controlled by a 3-way diverting valve hooked up to the Lab-View software. The adsorption column R-101 is outfitted with fiberglass insulation, column skin temperature probes, as well as inlet and outlet temperature probes to ensure accurate measurement of inlet and outlet temperature differences. In addition to the sensors, R-101 is also equipped with 2 manual valves (VLV-103 and VLV-104) to ensure isolation from the rest of the system after regeneration is complete and cooling us underway. Finally, the flow enters a mock "room" V-104. V-104 is outfitted with temperature and relative humidity sensors to monitor the progress of the experiment before flow exhausts to the atmosphere.

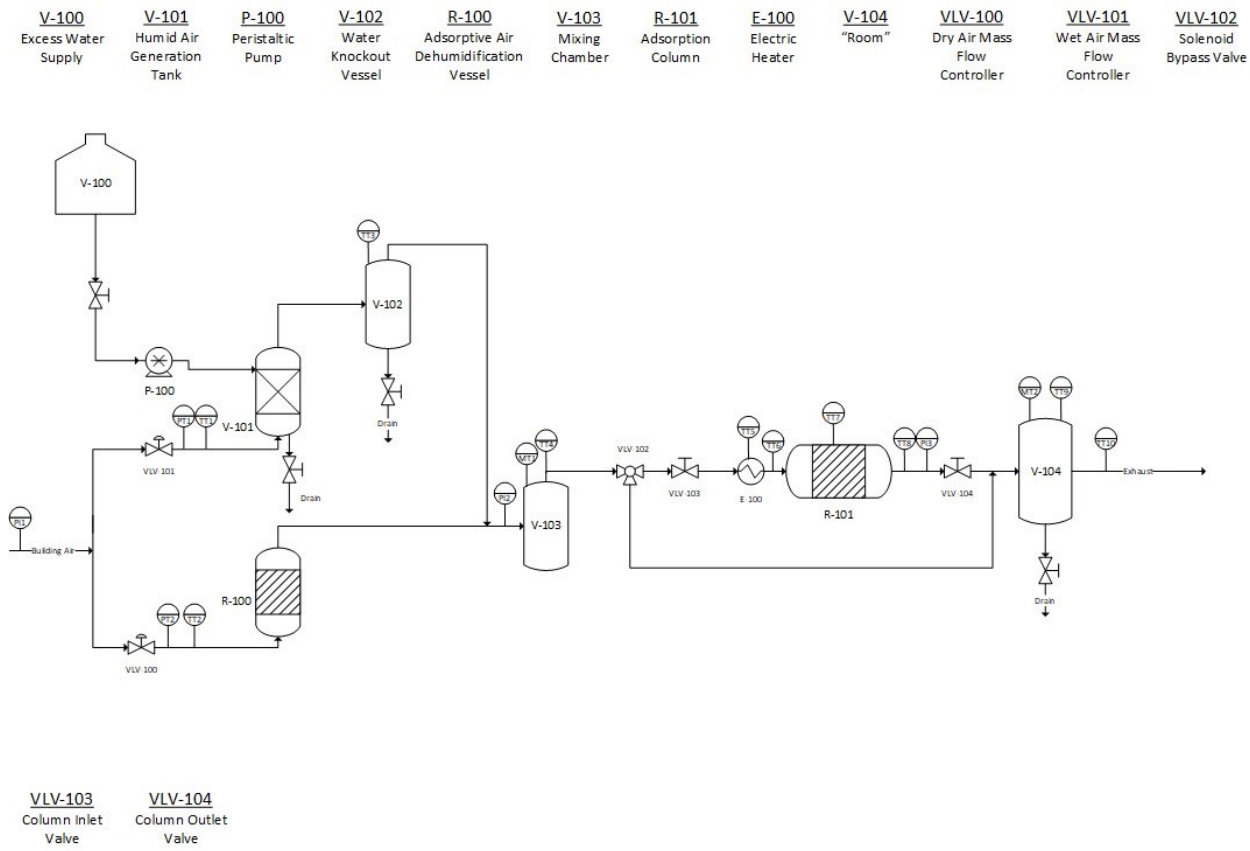


Figure 2.1: Process flow diagram of the thermal energy storage experimental setup.

2.3. Procedure

Before experiments can begin, the adsorption column must be packed with fresh adsorbent. First a clean empty column is weighed to get the mass before adsorbent is added. After it is weighed, the adsorption column is packed with fresh adsorbent ensuring even distribution of adsorbent throughout the column. Once the adsorbent is added, the column is weighed again to determine the mass of adsorbent added. Next regeneration is ready to begin. First, dry air flow is passed over the column until the outlet relative humidity stabilizes at a flowrate of 24SLPM. Once the outlet relative humidity stabilizes, high temperature regeneration can begin. The heater is then set at 120°C still under 24 SLPM of dry air flow and the outlet relative humidity is allowed to stabilize again for a couple hours before regeneration ends. Once regeneration is over, the flow is turned off and the column valves are closed to isolate the column from the rest of the system. Once the column is isolated it is allowed to cool overnight. Next, adsorption is ready to begin. Before moist air flow is sent through the column, the flow must be developed to ensure a steady flow of 80% relative humidity at 24 SLPM is available for the column. Once flow is developed through the column bypass, the column valves are opened and flow is sent through the column until thermal equilibrium is achieved at the column outlet. This adsorption desorption cycle is repeated 3 times to assess the cyclic stability of the adsorbent.

2.4. Model Outline

The model was developed using literature as a guide to form the mathematical expressions. The model was then fitted to experimental data using gPROMS modeling software. The code scripted in gPROMS is presented in Appendix A.1. Some assumptions were made in the development of the model. The first assumption is that air is not adsorbed and treated as a carrier gas. This assumption holds true for the tested adsorbent as the adsorbent readily and preferentially adsorbs moisture over air. The second assumption made is constant gas velocity. Because the moisture in the air is only in a very small portion of the overall gas stream, velocity changes with its adsorption would be negligible. The next assumption made was no pressure drop along the adsorbent bed. This assumption was implemented knowing the pressure drop along the adsorbent bed was minimal and negligible. The void fraction in the adsorbent bed was also treated as constant due to the almost perfect spherical pellets and tight packing of the adsorbent pellets. Another assumption made concerns the radial changes along the column. It is assumed that there are no concentration or temperature gradients along the radial axis of the column. This assumption is valid because there is a high enough superficial gas velocity that axial flow can dominate any special variations in concentration or temperature in the radial direction. Additionally, it is assumed that water adsorption is instantaneous on the pellet surface which is justified knowing the adsorption rate on the pellet surface is very high compared to the diffusion through the micropores [6]. Finally, the model considers the system to behave adiabatically, that is, no heat transfer from the column to the outside is considered. This assumption is justified due to the amount of insulation used in the experimental setup.

2.4.1. Mass Balance

The mass balance in the column was adapted from Cooney 1974 to include axial diffusion and is shown below in equation (2.1) [7,8].

$$-D_L \frac{\partial^2 C}{\partial z^2} + \frac{\partial C}{\partial t} + v \frac{\partial}{\partial z}(C) + \frac{\hat{\rho}_s}{\tilde{\rho}_f}\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{\partial \bar{q}}{\partial t} = 0 \qquad (2.1)$$

This mass balance in the bulk fluid phase includes axial diffusion, accumulation, convection, and mass transfer into the adsorbed phase due to adsorption to model the consumption of moisture in the bulk air stream. Parameters outlined in equation (2.1) include the axial diffusion coefficient (D_L), the interstitial velocity (v), the moisture concentration (C), solid density ($\hat{\rho}_s$), fluid density

($\tilde{\rho}_f$), column void fraction (ε), and the rate of moisture adsorption ($\frac{\partial \bar{q}}{\partial t}$). To note parameters denoted with a tild (Ã) indicate units of mole basis, parameters denoted with a circumflex (Â) indicate units of mass basis, and no accents denotes SI units. The boundary conditions used for the mass balance were derrived from Danckwerts boundary conditions and are presented below along with the initial conditions.

$$C(z)|_{t=0} = 0 \tag{2.2}$$

$$\bar{q}(z)|_{t=0} = 0 \tag{2.3}$$

$$\left.\frac{\partial C}{\partial z}\right|_{z=0} = \frac{\left(\frac{v}{\varepsilon}\right)(C(0) - C_{inlet})}{D_L(0)} \tag{2.4}$$

$$\left.\frac{\partial C}{\partial z}\right|_{z=L} = 0 \tag{2.5}$$

The axial diffusivity can be calculated from the following dimensionless correlation as shown in equation (2.6) [9].

$$D_L = \frac{D_m}{\varepsilon}(20 + 0.5ReSc) \tag{2.6}$$

Equation (2.4) includes the molecular diffusivity (D_m), the column void fraction (ε), the Reynolds number (Re), and the Schmidt number (Sc). The molecular diffusivity can be calculated using the Chapman-Enskog equation as shown in equation (2.7) [8].

$$D_m = \frac{0.001858T^{\frac{3}{2}}\left(\frac{1}{MW_a} + \frac{1}{MW_w}\right)^{\frac{1}{2}}}{P\sigma_{aw}\Omega_{aw}} \tag{2.7}$$

where MW_a and MW_w is the molecular weight of air and water respectively, T is the absolute temperature in Kelvin, P is the pressure in atmospheres, σ_{aw} is the collision diameter from the Lennard-Jones potential, and Ω_{aw} is the collision integral. The collision diameter and the collision integral can be calculated from the following set of equations [10].

$$\sigma_{aw} = \frac{1}{2}(\sigma_a + \sigma_w) \tag{2.8}$$

$$\Omega_{aw} = \frac{1.06036}{T^{*0.15610}} + \frac{0.19300}{e^{0.47635T^*}} + \frac{1.03587}{e^{1.52996T^*}} + 2\frac{1.76474}{e^{3.89411T^*}} \tag{2.9}$$

$$T^* = \frac{kT}{\varepsilon_{aw}} \tag{2.10}$$

$$\frac{\varepsilon_{aw}}{k} = \sqrt{\frac{\varepsilon_a}{k}\frac{\varepsilon_w}{k}} \tag{2.11}$$

where σ_a and σ_w are the Lennard-Jones potentials of air and water, respectively. The Lennard-Jones potential of air is 3.617 Å and that of water is 2.725 Å [10,11]. T^* is a parameter defined by equation (2.10), k is the Boltzman constant, and ε_{aw} is the Lennard-Jones force constant with constants for air 97 K and for water being 356 K [10,11]. The expression for axial diffusivity also includes 2 dimensionless numbers, the Reynolds number and the Schmidt number. The Reynolds number is a dimensionless number which describes the ratio of the inertial forces and the viscous forces as shown in equation (2.12). The Schmidt number relates the ratio of momentum diffusivity and mass diffusivity and is described by equation (2.13).

$$Re = \frac{\rho_f v d_p}{\mu_{MA}} \tag{2.12}$$

$$Sc = \frac{\mu_{MA}}{\rho_f D_m} \tag{2.13}$$

The Reynolds number is dependent on the density of moist air (ρ_f), the interstitial velocity (v) which is the superficial velocity divided by the void fraction, the particle diameter (d_p), and the viscosity of moist air (μ_{MA}). The Schmidt number is dependent on the molecular diffusivity as calculated in equation (2.7) as well as the gas viscosity and density. The density of moist air can be calculated following the CIPM-2007 method as shown below [12].

$$\rho_f = \frac{PMW_a}{ZRT}\left[1 - C\left(1 - \frac{MW_w}{MW_a}\right)\right] \tag{2.14}$$

$$C = \frac{p_{H2O}}{P} \tag{2.15}$$

$$p_{H2O} = CP \tag{2.16}$$

$$Z = 1 - \frac{P}{T}\cdot[a_0 + a_1 T + a_2 T^2 + (b_0 + b_1 T)C + (c_0 + c_1 T)C^2] + \frac{P^2}{T^2}\cdot(d + eC^2) \tag{2.17}$$

where P is the total pressure in Pascals, R is the ideal gas constant, C is the mole fraction of water in the air stream, Z is the compressibility factor, p_{H2O} is the partial pressure of water vapor, and

C is the concentration of water vapour in the air stream in mol water/ total moles. The compressibility factor can be calculated using equation (2.17) coupled with fitted parameters presented in *Table 2.1*.

Table 2.1: Fitting constants for compressibility factor calculation, fitted by Picard et al. (2008) through compressibility factor measurements at different temperatures and compositions of moist air [12]

Parameter	Value
a_0	1.58123 x10^{-6} K Pa^{-1}
a_1	-2.9331 x10^{-8} Pa^{-1}
a_2	1.1043 x10^{-10} K^{-1} Pa^{-1}
b_0	5.707x10^{-6} K Pa^{-1}
b_1	-2.051 x10^{-8} Pa^{-1}
c_0	1.9898 x10^{-4} K Pa^{-1}
c_1	-2.376 x10^{-6} Pa^{-1}
d	1.83 x10^{-11} K^2 Pa^{-2}
e	-0.765 x10^{-8} K^2 Pa^{-2}

The viscosity of moist air can be calculated by using a power law relationship coupled with fitting parameters knowing the fluid phase mole fraction (C_i) and temperature as shown below [13].

$$\mu_{MA} = \sum_{i=1}^{2} \frac{C_i \mu_i}{\sum_{j=1}^{2} C_j \varphi_{ij}} \tag{2.18}$$

$$\varphi_{ij} = \frac{1}{\sqrt{8}} \left(1 + \frac{MW_i}{MW_j}\right)^{-\frac{1}{2}} \left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{MW_j}{MW_i}\right)^{\frac{1}{4}}\right]^2 \tag{2.19}$$

$$\mu_i = \mu_{0,i} \left(\frac{T}{T_0}\right)^{n_{\mu,i}} \tag{2.20}$$

Table 2.2: Fitting Parameters for calculation of moist air viscosity, fitted by Kumar (2008) through viscosity measurements at different temperatures and concentrations [13]

	μ_0(Pa s)	$T_0(K)$	n_μ
μ_A	$1.716x10^{-5}$	273	0.666
μ_w	$1.12x10^{-5}$	350	1.15

The change in amount adsorbed with respect to time ($\frac{\partial \bar{q}}{\partial t}$) can be calculated using the linear driving force model of mass transfer which takes into account the overall mass transfer coefficient (k_m), the specific surface area of the adsorbent (a), and the driving force between the amount adsorbed at equilibrium (q^*) and the current amount adsorbed ($\bar{q}$) as shown in equation (2.21) [8].

$$\frac{\partial \bar{q}}{\partial t} = k_m a\,(q^* - \bar{q}) \tag{2.21}$$

$$\frac{1}{k_m} = \frac{m}{k_f} + \frac{1}{k_s} \tag{2.22}$$

The overall mass transfer coefficient can be calculated using equation (2.22). This equation considers the adsorption of water molecules onto the solid film (k_f) as well as the diffusion of water molecules from the film into the solid phase (k_s). The specific surface area (a) was estimated knowing the BET surface area results of a silica gel based adsorbent and the skeletal density of the simulated adsorbent (2668.6 kg/m^3) giving an estimated specific surface area of 1743 m^2/m^3. The equilibrium amount adsorbed was estimated using the Gugenheim Anderson De-Boer (GAB) isotherm model to model the water adsorption isotherm of the tested solid [14]. The GAB isotherm model along with its fitted parameters are presented in equations (2.23-2.26) and *Table 2.3*. The parameters presented in *Table 2.3* were all estimated using water vapour experimental adsorption isotherm data on a silica gel (SG) + calcium chloride ($CaCl_2$) solid

adsorbent excluding the parameter (M_{m0}). The parameter (M_{m0}) was instead fitted by gPROMS using experimental breakthrough curves. M_{m0} was instead fitted by gPROMS because the solid examined in this thesis was not the same as the solid used in the isotherm fitting. By changing only the M_{m0} parameter the shape of the fitted isotherm to silica gel is retained as well as its temperature dependence but the maximum adsorption capacity is modified accounting for the fact that a different adsorbent is used in this work. The parameters M_{m0} in equation (2.24), m in equation (2.22), and k_s in equation (2.22) were all estimated in gPROMS to provide the best fit with the experimental data. Values of the fitted parameters are presented in *Table 2.3*.

$$q^* = \frac{M_m B K\left(P_{H_2O}/P\right)}{\left[\left(1 - K\left(P_{H_2O}/P\right)\right)\right]\left[1 - K\left(P_{H_2O}/P\right) + BK\left(P_{H_2O}/P\right)\right]} \tag{2.23}$$

$$M_m = M_{m0}\, exp\left(\frac{q_m}{RT}\right) \tag{2.24}$$

$$B = B_0 \exp\left(\frac{\Delta H_B}{RT}\right) \tag{2.25}$$

$$K = K_0 \exp\left(\frac{\Delta H_k}{RT}\right) \tag{2.26}$$

Table 2.3: Fitted Isotherm Parameters by experimental moisture isotherm data and experimental breakthrough curves for the parameter M_{m0}

Parameter	Value
M_{m0}	12.025 mmol/g
q_m	2852.562 J/mol
B_0	0.227
ΔH_B	9294.969 J/mol
K_0	1.605
ΔH_k	-1537.831 J/mol

Table 2.4: Parameters fitted by gPROMS parameter estimation tool to experimental breakthrough curves

Parameter	Value
M_{m0}	12.025 mmol/g
m	331.302
k_s	3.139x10^{-6} m/s

The film mass transfer coefficient (k_f) was estimated using a dimensionless correlation for the Sherwood number as shown in equation (2.27) [8].

$$Sh = \frac{k_f * d_p}{D_m} = 2.0 + 1.1 Sc^{\frac{1}{3}} Re^{0.6} \quad (2.27)$$

2.4.2. Energy Balance

The energy balance was adapted from Cooney (1974) [7] and includes axial diffusion, accumulation in the fluid phase, convection term, accumulation in the solid phase, and heat generation due to adsorption and is shown in equation (2.28) [7,8].

$$-\frac{k_g}{\hat{c}_f\hat{\rho}_f\varepsilon}\frac{\partial^2 \mathrm{T}_f}{\partial z^2}+\frac{\partial T_f}{\partial t}+v\frac{\partial T_f}{\partial z}+\frac{\hat{\rho}_s\hat{c}_s}{\tilde{\rho}_f\tilde{c}_f}\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{\partial T_s}{\partial t}+\frac{\Delta\widetilde{H}\hat{\rho}_s}{\tilde{\rho}_f\tilde{c}_f}\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{\partial \bar{q}}{\partial t}=0 \tag{2.28}$$

The parameters in the equation include the thermal conductivity of the fluid (k_g), the heat capacity of the fluid ($\tilde{c}_f$), the interstitial velocity (v), the solid density ($\hat{\rho}_s$), the fluid density ($\tilde{\rho}_f$), the fluid heat capacity (c_f), the void fraction (ε), and the heat of adsorption ($\Delta\widetilde{H}$). The boundary conditions used for the energy balance were derived from Danckwerts boundary conditions and are presented below along with the initial conditions [15,8].

$$T_f(z)\big|_{t=0}=0 \tag{2.29}$$

$$T_s(z)|_{t=0}=0 \tag{2.30}$$

$$\left.\frac{\partial T_f}{\partial z}\right|_{z=0}=\frac{\left(\frac{\rho_f\hat{c}_f v}{\varepsilon}\right)\left(T_f(0)-T_{inlet}\right)}{k_g(0)} \tag{2.31}$$

$$\left.\frac{\partial T_f}{\partial z}\right|_{z=L}=0 \tag{2.32}$$

The fluid thermal conductivity can be calculated by the following a power law-based relationship knowing the fluid phase mole fraction (C_i) and temperature as shown below [13].

$$k_g=\sum_{i=1}^{2}\frac{C_i k_i}{\sum_{j=1}^{2}C_j\varphi_{ij}} \tag{2.33}$$

$$\varphi_{ij}=\frac{1}{\sqrt{8}}\left(1+\frac{MW_i}{MW_j}\right)^{-\frac{1}{2}}\left[1+\left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}}\left(\frac{MW_j}{MW_i}\right)^{\frac{1}{4}}\right]^2 \tag{2.34}$$

$$k_i=k_{0,i}\left(\frac{T}{T_0}\right)^{n_{k,i}} \tag{2.35}$$

Table 2.5: Fitting parameters for calculation of fluid thermal conductivity fitted by Kumar (2008) through thermal conductivity measurements at different temperatures and concentrations [13]

	$\boldsymbol{k_0}$**(W/m/K)**	$\boldsymbol{T_0(K)}$	$\boldsymbol{n_k}$
k_A	0.0241	273	0.81
k_w	0.0181	300	1.35

The heat capacity of moist air can be calculated through a weighted average of the heat capacity of water vapor and air by the water vapor concentration as shown below [16].

$$\tilde{c}_f = (1 - C)c_{DA} + Cc_w \tag{2.36}$$

$$c_i = R(A' + B'T + C'T^2 + D'T^{-2}) \tag{2.37}$$

where C is the mole fraction of water vapor in the air stream and the heat capacity of air and water vapor is calculated using equation (2.37) and their respective fitting parameters shown in *Table 2.6*. The fitting parameters are obtained by Smith et al. (2005) by curve fitting the experimental heat capacity data with equation (2.37) at many different temperatures [16].

The solid skeletal density ($\hat{\rho}_s$) of the tested sample was obtained from a helium pycnometer test performed at National Resources Canada (NRCan) and was evaluated to be 2668.6 kg/m^3. The solid heat capacity was estimated to be 1200 J/kg/K from data obtained by Scuiller et al. 2022 for dry and hydrated zeolite 13x [17].

Table 2.6:Fitting parameters given by Smith et al. (2005) for dry air and water vapor heat capacity fitted to temperature [16]

Dry Air	
A'	3.355
B'	0.575 x10^{-3}
C'	…
D'	-0.016 x10^{5}
Water Vapour	
A'	3.470
B'	1.450 x10^{-3}
C'	…
D'	0.121 x10^{5}

The solid accumulation term ($\frac{\partial T_s}{\partial t}$) can be calculated from equation (2.38) shown below. This equation is dependent on the solid fluid heat transfer coefficient which can be calculated using dimensionless correlations of the Nusselt number as shown in equations (2.39-2.42) [18].

$$\frac{\partial T_s}{\partial t} = \frac{ha}{\hat{\rho}_s \hat{c}_s}\left(T_f - T_s\right) + \frac{\left(-\Delta\tilde{H}\right)}{\hat{c}_s}\frac{\partial \bar{q}}{\partial t} \tag{2.38}$$

$$Nu_p = \frac{hd_p}{k_g} = Nu_0 (RePr)^{1-\varepsilon}\varepsilon^{-\beta} \tag{2.39}$$

$$Nu_0 = 2 + 0.478Re^{0.48}Pr^{0.45} + 0.012Re^{1.1}Pr^{0.78} \tag{2.40}$$

$$\beta = (3.92\varepsilon^2 + 0.34\varepsilon + 10.3) - 12\exp[0.38\varepsilon^2 \log Pr] + (3.13\varepsilon^3 - 5.43\varepsilon^2)\log Re \tag{2.41}$$

$$Pr = \frac{\mu_{MA}\hat{c}_f}{k_g} \tag{2.42}$$

2.5. Experimental Results

Experiments were conducted using a flowrate of 24SLPM and an inlet relative humidity of 80% based on previously conducted optimization studies conducted by Strong, et al. (2022) [19]. These studies show that using higher inlet relative humidities and higher flowrates of 24 SLPM yielded more heat released [19]. To study the effect of length to diameter (L/D) ratio and column volume on reported energy storage density (ESD), 9 different adsorption columns were designed and constructed to test 3 different column volumes at 3 different L/D ratios. Presented in Table 2.7 are the dimensions of each column tested.

Table 2.7: Dimensions of tested adsorption columns. SA: Surface Area, OD: Outer Diameter, ID: Internal Diameter

Measured	A	B	C	D	E	F	G	S	M
OD (cm)	4.96	5.06	4.82	5.06	6.01	4.84	3.33	1.16	3.78
ID (cm)	2.03	1.54	4.11	2.33	5.27	4.06	2.69	1.09	3.33
Thickness (cm)	1.47	1.76	0.36	1.37	0.37	0.39	0.32	0.03	0.23
Length (cm)	2.20	3.63	4.85	15.30	5.60	8.00	20.58	7.62	7.43
Volume (cm^3)	7.12	6.76	64.35	65.24	122.15	103.57	116.96	7.08	64.61
Length/Diameter	1.08	2.36	1.18	6.57	1.06	1.97	7.65	7.00	2.23
SA/Volume	6.56	8.82	1.55	3.10	1.22	1.38	1.79	4.07	1.59

As Shown in Table 2.7, the tested L/D ratios are 1, 2.2, and 7 while the tested column volumes are 7, 64, and 114 cm^3. These dimensions were selected knowing only a limited supply of adsorbent was on hand. It is important to highlight the variation in column thicknesses and in values like L/D ratios and volume. These variations were due to column machining and part purchasing limitations giving rise to differences between L/D ratios, volume, and thicknesses. However, the resulting trends from the collected experimental results can give estimates into the effects that L/D ratio, and column volume has on reported ESD. Included in *Table 2.7* is the surface area to volume ratio (SA/Volume) which is considered to evaluate which column walls can better participate in heat transfer to the surroundings and have more thermal mass which can affect the results. These effects are discussed in section 2.7. The column diameter to pellet diameter ranges from a minimum of 5 to a maximum of 25 with an average of 14 for all the experiments. If the ratio of column diameter to pellet diameter is less than the recommended 10 in some instances this can introduce inefficient packing and differences for the experiments.

However, due to the almost perfectly spherical shape of the pellets and the tight packing, the packing density remained constant for all the experiments at a packing density of 860 kg/m^3, which showed that these experiments are comparable.

Results show a decreasing trend with ESD as L/D ratio increases and an increasing trend of ESD with increase in column volume as shown in Table 2.8, Figure 2.2, and Figure 2.3. The average ESD was found to be 123 kWh/m^3. The trends observed in the ESD with respect to column volume is corroborated by Banaei et al 2021 [20].

Table 2.8: Experimental ESD vs volume & L/D ratio values

		Volume (cm3)		
	ESD (kWh/m3)	**7**	**64**	**114**
L/D	**1**	124	125	129
	2.2	120	123	128
	7	117	118	122

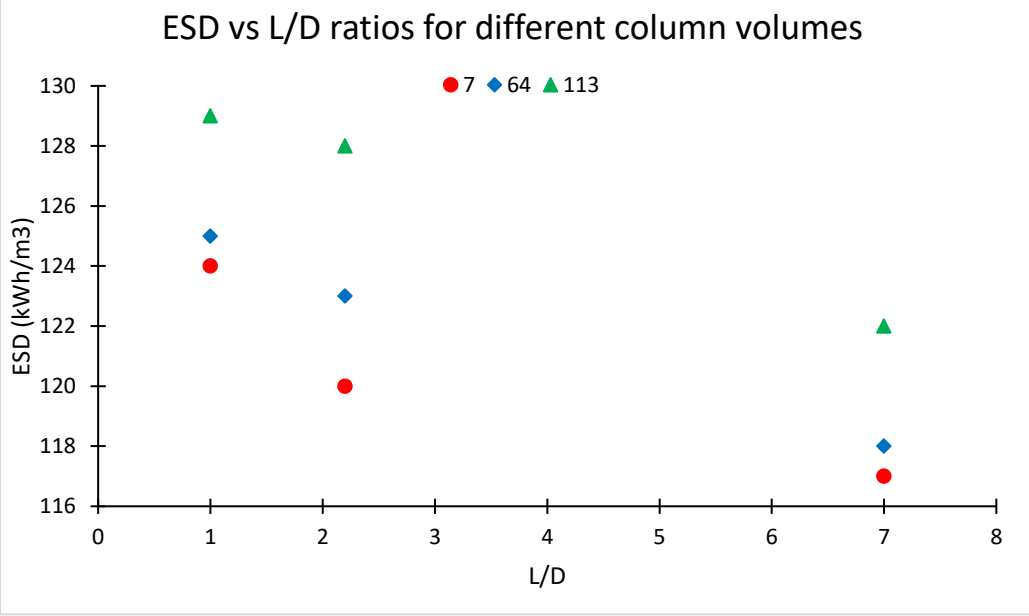


Figure 2.2: Experimental ESD vs L/D at column volumes of 7, 64, 113 cm^3

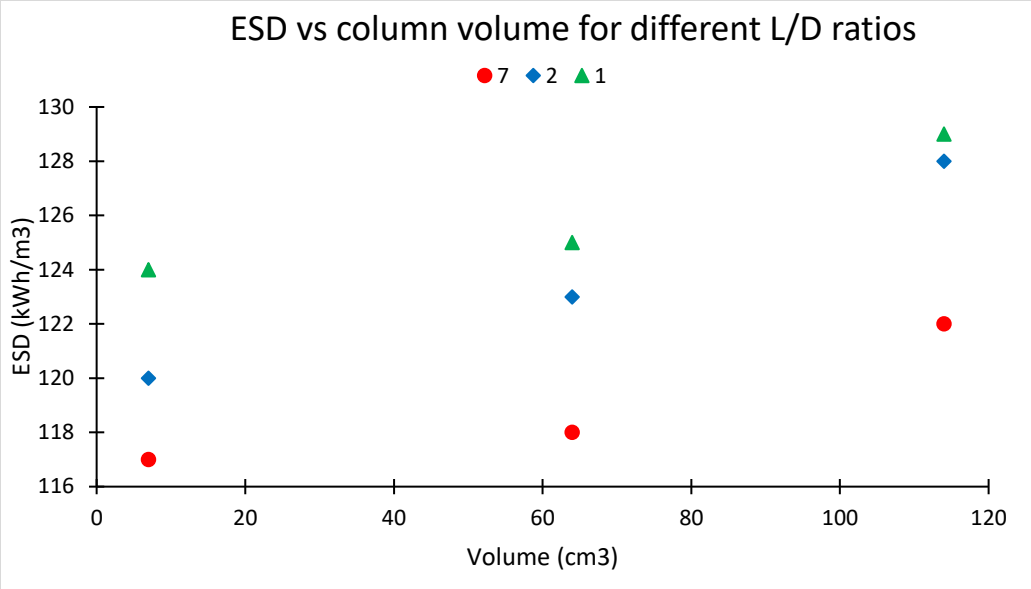


Figure 2.3: Experimental ESD vs column volume for L/D ratios of 1, 2.2, 7

2.6. Model Validity

The model was scripted in gPROMS model builder and simulations were conducted using inlet conditions from the experimental data set such as variable inlet concentration and temperature as well as a fixed inlet pressure and flowrate. The model has good agreement with the experimental data estimating the ESD within 85-92% of the ESD evaluated with experimental results with an average of 88%. Shown below in *Table 2.9* are the ESD values predicted by gPROMS for different column volumes and L/D ratios.

Table 2.9: ESD values predicted by gPROMS for different volumes and L/D ratios

		Volume (cm3)		
	ESD (kWh/m3)	**7**	**64**	**114**
L/D	**1**	114	108	110
	2.2	105	108	109
	7	102	104	110

As can be seen from the above table the general trends of increasing ESD with increasing volume and decreasing ESD with increasing L/D ratio are also observed. While it can be seen from the above table that there are some extraneous data points that do not follow this trend such as the column with a volume of 7 cm^3 and L/D ratio of 1, the column with a volume of 114 cm^3 and L/D ratio of 7, and the column with a volume of 64 cm^3 and L/D ratio of 2.2. These extraneous data points can be attributed to a couple of factors outside the experimental control. For example, the column with volume of 7 cm^3 and L/D ratio of 1 had the highest inlet concentration of any of the conducted experiments of 0.019 mole water/mole total. This higher inlet concentration allows for more water to be adsorbed in the simulation allowing for more heat to be released resulting in a higher reported ESD which does not fit the expected trends. When examining the effects of L/D ratio at a column volume of 114 cm^3 it can be seen that the ESD does not change between L/D of 1 and 7. This can also be attributed to the fact that the inlet concentration was higher in the column with L/D of 7 also giving rise to a higher ESD than expected. This same effect was also observed between L/D ratios of 1 and 2.2 with a column volume of 64 cm^3 where the inlet concentration was higher at L/D ratio of 2.2 which skews the results showing the effect of L/D and column volume. While select experiments had different inlet conditions which effected the reported ESD, the gPROMS model was still able to capture trends seen with experimental results. In addition to the trends captured by gPROMS the model was also able to accurately capture

breakthrough curves and outlet temperature profiles. Outlined below are the model predicted concentration breakthrough curves and temperature profiles vs their experimental counter parts. It is important to note that because of inaccuracies in relative humidity sensors some of the experimental concentration breakthrough data does not perfectly overlap the inlet concentration data while the gPROMS data does because it was fed the inlet concentration data.

As can be seen from the experimental results, a higher temperature lift is observed in larger column volumes compared to smaller column volumes. This can be attributed to the fact that larger column volumes have more adsorbent to participate in heat and mass transfer and increase the temperature of the moist air stream. Comparing the L/D ratios for the largest column volume shows the highest temperature lift observed at a L/D ratio of 7 with a lift of 28.4°C as shown in *Figure 2.12*. For L/D of 2.2 and 1, the temperature lifts are 22°C and 19°C, respectively, for the largest column volume of 114 cm^3 as shown in *Figure 2.10* and *Figure 2.11* respectively. While the higher temperature lift is observed for the L/D ratio of 7, the peak was narrower compared to lower L/D ratios at the same volume as shown in *Figure 2.11* and *Figure 2.10*. This can be attributed to the higher surface area (for column G) participating in heat transfer outside of the column compared to lower L/D ratios of the same volume (column E and F) as shown in *Table 2.7*. This reduces the outlet temperature quicker compared to smaller L/D ratios decreasing the reported ESD.

The same effect of increasing temperature lifts with increasing L/D ratio was also observed in the smallest column volume of 7cm^3 as shown in *Figure 2.4-2.6*.

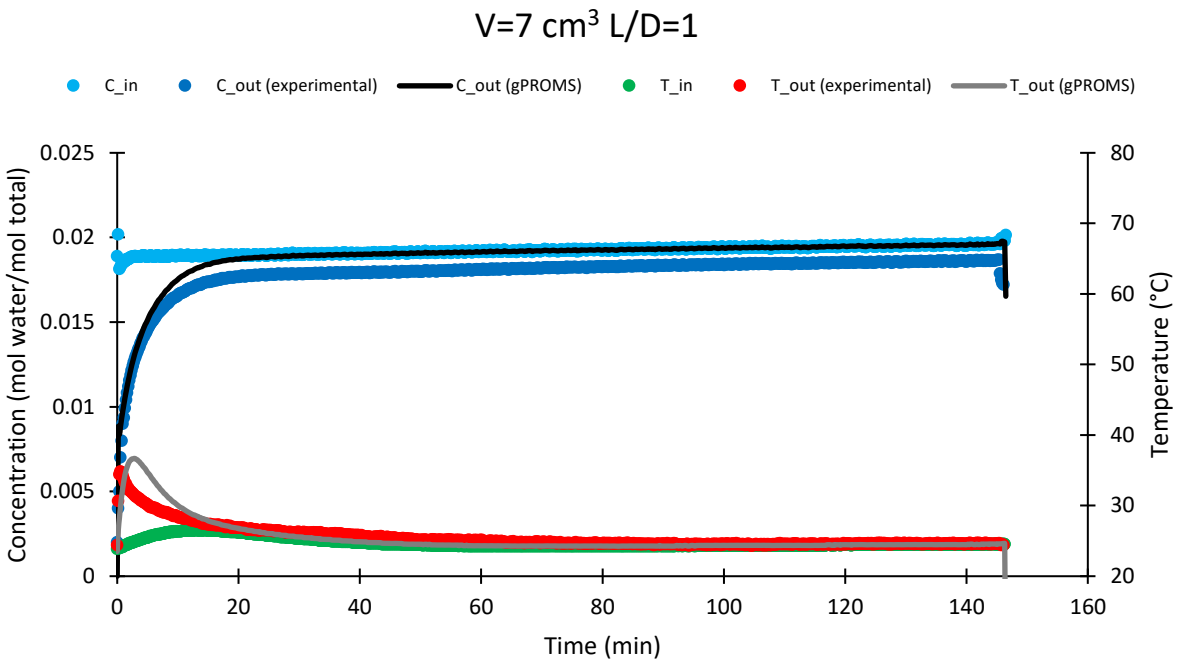


Figure 2.4: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=7cm^3 and L/D=1 for column A

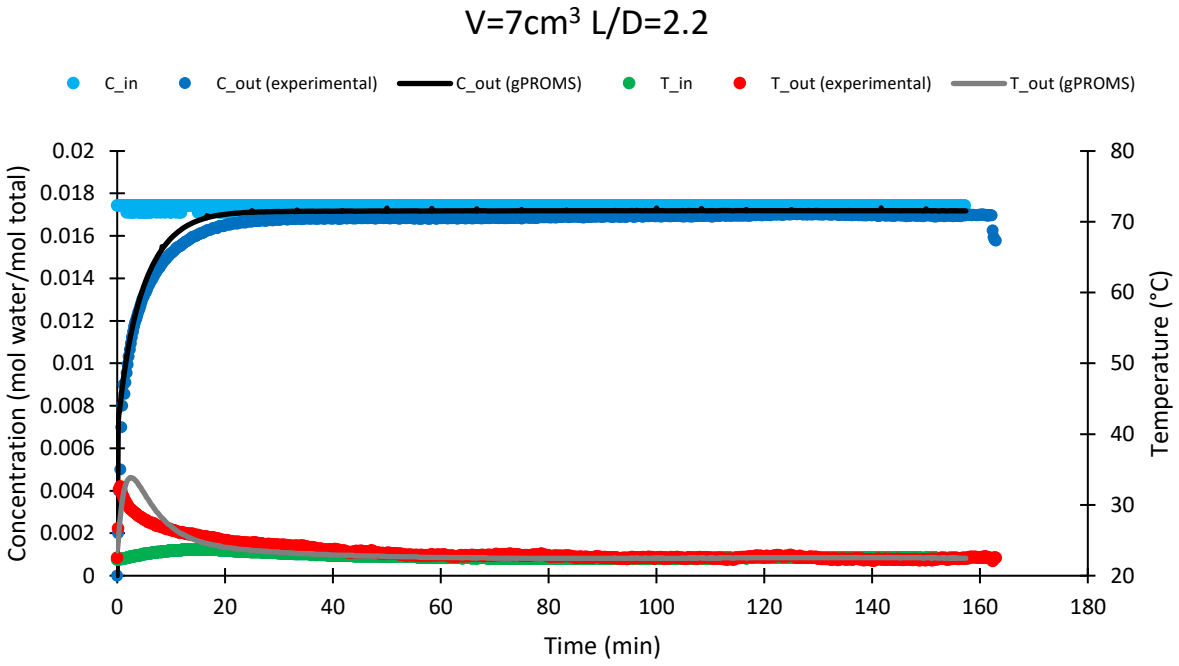


Figure 2.5: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=7cm^3 and L/D=2.2 for column B

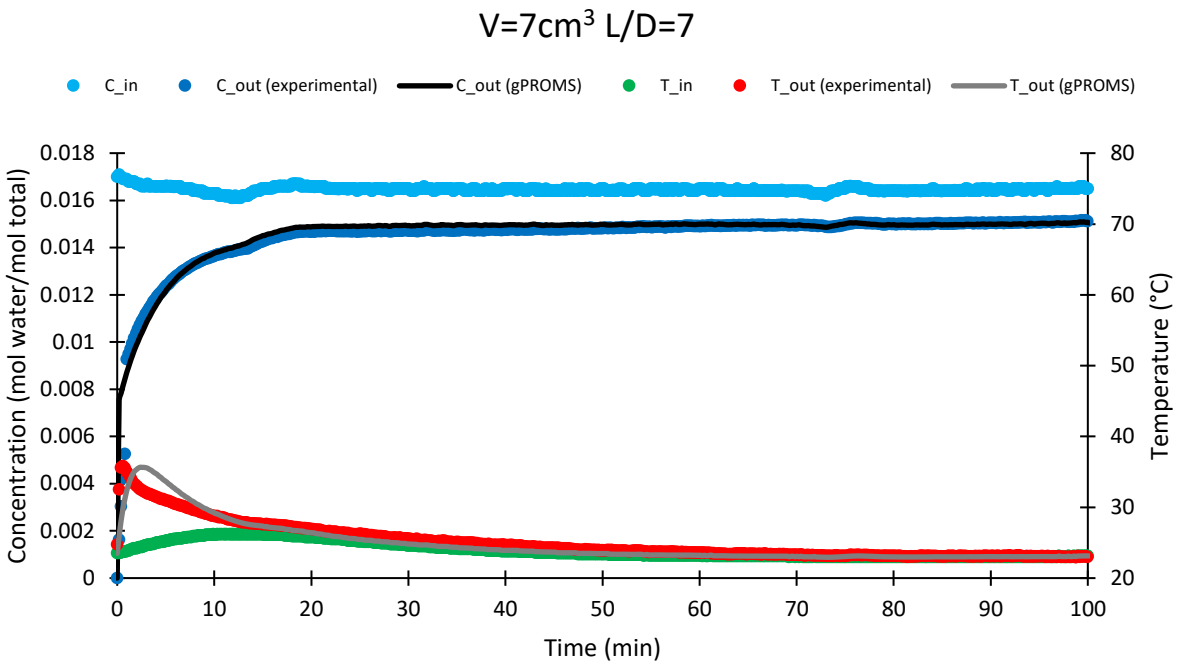


Figure 2.6: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=7cm^3 and L/D=7 for column S

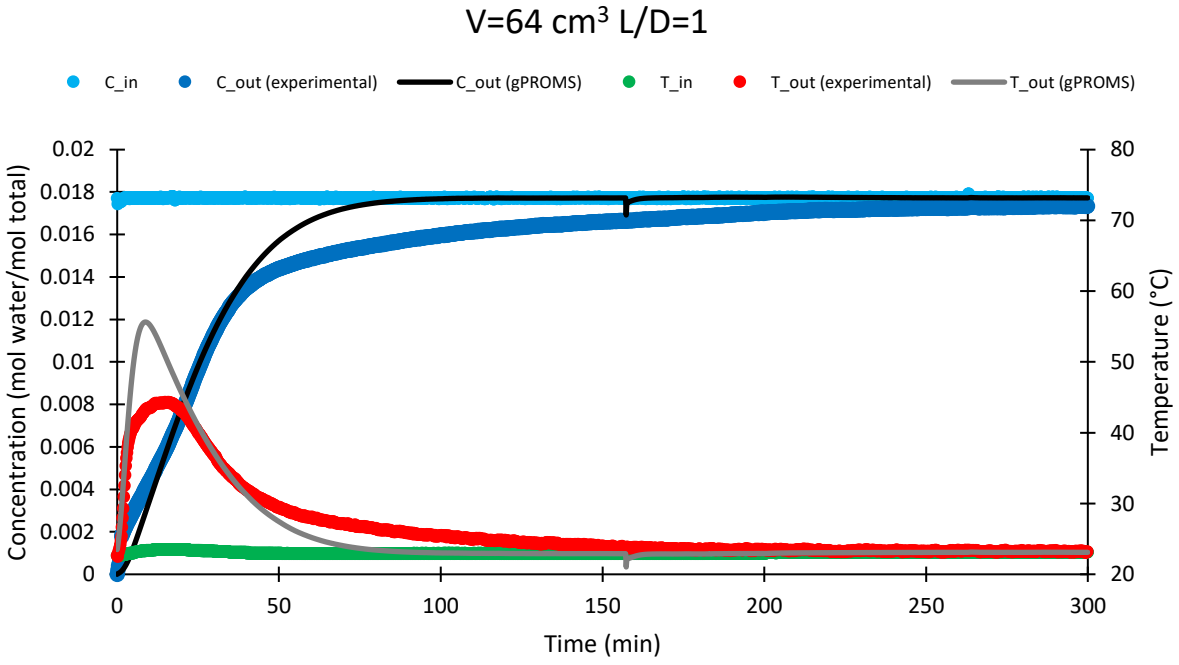


Figure 2.7: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=64 cm^3 and L/D=1 for column C

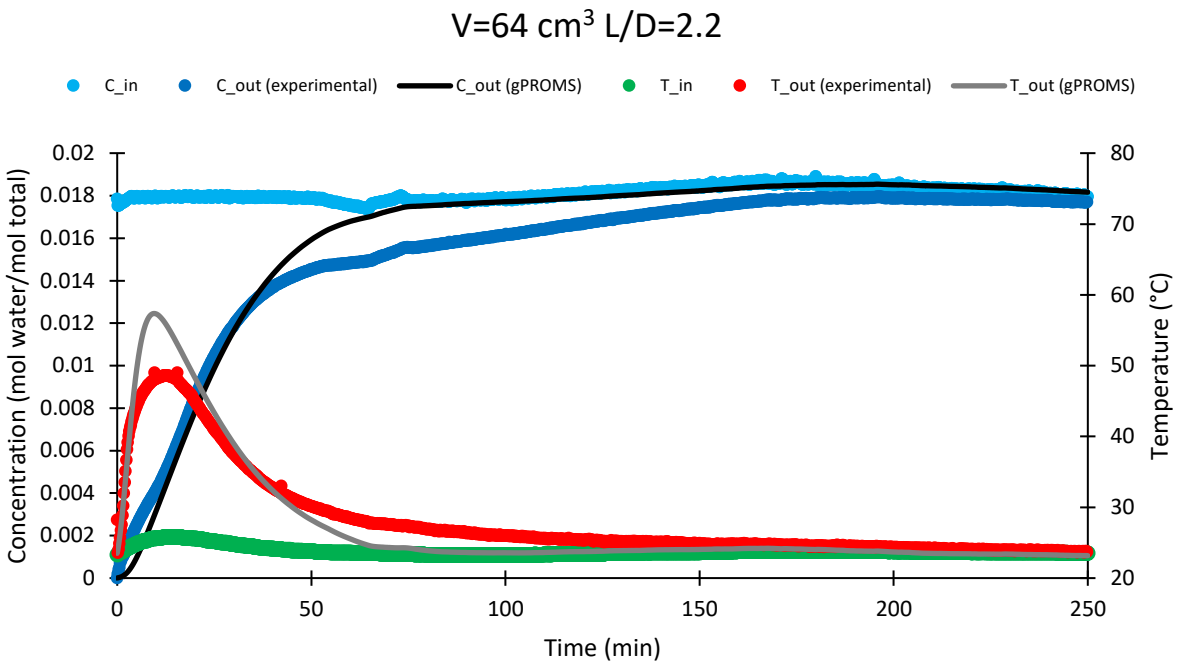


Figure 2.8: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=64 cm^3 and L/D=2.2 for column M

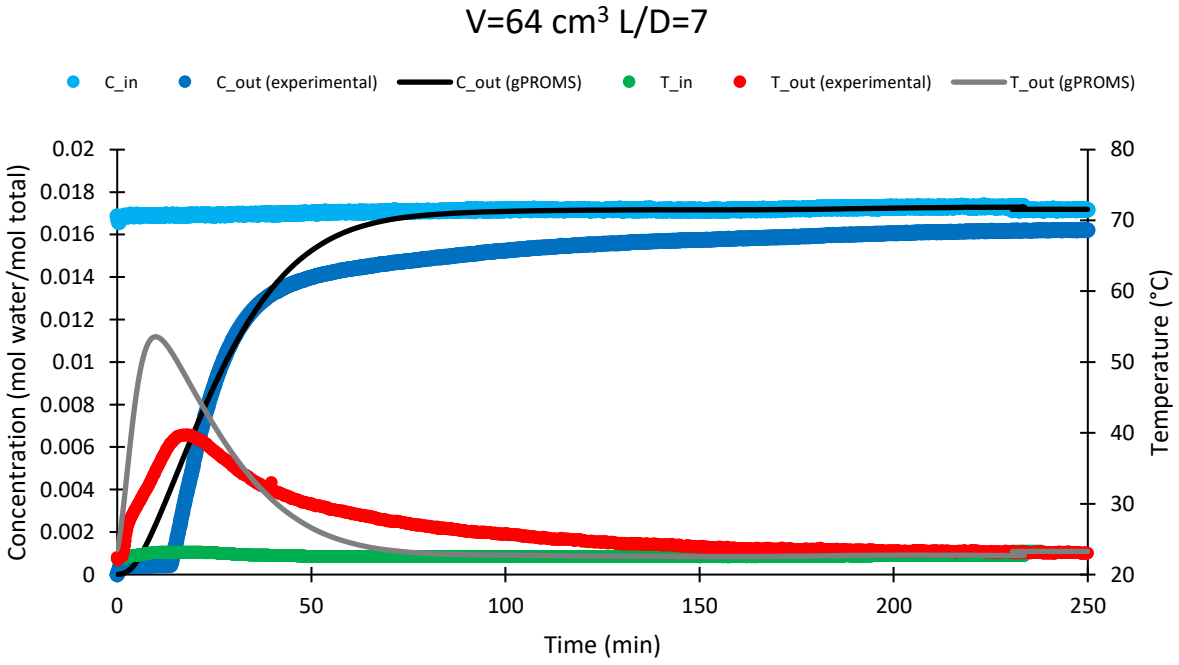


Figure 2.9: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=64 cm^3 and L/D=7 for column D

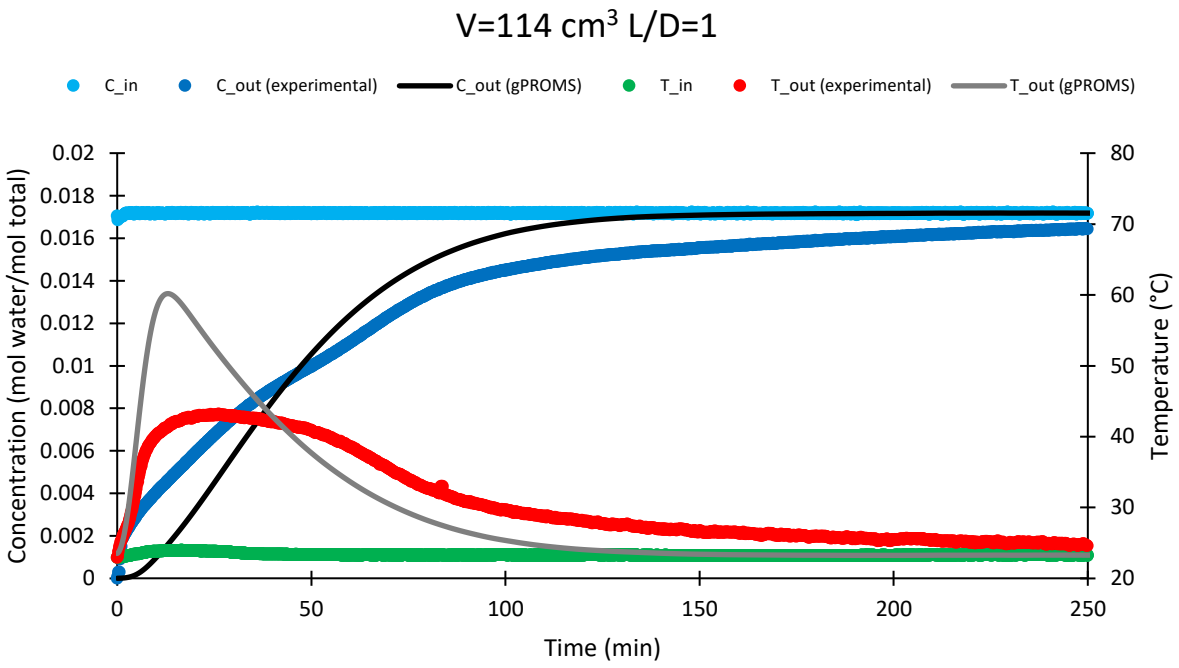


Figure 2.10: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=114cm^3 and L/D=1 for column E

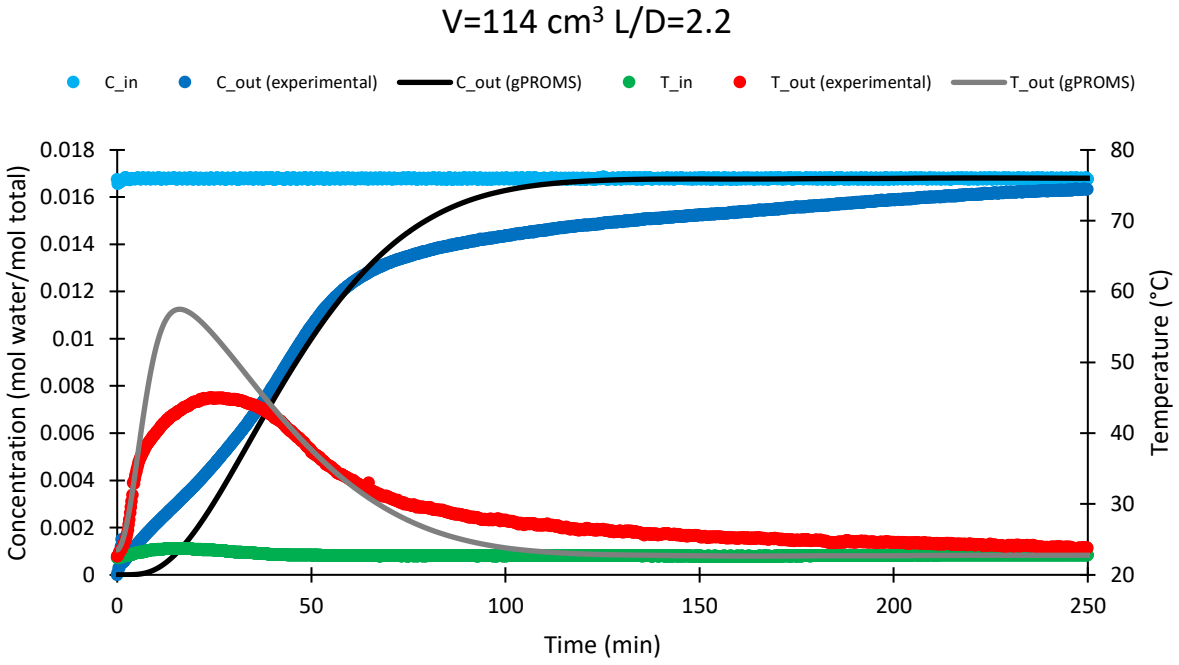


Figure 2.11: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=114 cm^3 and L/D=2.2 for column F

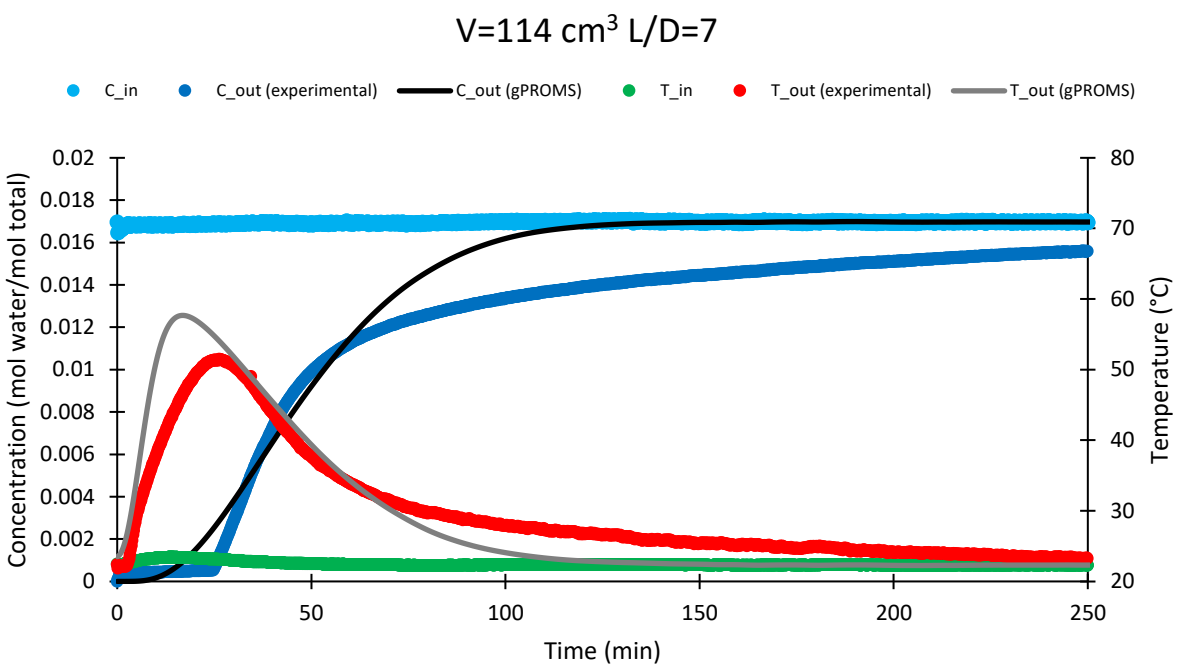


Figure 2.12: Comparison of experimental results with g-PROMS model results at inlet flow rate of 24 SLPM at 80 % RH, for V=114 cm^3 and L/D=7 for column G

2.7. Discussion

As previously discussed, the average energy storage density was experimentally evaluated to be 123 kWh/m^3 with a decreasing trend in energy storage density with increasing L/D ratios and an increasing trend in ESD as column volume increases. The decreasing trend of ESD with increasing L/D ratios can be attributed to the fact that longer column walls act as a better heat sink because of the higher surface area the fluid encounters allowing for more heat to be transferred to the column walls. This is corroborated by the surface area to volume ratios presented in *Table 2.7*, showing higher surface area to volume ratios for higher L/D ratios for the same total volume of columns, except for the smallest volume columns (7 cm^3) where column S with the thinnest column wall (with a thickness of 0.03 cm) had a smaller SA/Volume ratio than column B. Column S did not match this trend where the smaller L/D ratios of the 7 cm^3 columns had larger surface area to volume ratios than the largest L/D ratio. This is because they (columns A and B) were constructed with thicker walls due to part purchasing and machining capabilities, which increased their SA/Volume ratio. Despite this deviation, the ESD still followed the expected trend where the lowest ESD was observed at L/D of 7 and the highest ESD was observed at L/D of 1 for the smallest volume columns A, B, and S.

The fact that the ESD still follows decreasing trend with increasing L/D ratio despite the difference in SA/Volume ratio can be explained by the difference in inlet concentrations where the lowest inlet concentration was observed at L/D ratio of 7 and highest at L/D ratio of 1. This difference in concentrations allowed more moisture to be adsorbed in the smaller L/D ratios (columns A and B) compared to the largest (column S) giving rise to a higher ESD in lower L/D ratios although they had a higher SA/Volume ratio. Column A with volume 7 cm^3 and L/D 1 had an inlet concentration of 0.019 mol H_2O / total mol, column B with volume 7 cm^3 and L/D ratio 2.2 had an inlet concentration of 0.017 mol H_2O / total mol, and column S with volume 7 cm^3 and L/D 7 had an inlet concentration of 0.016 mol H_2O / total mol.

The assumed reason that the increasing trend of ESD with increasing column volume is observed is due to the fraction of heat transferred out of the column being less than that of the smaller column. Because less total heat is released in smaller columns any heat losses from the column wall will have a bigger impact on reported ESD's than larger columns with much more total heat released.

As discussed in Section 6 the model shows good agreement with experimental data while the ESD estimates are within 85-92% of the experimentally determined ESD. As can be seen with the larger column volumes in Figure 2.7-Figure 2.12, the concentration breakthrough and outlet temperature profiles deviate more from the model than for the smaller column volumes. This can be attributed to the fact that the adsorption model does not consider mass transfer inside the pellet. This consideration would slow down the breakthrough time and time to reach thermal equilibrium as moisture diffuses from the film to the adsorbent pellet center slowly. This process allows for a slower saturation of the adsorbent and broader concentration breakthrough curves and outlet temperature profiles providing a better fit.

While it is desirable to have the most ideal experimental conditions such as constant inlet concentration, temperature, and pressure between all experimental trials, it was impossible in the experimental setup. Because building supplied air was used as a carrier gas it is subject to temperature fluctuations outside of experimental control resulting in slightly different inlet temperatures between experiments. Additionally, because the experimental setup controls the inlet relative humidity (which is dependent on temperature and pressure), constant inlet concentration was not observed between experiments due to different pressure drops of different columns and different inlet temperatures between experiments. That being said, good estimates on the behaviour of column geometry can be extrapolated from the data. It can be seen in the adsorption analysis graphs in Section 2.6 that the temperature lifts are generally higher in gPROMS than experimental results. This can be explained by the model being treated as completely adiabatic (that is no heat is transferred from the fluid to the column wall or surroundings) as well as mass transfer in the pellet not being accounted for.

Exclusion of adiabatic operation from the model would reduce the temperature lift of the model and broaden out the temperature profile. Additionally, inclusion of mass transfer inside the pellet would allow for more heat to be released and further broaden the temperature profile (as well as concentration breakthrough profile). However, because of the complexity of estimating heat transfer coefficients between the fluid, column wall, and surroundings it was omitted from the model. Additionally, due to the complexity of examining mass transfer in the pellet it was also omitted from the model. Omission of these factors also aided in reducing the simulation time which was also an important factor in simulating these experiments, while the model still provided good insights and estimates into the behaviour of this adsorption system.

2.8. Conclusions

In conclusion, the effects of column geometry were studies by varying 3 different lengths to diameter ratios at 3 different column volumes. Geometries studied were L/D ratios of 1, 2.2, and 7 at column volumes of 7, 64, and 114 cm^3. Results observed from studying the effect of column geometry on reported energy storage density showed a decreasing trend of energy storage densities with increasing length to diameter ratios and increasing trend of energy storage density with increasing column volume. The average evaluated energy storage density was found to be 123 kWh/m^3 across all 9 experiments. The decreasing trend of ESD with increasing L/D ratios can be attributed to the fact that longer columns' walls act as better heat sinks because of the higher surface area the fluid encounters allowing for more heat to be transferred to the column walls. This effect can be seen when examining the surface area to volume ratios of the columns in *Table 2.7*. In this table, the highest surface area to volume ratios are observed with the columns with the highest L/D ratios allowing more heat to be lost per unit volume. However, column S does not follow these same trends having a smaller surface area to volume ratio than columns A and B. This is attributed to the fact the column walls are much thicker in columns A and B then S due to part purchasing and machining capabilities, increasing the surface area to volume ratios for these columns. The reason that the increasing trend of ESD with increasing column volume is observed is the fact that the fraction of heat lost out of the column being less in larger columns compared to that of the smaller column. Because less total heat is released in smaller columns, any heat losses from the column wall will have a bigger impact on reported ESD's than larger columns with much more total heat released.

The fitted model showed good agreement with experimental data estimating the energy storage density with 85-92% accuracy. If the mass transfer inside the pellets were considered it is expected that the model would show better agreement with experimental data set, however, due to the complexity of the simulation and not having access to the kinetic parameters within the column, it was omitted to help with the speed of the simulation which is an important parameter for this use case.

2.9. Acknowledgement

The work of this thesis was done in collaboration with a couple different people. Leonardo Rubalcava and Jeremy Simanjuntak assisted me with performing and analysing experimental results while studying the effects of column geometry. Dr. Cynthia Cruickshank from the Department of Mechanical Engineering at Carleton University provided support and insights into modelling this technology in a TRNSYS simulation, as well as partial funding of this work through NSERC (Natural Sciences and Engineering Research Council Council) of Canada. Dr. F. Handan Tezel supervised and supported me in performing computational modelling as well as analysis of raw data collected in her lab at the University of Ottawa. Most of the financial support for this work was also provided by her through funding received from NSERC. Dr. Alexandra Mallett from the School of Public Policy and Administration at Carleton University provided different outlooks on the project from a policy perspective to aid in conclusions from different points of view.

2.10. References

[1] Natural Resources Canada, "Natural Resources Canada," Government of Canada, 2020. [Online]. Available: https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/showTable.cfm?type=CP&sector=res&juris=ca&rn=7&year=2020&page=3. [Accessed 02 04 2024].

[2] NASA, "Carbon Dioxide," NASA, February 2024. [Online]. Available: https://climate.nasa.gov/vital-signs/carbon-dioxide/?intent=121. [Accessed 04 04 2024].

[3] Core Writing Team, H. Lee and J. Romero, "Climate Change 2023: Synthesis Report," IPCC, Geneva, Switzerland, 2023.

[4] V. Masson-Delmotte, P. Zhai, H. Portner, D. Roberts, J. Skea, P. Shukla, A. Pirani, W. Moufouma-Okia, C. Peann, R. Pidcock, S. Connors, J. Matthews, Y. Chen, X. Zhou, M. Gomis, E. Lonnoy, T. Maycock, M. Tignor and T. Waterfields, "IPCC, 2018: Summary for Policymakers," Cambridge University Press, New York, 2018.

[5] M. Green, "Community power," Nature Energy, no. 1, 2016.

[6] D. Lefebvre, P. Amyot, B. Ugur and F. H. Tezel, "Adsorption Prediction and Modeling of Thermal Energy Storage Systems: A Parametric Study," Industrial & Engineering Chemistry Research, vol. 55, no. 16, pp. 4760-4772, 2016.

[7] D. O. Cooney, "Numerical Investigation of Adiabatic Fixed-Bed Adsorption," Industrial & Engineering Chemistry Process Design and Development, vol. 13, no. 4, pp. 368-373, 1974.

[8] D. M. Ruthven, Principles of Adsorption and Adsorption Processes, John Wiley & Sons, Inc, 1984.

[9] N. Wakao and T. Funazkri, "Effect of fluid dispersion coefficients on particle-to-fluid mass transfer coefficients in packed beds," Chemical Engineering Science, vol. 33, pp. 1375-1384, 1978.

[10] J. O. Hirschfelder, C. F. Curtiss and R. B. Bird, Molecular Theory of Gases and Liquids, John Wiley & Sons, Inc., 1964.

[11] O. Singh and A. W. Joshi, "Effective potential for water vapour," Pramana, vol. 15, no. 5, pp. 407-412, 1980.

[12] A. Picard, R. S. Davis, M. Glaser and K. Fujii, "Revised formula for the density of moist air (CIPM-2007)," Metrologia, vol. 45, pp. 149-155, 2008.

[13] S. A. Kumar, "Estimating the Effect of Moist Air on Natural Convection Heat Transfer in Electronics Cooling," Qpedia, Bangalore, India, 2008.

[14] M. Sultan, I. I. El-Sharkawy, T. Miyazaki, B. B. Saha, S. Koyama, T. Maruyama, S. Maeda and T. Nakamura, "Insights of water vapor sorption onto polymer based sorbents," Adsorption, vol. 21, pp. 205-215, 2015.

[15] P. V. Danckwerts, "Continuous Flow Systems. Distribution of Residence Times," Chemical Engineering Science, vol. 2, p. 3857, 1953.

[16] J. M. Smith, H. C. Van Ness and M. M. Abbott, Introduction to Chemical Engineering Thermodynamics seventh edition, New York: McGraw Hill, 2005.

[17] E. Scuiller, P. Dutournie, M. Zbair and S. Bennici, "Thermo-physical properties measurements of hygroscopic and reactive material (zeolite 13X) for open adsorptive heat storage operation," Journal of Thermal Analysis and Calorimetry, vol. 147, pp. 12409-12416, 2022.

[18] Z. Qi and A. B. Yu, "A new correlation for heat transfer in particle-fluid beds," International Journal of Heat and Mass Transfer, vol. 181, p. 121844, 2021.

[19] C. Strong, Y. Carrier and H. F. Tezel, "Experimental optimization of operating conditions for an open bulk-scale silica gel/water vapour adsorption energy storage system," Applied Energy, vol. 312, p. 118533, 2022.

[20] A. Banaei and A. Zanj, "A Review on the Challenges of Using Zeoloite 13X as Heat Storage Systems for the Residential Sector," energies, vol. 14, no. 23, p. 8062, 2021.

3. Modeling and Simulation of Adsorptive Thermal Energy Storage and its Effect on Home Heating

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2024/09/25

Abstract

In 2020, over half (53.5%) of the energy consumed by the space heating sector in Canada was supplied by fossil fuels [1]. To help reduce fossil fuel emissions and reach 2050 net zero goals a reduction in greenhouse gas emissions is required. To help reduce emissions in the residential space heating sector adsorptive thermochemical energy storage is explored as a potential technology. In this paper, the integration of an adsorptive thermochemical energy storage device is explored in real home heating scenarios. Using Fortran, a fitted model from gPROMS simulation software was coded and integrated in TRNSYS software. A building model was used to simulate the thermal battery in real home heating scenarios. The building model in TRNSYS was designed as a 1500 ft^2 building in Ottawa, Ontario, Canada using the Carleton University CHEeR house as a basis of material of construction which takes into account heat gains from people, lighting, appliances, and electronics. Heat losses were also considered in the form of infiltration and heat transfer through the walls of the building. Simulations were conducted over 1 week where the 22m^3 thermal battery was operational for 25 hours of that time using a flowrate of 1500 kg/hr moist air at 90% relative humidity. In this time the thermal battery was depleted. The main cost for this adsorption column is the adsorbent itself costing approximately $31,000-$42,000 USD. Because of the high capital cost of the system and amount of time the battery is able to supply reliable heat this technology is not currently ready for widespread adoption and more development in materials and adsorption column designs are required.

Nomenclature

Symbol	Name	Units
a	Surface area to volume ratio	m^{-1}
C	Gas phase concentration	*mol water/total mol*
$\hat{c}_f$	Fluid heat capacity	*J/kg moist air/K*
$\tilde{c}_f$	Fluid heat capacity	*J/mol moist air/K*
$\hat{c}_s$	Solid heat capacity	*J/kg solid/K*
D_L	Axial diffusivity	m^2/s
h	Solid fluid heat transfer coefficient	$W/m^2/K$
k_g	Fluid thermal conductivity	*W/m/K*
k_m	Overall mass transfer coefficient	*m/s*
$\bar{q}$	Amount adsorbed	*mol /kg*
q^*	Equilibrium amount adsorbed	*mol/kg*
t	Time	*s*
T_f	Fluid temperature	*K*
T_s	Solid temperature	*K*
v	Column interstitial velocity	*m/s*
z	Position in column	*m*
$\Delta\tilde{H}$	Heat of adsorption	*J/mol*
ε	Void fraction	-
$\tilde{\rho}_f$	Fluid density	*mol moist air/m^3*
$\hat{\rho}_f$	Fluid density	*kg moist air/m^3*
$\hat{\rho}_s$	Solid density	*Kg solid/m^3*

Abbreviations

Abbreviation	Meaning
CHEeR	Centre for home energy research
ESD	Energy storage density
HVAC	Heating ventilation and air conditioning
GAB	Gugenheim Anderson De-Boer
gPROMS	General process modelling system
H_2O	Water
L/D	Length to Diameter
RH	Relative humidity
SLPM	Standard liters per minute
TES	Thermal energy storage
TRNSYS	Transient system simulation tool
V	Volume

3.1. Introduction/Background

The global average atmospheric carbon dioxide concentration has risen by 50% over the past 200 years [2]. This increase corresponds to a 1.1°C rise above previous average global temperatures in 1850-1900 [3]. If the average temperature change rises past 1.5°C, the human population will face unprecedented climate-related risks and weather events [4]. Currently, over half (53.5%) of the energy consumed by the space heating sector in Canada was supplied by fossil fuels [1]. Because of the need to reduce fossil fuel emissions and reach 2050 net zero goals a reduction in greenhouse gas emissions is required. To reduce these emissions the world is starting to "decentralize, decarbonize, and democratize" the energy grid, moving away from fossil fuels [5]. The most attractive option to reduce these emissions is the use of renewable energy sources. Since renewable energy generation generally depends on factors outside of our control such as wind speed, solar radiation, and temperature, different energy storage and conversion technologies must be developed to help with low energy supply times. Since these weather conditions are not always optimal, there may not be enough energy supply to balance out the demands of the grid. This is where energy storage devices come into play.

When there is an excess of renewable energy, this energy can be stored and later discharged when energy supply is low ensuring reliable energy supply to the end user. To help address this problem, an adsorptive thermochemical energy storage device is explored to store and later release waste heat. There are three main types of thermal energy storage systems: sensible, latent, and thermochemical energy storage processes. Sensible heat storage utilizes the sensible heat of a material to store heat. Latent heat storage utilizes the latent heat of a material to store heat in a phase change. Thermochemical energy storage utilizes a reversible chemical reaction/interaction to store heat. Because sensible and latent heat are heavily dependent on the temperature of the material their efficiency is heavily dependent on the insulation that these batteries are encompassed by. Because thermochemical energy storage utilizes a reversible chemical process its efficiency isn't hindered by this insulation nearly as much. As long as the reactants/components are kept away from each other, the energy is effectively stored forever with no reduction in the amount of stored energy. This gives adsorptive thermochemical energy storage the highest storage efficiencies than sensible and latent heat storage. The main process behind thermochemical energy storage is the exothermic nature of adsorption. Adsorption processes work using mass transfer where matter from one phase gets adsorbed/sticks to a solid

in another phase. Physical adsorption is exothermic in nature, which means when an adsorbate (the matter that sticks to the solid) changes phase from the fluid phase to the adsorbed phase, heat is released. This released heat is the main goal in this process. The thermal battery developed utilizes water as the adsorbate and a solid adsorbent. Design of these batteries can be simple. A solid adsorbent is packed into a cylindrical column and humid air is passed over the adsorbent releasing heat via water vapor adsorption that can then be used to space heat.

The focus in this chapter is to implement the model previously developed in gPROMS into Fortran for use in a TRNSYS (Transient System) simulation tool simulation to simulate the performance of this thermal battery in home heating scenarios.

3.2. Fortran

To model the performance of the thermal battery in real space heating scenarios TRNSYS (Transient System) simulation software was used. TRNSYS is a program used primarily in the fields of renewable energy and building simulation to assess and predict the performance of different solutions in space heating/cooling scenarios and renewable energy infrastructure development. In this study, TRNSYS was used to assess the performance of a thermal battery in space heating scenarios.

TRNSYS has many different "Types" pre-coded for use in simulations. Types are components in TRNSYS that perform an action, such as furnaces, fans, solar panels, etc. These types connect to each other and build off of each other to make up a larger simulation. Because the focus of this research is on adsorptive thermo-chemical energy storage, a Type used to model this technology is needed. Unfortunately, TRNSYS does not have a native Type that simulates this type of technology. Because of this, a custom Type (Type299) was coded based on the model described in this section. Since TRNSYS was coded in Fortran, it is easier to develop a new Type in Fortran using the fitted parameters from gPROMS, compared to C++ and Python. To properly code a new Type in TRNSYS, the structure of the code and required functions must follow a preset format developed by TRNSYS developers. Some of the main functions that must be called gives the thermal battery code insight on what is happening in the rest of the simulation.

The first function call "getIsFirstCallofSimulation()" tells TRNSYS how many parameters are used in the simulation, how many time dependent inputs are there, and how many time dependent outputs are there. Additionally, this call also tells TRNSYS the units used by the thermal battery code so TRNSYS can handle any necessary unit conversions so proper units are supplied to Type299. The next call is "getIsStartTime()" that passes all of the initial values of the simulation from TRNSYS to the Type299 code and initializes it. Additionally, "getIsStartTime()" defines the ranges where parameters and input values are valid for the Type299 code. Now that the model is initialized and ready for simulation, the code will receive time dependent inputs at each time step, check their validity, and simulate the requested time step by TRNSYS and pass along the outlet values by calling the "setOutputValue()" function. This code is outlined in Appendix B.1.

Type299 was coded with two modules, the first module interacts with TRNSYS and controls the execution of the second module and reports the results back to the main simulation. The second module contains all necessary equations and algorithms to solve the set of differential equations that makeup the model.

The model developed for this technology was adapted from Cooney 1974 and Ruthven 1984 [6,7]. The model assumes that air is not adsorbed and is treated as a carrier gas, as the adsorbent readily and preferentially adsorbs water vapor over air. The next assumption made is that of constant gas velocity. Because moisture in the air stream is only a small fraction of the overall gas stream, velocity changes associated with water vapor adsorption can be neglected. The next assumption made is that there is no pressure drop across the adsorbent bed which is validated through experimental results showing a negligible pressure drop. It is also assumed that there is no radial changes across the column due to the high superficial gas velocity allowing for good radial mixing throughout the column. Additionally, it is assumed that water adsorption is instantaneous on the pellet surface which is justified knowing the adsorption rate on the pellet surface is very high compared to the diffusion through the micropores [8]. Finally, it is assumed that the system behaves adiabatically, that is, no heat transfer takes place from the system to the column wall as well as to the surroundings, which is justified by the amount of insulation used in the experimental setup. The equations used to model the water vapor adsorption process are depicted in equations (3.1-3.4).

$$-D_L \frac{\partial^2 C}{\partial z^2} + \frac{\partial C}{\partial t} + v \frac{\partial}{\partial z}(C) + \frac{\hat{\rho}_s}{\tilde{\rho}_f}\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{\partial \bar{q}}{\partial t} = 0 \tag{3.1}$$

$$\frac{\partial \bar{q}}{\partial t} = k_m a\,(q^* - \bar{q}) \tag{3.2}$$

$$-\frac{k_g}{\hat{c}_f \hat{\rho}_f \varepsilon}\frac{\partial^2 \mathrm{T}_f}{\partial z^2} + \frac{\partial T_f}{\partial t} + v\frac{\partial T_f}{\partial z} + \frac{\hat{\rho}_s \hat{c}_s}{\tilde{\rho}_f \tilde{c}_f}\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{\partial T_s}{\partial t} + \frac{\Delta\tilde{H}\hat{\rho}_s}{\tilde{\rho}_f \tilde{c}_f}\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{\partial \bar{q}}{\partial t} = 0 \tag{3.3}$$

$$\frac{\partial T_s}{\partial t} = \frac{ha}{\hat{\rho}_s \hat{c}_s}\left(T_f - T_s\right) + \frac{\left(-\Delta\tilde{H}\right)}{\hat{c}_s}\frac{\partial \bar{q}}{\partial t} \tag{3.4}$$

In the equations presented above parameters denoted with a tild (Ã) indicate units of mole basis, parameters denoted with a circumflex (Â) indicate units of mass basis, and no accents denotes SI units. Equation (3.1) presented above is the mass balance on the moist air. This equation includes terms for axial diffusion, accumulation, convection, and mass transfer into the adsorbed phase due to adsorption to model how the concentration changes at each point along the column. The mass transfer due to adsorption presented in equation (3.2) is modeled as a linear driving force mass transfer model where the mass transfer coefficient k_m is estimated using gPROMS parameter estimation tool to fit the model to experimental breakthrough curves. More information on the gPROMS simulation is outlined in chapter 2. The equilibrium isotherm behaviour of water vapor adsorption on the solid adsorbent used is modeled using the Guggenheim-Anderson-de Boer (GAB) isotherm model. Equation (3.3) models the temperature change of the process fluid (moist air) and includes terms for axial diffusion, accumulation in the fluid phase, a convection term, accumulation in the solid phase, and heat generation due to adsorption. The term representing heat transfer from the solid is depicted in equation (3.4) and uses a heat transfer coefficient obtained from dimensionless correlations found in literature [9]. The model shows good agreement with the experimental data collected. The model successfully demonstrated the effect that length to diameter ratio and column volume has on reported energy storage densities (ESD). The model was able to predict the energy storage density with 85-92% accuracy with an average of 88%. More information on the mathematical model used and conducted experiments is presented in Chapter 2.

To solve the set of differential equations the spacial derivatives must be discretized so axial variations could be captured. To discretize the set of differential equations, the central finite difference method was applied as shown in equations (3.5) and (3.6).

$$\frac{\partial^2 x_i}{\partial z^2} = \frac{x_{i-1} - 2x_i + x_{i+1}}{dz^2} \tag{3.5}$$

$$\frac{\partial x_i}{\partial z} = \frac{x_{i+1} - x_{i-1}}{2dz} \tag{3.6}$$

This discretization is applied at each point along the column. As can be seen from the above discretized spacial derivatives, each spacial derivative depends on the state of each neighboring points. In Fortran, each point in space along the column is defined as an element in an array. Because of this the boundaries must be treated differently. In this simulation ghost

boundary conditions are applied at the inlet and the outlet of the column which defer from the boundary conditions used in Chapter 2. Since there is not a defined point before element 0 in the array at the inlet of the column, a ghost point must be created that holds the value of inlet conditions. Similarly, at the outlet of the column, a ghost point must be created representing the outlet conditions. This allows the same discretization to be applied at the inlet and outlet of the column. These boundary conditions were used instead in Fortran as the boundary conditions used in gPROMS were providing instability to the model during initialization and startup. To solve the set of differential equations Eulers method is applied to step through time as shown in equation (3.7).

$$y_{i+1} = y_i + dt * \frac{dy_i}{dt} \tag{3.7}$$

In this equation, y is an array of the state at each point in the column. Because of the amount of calculations the code must do in each time step to calculate unknown parameters, less important parameters are not calculated each time step but every 500 seconds to help with the speed of simulation while still retaining the accuracy of the simulation. This is because stability is only achieved with small time steps of 5 seconds or less while the entire simulation runs for weeks. Since this code is used in a larger simulation to simulate many days of performance of the thermal battery, the code must run quickly to avoid long simulation times. Despite the small amount of spacial intervals used (75 spacial intervals) Eulers method still provides an accurate simulation of the thermal battery matching the results of the gPROMS simulation as shown in Figure 3.1. The gPROMS Fortran validation presented in Figure 3.1 was simulated using column G as referenced in Chapter 2 with a length of 19 cm and internal diameter of 2.75cm.

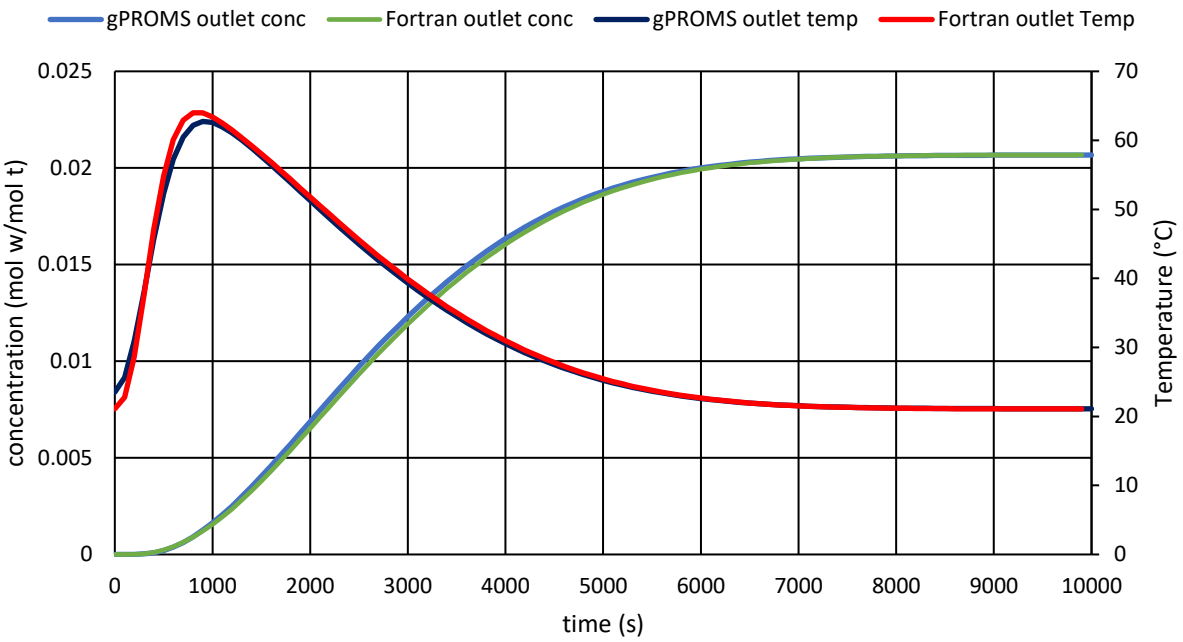


Figure 3.1:gPROMS vs Fortran Outputs

3.3. Designed simulation

The designed simulation is presented in Figure 3.2 and Figure 3.3. The simulation includes a building model which will be described in more detail in Section 3.4, as well as the designed heating ventilation and air conditioning (HVAC) system including the tested thermal battery. The HVAC system includes a supplemental furnace, air conditioning unit for summer months, thermal battery, evaporated tube collector, thermostat, blowers, and a humidifying bath.

The air conditioning unit (Type964) was sized as a 2.5 tonne air conditioning unit based on recommended sizing for a 1500 ft^2 home in Canada [10]. The furnace (Type 967) was sized as a 90000 btu/hr furnace based on recommended sizing for a 1500 ft^2 home in Canada [10]. The size of the thermal battery was estimated through trial and error to see which sizes can support running times of a week or more. It was found that a thermal battery 7m long and 2m in diameter was able to support home heating between the hours of 10am and 3 pm for approximately 5 days. This means a very large $22m^3$ thermal battery can only support 25 hours of operation under a flowrate of 1500kg/hr air at 90% relative humidity. 1500kg/hr was chosen as the flowrate in the HVAC system as it provided a good heat flow to the home balancing out the thermal losses from the house. This number was obtained through trial and error with the simulation as well as HVAC heuristics obtained online [11].

The evacuated tube collector (Type71) was sized as 2 evacuated tube collectors in series with 30 tubes at 1.8m each for a total of 60 tubes and 108 m^2 of collector area. The evacuated tube collector works by concentrating solar energy on a fluid stream to heat the stream up. In this case an air stream is used as the working fluid. The blower (Type744) was sized to handle a flow of up to 2000 kg/hr where the HVAC system utilizes a flow of 1500 kg/hr. The humidifier (Type641) was sized using online tools to provide 2.65 kg of water per hour to the thermal battery air stream [12]. Finally, the thermostat (Type166) was set to control the temperature using a 2°C deadband with a heating set point of 20°C and a cooling set point of 23°C.

Presented in Figure 3.2 is the designed HVAC system to encompass the thermal battery. In the figure the dashed lines represent information being passed along to the Type65's for use in graphing. Pink lines represent information on the current state of the house to components for use in calculations. Blue lines represent "cold" streams whereas red lines represent "hot" streams. Green lines represent control signals and logical information for control of the HVAC system.

Finally, black lines represent intermediate information transfer between components not directly related to the main scope of the simulation but are necessary for automatic simulation and unit conversions. Because of how TRNSYS is coded, only one component can be connected to each value. For example, you can only have one stream dictating the inlet flow conditions to the house. Because of the multiple components that dictate these conditions (thermal battery, furnace, air conditioning) their outlets must be combined to one stream before entering the house. To do this, Type148b is used to combine flows so that only one value is given to the building model. This works because during simulation only one unit is active at each time. For example, if the thermal battery is on and the air conditioner and furnace are off and their flows are set at 0. When combining flows in Type148b, if the furnace and air conditioning flows are set to 0 the output of Type148b is the outputted conditions of the thermal battery. Also included in the simulation is Type15-6, Type517, Type56, and Equa. Type 15-6 is weather data collected in Ottawa, Ontario, Canada. This weather data passes information to many components so that seasons can be accurately simulated.

Type517 is a scheduler which dictates the hours the thermal battery should be on. In this case, 10am to 3pm was chosen for 5hours of operation per day. Type 56 is the building model which models all of the heat and mass (moisture) transfer to and from the house, and it is discussed in more detail in section 5. Finally, the last component is Equa, which stands for Equation. This component is used for some logical control following user defined equations as well as unit conversions using user defined equations.

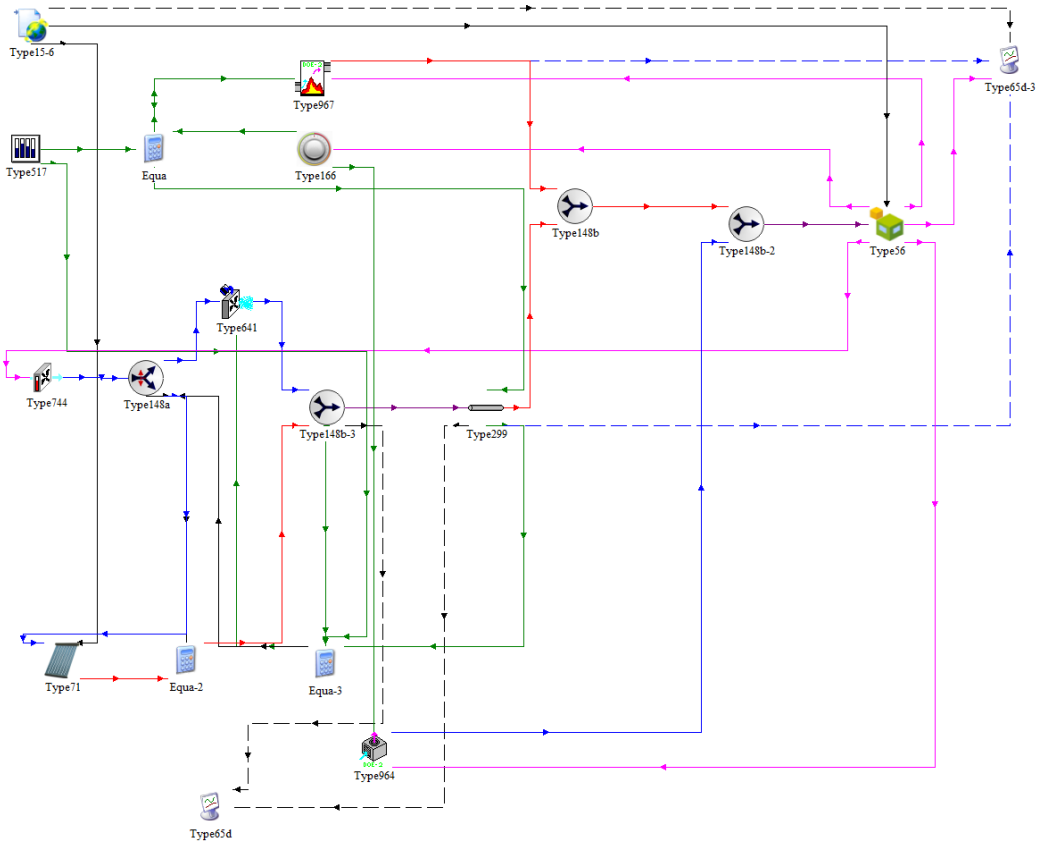


Figure 3.2: Designed HVAC System

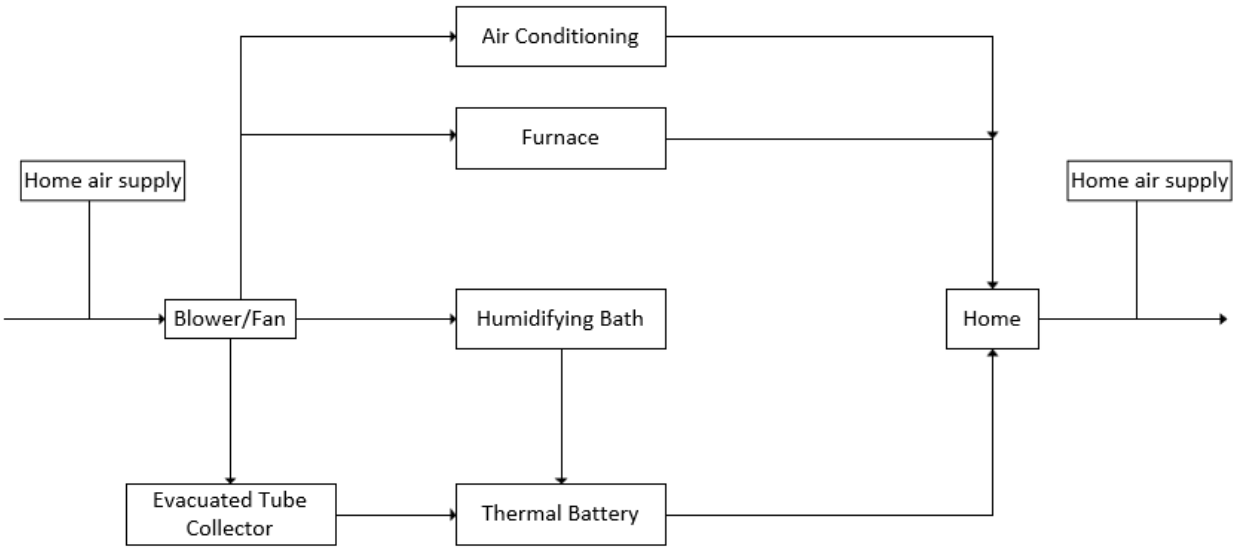


Figure 3.3: Simplified Designed HVAC System

3.4. Building Model

The building model is designed as a standard 1500 ft^2 bungalow with a basement in Ottawa, Ontario, Canada. Weather data is obtained from TRNSYS that was collected in Ottawa. The material of construction for the building model was adapted from the Carleton University CHEeR house (Urbandale Centre for Home Energy Research) [13]. The materials of construction is an important parameter in building modeling, as it defines how heat and mass (moisture) is transferred to and from the house. The windows are modeled as double pane windows for their superior insulation properties. The window model was selected from a predefined double pane model inside TRNSYS libraries. The materials of construction used to model each part of the house including walls, flooring, roofing, and ceilings are presented in Table 3.1. Each massive material of construction has its own corresponding thickness which adds together to get the total wall thickness. Some components in the TRNSYS library are treated as massless and does not have a corresponding thickness, only an overall heat transfer coefficient is used in heat transfer calculations. This means that some of the thicknesses does not represent the actual thickness of the construction type. The material of construction and their properties were also all obtained from predefined components in the TRNSYS libraries.

Also included in the building model are some gains/losses. Because no house is perfectly insulated there are some heat losses associated with infiltration (asides from heat transfer through building construction). Infiltration rate is the amount of air exchanged from inside the house with the outside over the course of one hour. In this case a standard infiltration rate of 0.5/hr is used. Meaning over one hour half of the air inside the house is exchanged with the outside. In addition to infiltration losses, there are some heat gains as well. The gains considered were lighting, people, electronics, and appliances. These gains were implemented based on models predefined in the TRNSYS library based on floor area. These models use correlations for heat gains based on referenced floor area and are obtained by conducted experiments from TRNSYS developers. In addition to modeling heat and mass transfer from the outside air to the inside, heat transfer is also taken into account from solar irradiance through windows and roofing.

Table 3.1: Material of Construction of Building Model

Construction	Material of Construction	Thickness (m)
Walls	Drywall, R-24 Fiberglass Batt, ½ inch plywood, R-24 Fiberglass Batt, Wood Siding	0.343
Floor/Basement Ceiling	Timber Flooring, 2x4 Studd Wall, R-51 Fiberglass Insulation, 2x4 Studd Wall, Dry Wall	0.127
Ceiling	Drywall, 2x4 Studd Wall, R-51 Fiberglass Insulation, 2x4 Studd Wall, Timber Flooring	0.127
Basement Floor	Concrete Slab, Foam Insulation, Sand/Gravel Bed	0.280
Basement Wall	Drywall, Fiberglass Batt, 2x4 Studd Wall, Foam Insulation, Concrete Slab	0.515
Roofing	Drywall, 2x4 Studd Wall, R-51 Fiberglass Insulation, R-24 Fiberglass Batt, ½ inch Plywood, Bitumen, Asphalt Tiles	0.228

3.5. Results/discussion

Results of the TRNSYS simulation for the week of January 1st to January 7th is presented in Figure 3.4. In this figure, the pink line represents the outside temperature, the red line represents the main floor temperature, and the blue line represents the basement temperature.

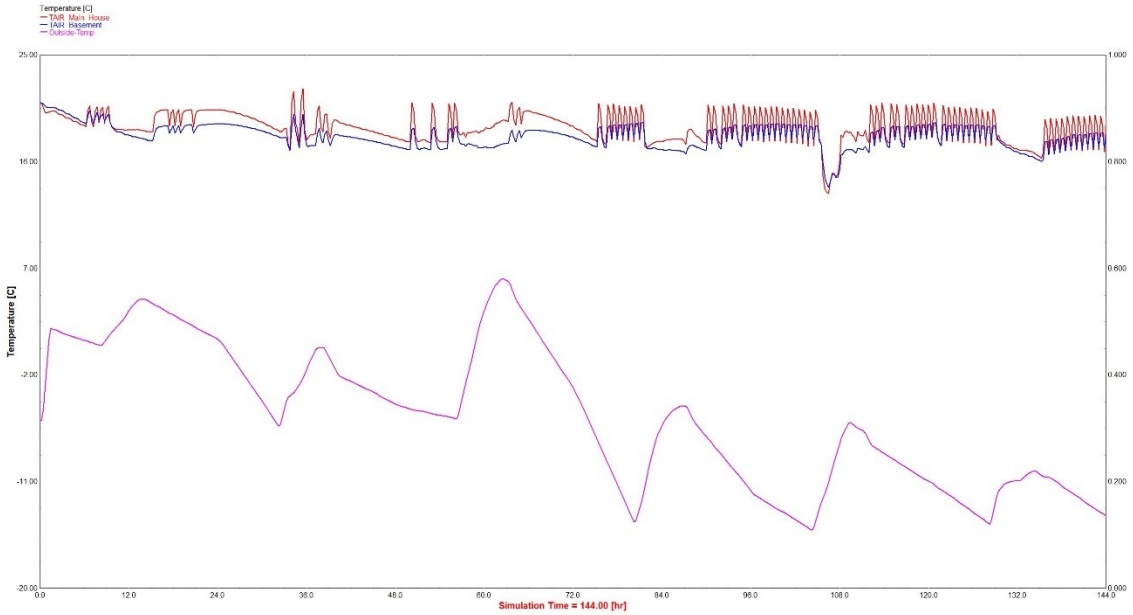


Figure 3.4: January 1st to 7th TRNSYS Simulation

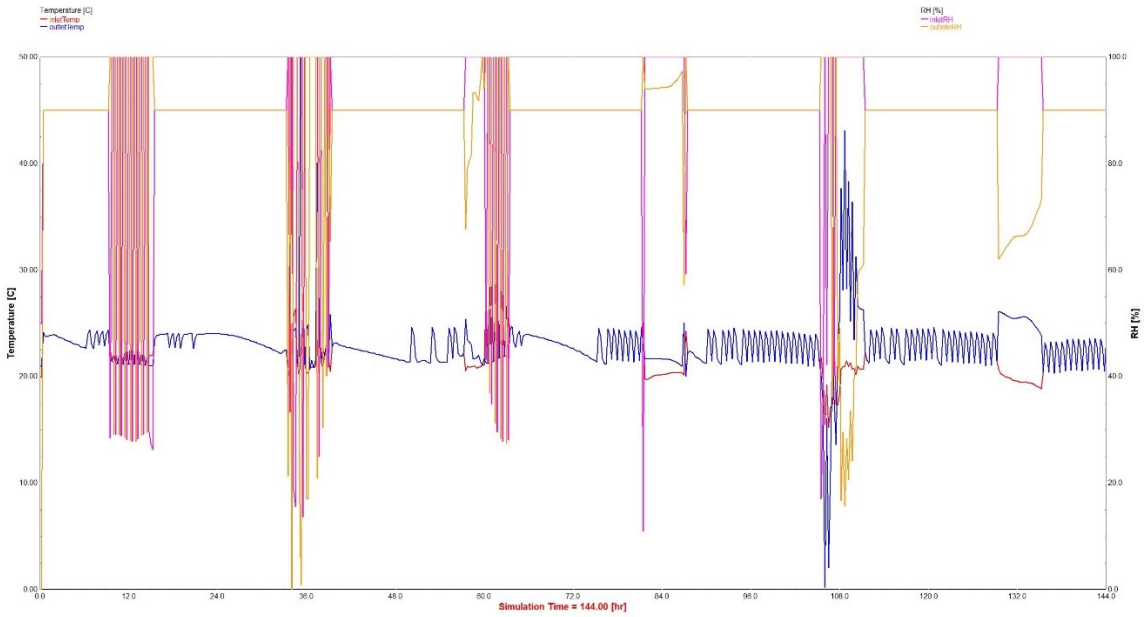


Figure 3.5: Thermal Battery Performance January 1st to 7th

Presented in Figure 3.5 is the performance of the thermal battery over the course of January 1st -7th. The red line represents the inlet temperature, the blue line represents the outlet temperature, the pink line represents the inlet relative humidity, and the orange line represents the outlet relative humidity. The oscillation in temperatures seen on colder days is the on/off cycling of the thermal battery as well as the furnace. This is normal as the furnaces turn on and off more to meet the required temperature.

As can be seen from the aforementioned figures the thermal battery took 6 days to become fully saturated with moisture to the point where minimal heat could be recovered and its heating performance diminished substantially requiring regeneration. Using evaporated tube collectors in the winter months becomes problematic where temperatures above 90°C are only obtainable at most a couple hours a day and other regenerating methods would be needed to extend the operation of this battery throughout the winter months. A promising option to regenerate the thermal battery in the winter months includes solar panels, Li-ion batteries, and resistive heaters. However, this method raises the question of why not replace the thermal battery itself for this technology curving the need for a high capital cost of the thermal battery system.

The cost of the HVAC system excluding the thermal battery, installation, and ducting is estimated to be $8303.28 CAD. This cost is based on adding up the cost of the furnace, air conditioning unit, humidifier, fan, and evacuated tube collectors. The furnace costs $1472.98 CAD and the air conditioner costs $2064.9 CAD [14,15]. Now looking at the cost to supply the thermal battery with the proper air streams, the evacuated tube collectors used for regeneration cost $1906.2 CAD each, the fan costs $688 CAD, and the humidifier costs $265 CAD [16,17,18]. This amounts to a total capital cost of $4765.4 CAD excluding installation costs, ducting, and the thermal battery itself to supply the thermal battery with the proper inlet streams. Looking at the cost of the thermal battery itself shows a very high required price for the thermal battery. Based on experimental results the packing density of the adsorbent pellets in the column is on average 860.41 kg/m^3. This number was obtained by knowing the weights of adsorbent added to each column as referenced in Chapter 2 and averaging out their respective packing densities. Knowing the designed volume of the thermal battery is 22m^3 a whopping 18,929 kg of adsorbent is required to fill the column. Online results for bulk zeolite 13X is taken as a price standard to estimate the adsorbent cost with costs ranging from $1.65-$2.20 USD/kg of zeolite 13X [19]. This amounts to a cost of $31000-$42000 USD for adsorbent alone. Because of the high capital cost required

for operation of the thermal battery and the size needed to support 25 hours of operation, the thermal battery is not economically viable when instead a solar panel, battery, and resistive heater can be purchased at a much lower cost to supply heat more reliably.

3.6. Conclusions

In conclusion, a 1500 ft^2 building model was developed in TRNSYS using the Carleton University CHEeR house as a basis of material of construction which takes into account heat gains from people, lighting, appliances, and electronics. Heat losses were also considered in the form of infiltration and heat transfer through the walls of the building. The model for the thermal battery was developed and fitted in gPROMS and implemented in Fortran for use in the TRNSYS simulation. The HVAC system was designed around the thermal battery to tests its performance in real home heating scenarios. The HVAC system includes a supplemental furnace, air conditioning unit for summer months, thermal battery, evaporated tube collector, thermostat, blowers, and a humidifying bath. The thermal battery was simulated from 10am – 3pm every day from January 1^{st} – 7^{th}. Over 25 hours of operation using a flowrate of 90% relative humidity moist air at 1500 kg/hr the adsorbent bed became fully saturated. Because of the high heat flowrate required to balance out the heat losses to heat the home a large 22 m^3 adsorption column was selected for this simulation. Because the large adsorption column expired in under 1 week and the high capital cost required for the thermal battery, this technology is not currently ready for widespread adoption. The main cost for this adsorption column is the adsorbent itself costing approximately $31000-$42000 USD. Because of the high capital cost required for operation of the thermal battery and the size needed to support 25 hours of operation, the thermal battery is not economically viable when instead a solar panel, battery, and resistive heater can be purchased for much cheaper to supply heat more reliably.

3.8. References

[1] Natural Resources Canada, "Natural Resources Canada," Government of Canada, 2020. [Online]. Available: https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/showTable.cfm?type=CP&sector=res&juris=ca&rn=7&year=2020&page=3. [Accessed 02 04 2024].

[2] NASA, "Carbon Dioxide," NASA, February 2024. [Online]. Available: https://climate.nasa.gov/vital-signs/carbon-dioxide/?intent=121. [Accessed 04 04 2024].

[3] Core Writing Team, H. Lee and J. Romero, "Climate Change 2023: Synthesis Report," IPCC, Geneva, Switzerland, 2023.

[4] V. Masson-Delmotte, P. Zhai, H. Portner, D. Roberts, J. Skea, P. Shukla, A. Pirani, W. Moufouma-Okia, C. Peann, R. Pidcock, S. Connors, J. Matthews, Y. Chen, X. Zhou, M. Gomis, E. Lonnoy, T. Maycock, M. Tignor and T. Waterfields, "IPCC, 2018: Summary for Policymakers," Cambridge University Press, New York, 2018.

[5] M. Green, "Community power," Nature Energy, no. 1, 2016.

[6] D. O. Cooney, "Numerical Investigation of Adiabatic Fixed-Bed Adsorption," Industrial & Engineering Chemistry Process Design and Development, vol. 13, no. 4, pp. 368-373, 1974.

[7] D. M. Ruthven, Principles of Adsorption and Adsorption Processes, John Wiley & Sons, Inc, 1984.

[8] D. Lefebvre, P. Amyot, B. Ugur and F. H. Tezel, "Adsorption Prediction and Modeling of Thermal Energy Storage Systems: A Parametric Study," Industrial & Engineering Chemistry Research, vol. 55, no. 16, pp. 4760-4772, 2016.

[9] Z. Qi and A. B. Yu, "A new correlation for heat transfer in particle-fluid beds," International Journal of Heat and Mass Transfer, vol. 181, p. 121844, 2021.

[10] "HVAC Direct," HVAC Direct, [Online]. Available: https://hvacdirect.com/sizing-air-conditioner-and-heater.html?srsltid=AfmBOoqbRcEQy6Q85KB0A0bGd4OQKqODEGwDRyS7yIUxmVpLayWWD9gC. [Accessed 19 08 2024].

[11] "Cubic Feet Per Minute (CFM)," American Standard Heating & Air Conditioning, [Online]. Available: https://www.americanstandardair.com/resources/glossary/cfm/. [Accessed 15 09 2024].

[12] "Humidifier Calculation," Vacker, [Online]. Available: https://tools-dashboard.vackerglobal.com/humidifier-calculation. [Accessed 10 08 2024].

[13] "Building Performance Research Centre," Carleton Universiy, [Online]. Available: https://carleton.ca/bprc/people/cher-house/. [Accessed 19 08 2024].

[14] "100,000 BTU Multi-Speed Single Stage Goodman Gas Furnace," HVAC Direct, [Online]. Available: https://hvacdirect.com/goodman-100-000-btu-80-afue-multi-speed-single-stage-gas-furnace-gces801005cn.html. [Accessed 19 08 2024].

[15] "Goodman 2.5 Ton 13.4 SEER2 Air Conditioner Condenser," HVAC Direct, [Online]. Available: https://hvacdirect.com/2-5-ton-13-4-seer2-goodman-air-conditioner-condenser-gsxn3n30106.html. [Accessed 19 08 2024].

[16] "Vacuum Tube Solar Collector Kit," Hydro Solar Innovative Energy, [Online]. Available: https://hydrosolar.ca/products/xianke-xkph58-srcc-certified-evacuated-tube-solar-collector?srsltid=AfmBOooV7CjiU2NDgdDAbBmpLcvu6sHjdfowz2DTkjEWqGeHlIsRwFzI. [Accessed 19 08 2024].

[17] "A.O. Smith 9477 1000 CFM 2 Speed Centrifugal Blower," Amazon, [Online]. Available: https://www.amazon.ca/Smith-9477-Volts-Centrifugal-Blower/dp/B007ATNCPO. [Accessed 19 08 2024].

[18] "Goodman HE12MB Humidifier - CleanComfort," Canada HVAC, [Online]. Available: https://canadahvac.com/product/goodman-he12mb-humidifier-cleancomfort/. [Accessed 19 08 2024].

[19] "Adsorbent Zeolite 13x molecular sieve for CO2 removal 13x molecular sieve beads dehydration and desulphurization," Alibaba, [Online]. Available: https://www.alibaba.com/product-detail/Adsorbent-zeolite-13x-Molecular-Sieve-For_60701038274.html?spm=a2700.7724857.0.0.2b08742a4fdAtE. [Accessed 19 08 2024].

4. Conclusions

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2024/09/25

4.1. Conclusions

In conclusion, a mechanistic model was developed and fitted in gPROMS software to model the moisture vapour adsorption breakthrough curves by a solid adsorbent. The model showed good agreement with the experimental data set predicting the energy storage density (ESD) with 85-92% accuracy. Effects of length to diameter ratio and column volume showed a decreasing trend in energy storage densities with increasing column length to diameter (L/D) ratios where an increase in energy storage density is observed with increasing column volume. The decreasing trend of ESD with increasing L/D ratios can be attributed to the fact that longer columns' walls act as better heat sinks because of the higher surface area the fluid encounters allowing for more heat to be transferred through the column walls. The assumed reason that the increasing trend of ESD is observed with increasing column volume is due to the fraction of heat transferred out of the column being less than that of a smaller column. Because less total heat is released in smaller columns any heat losses from the column wall will have a bigger impact on reported ESD's than larger columns with much more total heat being released. Over the 9 conducted experiments, the average energy storage density was evaluated to be 123 kWh/m^3. Improvements to the mathematical model can be obtained by incorporating mass transfer inside the pellet at the cost of slowing down simulation times.

Using the fitted model in gPROMS the model was re-coded in Fortran for use in a TRNSY simulation to estimate the performance of the battery in real home heating scenarios. To model real home heating scenarios, a 1500 ft^2 building model in Ottawa, Ontario, Canada was designed in TRNSYS using the Carleton University CHEeR house as a basis of material of construction. The building model takes into account heat gains from people, lighting, appliances, electronics, and solar irradiance. In addition to heat gains, the model also takes into account losses through building walls, and infiltration. The HVAC system was designed in TRNSYS around the thermal battery and includes a supplemental furnace, air conditioning unit for summer months, thermal battery, evaporated tube collector, thermostat, blowers, and a humidifying bath. The thermal battery was simulated using a schedule of on between the hours of 10am – 3pm and off outside that range. The simulation was conducted using weather data from Ottawa, Ontario, Canada over the time periods of January 1^{st}-7^{th}. Because of the high heat flow required to balance the heat losses, a large 22 m^3 adsorption column was selected for this simulation. Using a flowrate of 1500

kg/hr for a 90% relative humidity moist air stream over the adsorbent bed, the thermal battery became fully depleted within the 25 hours of operation.

This large thermal battery requires a high capital cost of adsorbent in the range of $31,000-$42,000 USD. Results of the simulation showed the expiry of the thermal battery in under 1 week (25 hours of operation) requiring regeneration. Using evaporated tube collectors as a heat source in the winter months becomes problematic where temperatures above 90°C are only obtainable at most a couple of hours a day and other regenerating methods would be needed to extend the operation of this battery throughout the winter months. Because of this rapid expiry and high capital cost of the system, this technology is not currently ready for widespread adoption and more development of adsorbent materials and adsorption column designs are required. Instead, it is recommended that a solar panel accompanied with a battery and resistive heater is used because of its lower capital cost and ease of use.

A. Appendix

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2024/08/20

A.1. GPROMS Code

A.1.1. Process

```
UNIT

TESModel As TESModel

SET

WITHIN TESModel DO

   components := ['H2O','Air'];

   Axial := [ CFDM, 2, 200 ] ; #[discretization method, order of approximation,
number of interervals for distribution domain]


   Dc := 2.75336/100;

   L := 18.9911/100;

   wall := 0.81E-3;

   epsilon := 0.54;

   VolumetricFlowrate := 20.25;

   pressure := 109004.43;


   m := 331.302;


   Mm0 := 12.025908651;

   qm := 2852.562025;

   C0 := 0.227163506;

   dHC := 9294.968815;

   K0 := 1.604923623;
```

```
    dHk := -1537.830778;
END

ASSIGN
WITHIN TESModel Do

    km := 3.13904E-6;
    deltaH := -57935.8;

    InletConcentration := 0.02066940691716072;
    InletTemperature := 21.1;
END

INITIALISATION_PROCEDURE
# Start Default Initialisation Procedure
    USE
          : DEFAULT;
    END
    SAVE "TESModelRun_InitialGuess";
# End Default Initialisation Procedure
SOLUTIONPARAMETERS
DASolver := "DAEBDF" [
        "MaxCorrectorIterations" := 20,
        "MaxSuccessiveCorrectorFailures" := 30,
```

```
#       "OutputLevel" := 3
"ErrorNormComputation" := "DifferentialAndScaledAlgebraic",
   ]
SCHEDULE
# OperationSchedule
SEQUENCE
SAVE "SVS"
RESTORE "SVS"
CONTINUE FOR 10000
END
```

A.1.1. Model

PARAMETER

components As ORDERED_SET

#-----Constants-----#

PI AS REAL DEFAULT 3.14159265359 #PI

e AS REAL DEFAULT 2.718281828459 #e

g AS REAL DEFAULT 9.81 #Gravitational Acceleration, m/s^2

R AS REAL DEFAULT 8.3144621 #Ideal gas constant, Pa*m^3/mol*K or J/mol*K

#-----Bed Parameters-----#

L As Length #Column length, m

Dc as Length #Column Diameter, m

wall AS Length #Column thickness

D_o AS REAL DEFAULT Dc+2*wall

ri AS REAL DEFAULT Dc/2

ro AS REAL DEFAULT D_o/2

epsilon AS VoidFraction #Void fraction of the bed

#-----Operating Conditions-----#

VolumetricFlowrate AS VolumetricFlowrate

velocity AS REAL DEFAULT VolumetricFlowrate*4/(PI*(Dc^2)*60000) #Superficial velocity treated as constant due to low water concentration, m/s

pressure As Pressure #Pa

initialTemperature AS REAL DEFAULT 23.5+273.15

ambientTemperature AS REAL DEFAULT 294.15

RH_inlet AS REAL DEFAULT 0.9 #inlet Relative Humidity

#------Solid Parameters for pure SG-------#

m as dimensionless

#km as massTransferCoefficient

Mm0 AS dimensionless3

qm AS dimensionless3

C0 AS dimensionless3

dHC AS dimensionless3

K0 AS dimensionless3

dHk AS dimensionless3

#deltaH as dimensionless3

dp AS REAL DEFAULT 2.097E-3 #Particle Diameter, m

#-----Mass Transfer Parameters-----#

#m as REAL DEFAULT 405740

#km as REAL DEFAULT 0.000487847

#-----Heat Transfer Parameters-----#

Cps AS REAL DEFAULT 1200 #J/kg*K. Placeholder for now. Should vary with Temperature. Default 880

rho_s AS REAL DEFAULT 2668.6 #measured solid skeletal density

a AS REAL DEFAULT 653.4192*rho_s*1E-3 #/m, specific surface area (area per unit volume of particle), 300 original

#Psat_inlet AS REAL DEFAULT 1*e^(1.2378847*10^(-5.)*initialTemperature^(2.)+(-1.9121316*10^(-2.)*initialTemperature)+33.93711047+(-6.3431645*10^(3.))/initialTemperature) #inlet saturation vapour pressure

#inletGasConcentration AS REAL DEFAULT RH_inlet*Psat_inlet/pressure #mol w/mol f

#fluidThermalConductivity AS REAL DEFAULT 1.015 #Solid thermal conductivity, W/m/K

Ma AS REAL DEFAULT 28.96546*10^(-3.) #kg/mol air

Mv AS REAL DEFAULT 18.01528*10^(-3.) #kg/mol water

Ms AS REAL DEFAULT 162.05E-3 #kg/mol zeolite

collisionDiameter AS REAL DEFAULT 0.5*(3.617+2.725) #need leonard jones potential of water, used 2x air here

kamb AS REAL DEFAULT 0.02534 #% W/m.K; thermal conductivity of air

nuamb AS REAL DEFAULT 0.00001455 #% m^2/s; air kinetic visc at 288.15K=15C

Alphamb AS REAL DEFAULT (15.57+(22.07-15.67)*(ambientTemperature-250)/50)*(10^(-6)) #% m^2/s; thermal diffusivity of air:

Pramb AS REAL DEFAULT 0.71 #% --; Prandlt number of ambient air outside column;

#-----Wall Heat Transfer Parameters-----#

rhow AS REAL DEFAULT 8000 #% kg/m^3; density of SS

Cpw AS REAL DEFAULT 500 #% J/kg.K; heat capacity of SS

kw AS REAL DEFAULT 16.2 #% W/m.K; thermal conductivity of SS

DISTRIBUTION_DOMAIN

Axial AS [0 : L]

PORT

VARIABLE

gasConcentration AS DISTRIBUTION(Axial) OF GasConcentration #Concentation of Water in the air carrier gas, mol/m3

amountAdsorbed,q AS DISTRIBUTION(Axial) OF AmountAdsorbed #Averaqe amount adsorbed, mol/m3

fluidTemperature,solidTemperature AS DISTRIBUTION(Axial) of Temperature

#wallTemperature AS DISTRIBUTION(Axial) of Temperature

Sh AS DISTRIBUTION(Axial) OF Dimensionless

Sc AS DISTRIBUTION(Axial) OF Dimensionless

Re AS DISTRIBUTION(Axial) OF Dimensionless

Pr AS DISTRIBUTION(Axial) OF Dimensionless

Nu_0 AS DISTRIBUTION(Axial) OF Dimensionless

beta_ AS DISTRIBUTION(Axial) OF Dimensionless2

Nu_p AS DISTRIBUTION(Axial) OF Dimensionless

#Nuw AS DISTRIBUTION(Axial) OF Dimensionless

rho_g AS DISTRIBUTION(Axial) OF Density

CompressabilityFactor AS Distribution(Axial) OF Dimensionless

x_v AS Distribution(Axial) OF MoleFraction

p_H2O AS Distribution(Axial) OF Pressure

#p_sv AS Distribution(Axial) OF Pressure

mu_w,mu_a,mu_g AS Distribution(Axial) OF Viscocity

phi_ww,phi_wa,phi_aa,phi_aw AS DISTRIBUTION(Axial) OF Dimensionless

k_w,k_a,k_g AS Distribution(Axial) OF ThermalConductivity

Tstar AS Distribution(Axial) OF Temperature

CollisionIntegral AS Distribution(Axial) of Dimensionless

MolecularDiffusivity AS Distribution(Axial) of Diffusivity

MassTransferCoefficient AS Distribution(Axial) of MassTransferCoefficient

HeatTransferCoefficient_solid_fluid AS Distribution(Axial) of HeatTransferCoefficient

#HeatTransferCoefficient_fluid_wall as Distribution(Axial) of HeatTransferCoefficient

#HeatTransferCoefficient_wall_ambient as Distribution(Axial) of HeatTransferCoefficient

Cp_g,Cp_w,Cp_a AS Distribution(Axial) of HeatCapacity

humidity AS Distribution(Axial) of MassFraction

Mf AS Distribution(Axial) of MolecularWeight

ConcentrationBreakthrough AS Dimensionless

TemperatureBreakthrough AS TemperatureC

D_L AS Distribution(Axial) of Diffusivity

OutletConcentration as GasConcentration

InletConcentration as GasConcentration

InletTemperature as TemperatureC

#Rao,Nuo AS Distribution(Axial) of Dimensionless

Mm as distribution(Axial) of dimensionless

C as distribution(Axial) of dimensionless

K as distribution(Axial) of dimensionless

km as massTransferCoefficient

deltaH as dimensionless3

ESD as EnergyDensity

#Dc as Length

#L as Length

#wall as Length

#epsilon as VoidFraction

#VolumetricFlowrate as VolumetricFlowrate

#Pressure as Pressure

SELECTOR

cal_mode AS (initialize, calculate) DEFAULT calculate

BOUNDARY

Boundary condition equations

#-------------------Fluid Phase Mass Balance Boundary Conditions---------------------------#

CASE cal_mode OF

WHEN initialize:

PARTIAL(gasConcentration(0),Axial) = (velocity/epsilon)*(gasConcentration(0) - InletConcentration)/0.00603;

WHEN calculate:

```
   PARTIAL(gasConcentration(0),Axial) = (velocity/epsilon)*(gasConcentration(0) -
InletConcentration)/D_L(0);

END


PARTIAL(gasConcentration(L),Axial) = 0; #No concentration change at the outlet as no
adsobent to adsorb


#--------------------Fluid Phase Heat Balance Boundary Conditions----------------------------#

CASE cal_mode OF

    WHEN initialize:

      PARTIAL(fluidTemperature(0),Axial) =
1.20515*30.7631*(velocity/epsilon)*(fluidTemperature(0) -
(InletTemperature+273.15))/0.0255308;

    WHEN calculate:

      PARTIAL(fluidTemperature(0),Axial) =
rho_g(0)*Cp_g(0)*(velocity/epsilon)*(fluidTemperature(0) -
(InletTemperature+273.15))/k_g(0);


END


PARTIAL(fluidTemperature(L),Axial) = 0;


#wallTemperature(0) = (initialTemperature+ambientTemperature)/2;

#PARTIAL(wallTemperature(L),Axial) = 0;



EQUATION

# Model equations

#-Fluid Property Determination-#
```

```
#Density

#p_sv=1*e^(1.2378847*10^(-5.)*fluidTemperature^(2.)+(-1.9121316*10^(-
2.)*fluidTemperature)+33.93711047+(-6.3431645*10^(3.))/fluidTemperature);

#x_v=relativeHumidity*p_sv/pressure;


#p_H2O=gasConcentration*R*fluidTemperature;#partial pressure of water

p_H2O=gasConcentration*pressure;

#x_v = p_H2O/pressure;

x_v=gasConcentration;

CompressabilityFactor=1-pressure/fluidTemperature*( 1.58123*10^(-6.) + (-2.9331*10^(-
8.)) * fluidTemperature + (1.1043*10^(-10.))*fluidTemperature^(2.) + (5.707*10^(-6.)+(-
2.051*10^(-8.)*fluidTemperature))*x_v + (1.9898*10^(-4.)+(-2.376*10^(-
6.))*fluidTemperature)*x_v^(2.))+pressure^(2.)/(fluidTemperature^(2.))*(1.83*10^(-11.)+(-
0.765*10^(-8)*x_v^(2.)));

rho_g=pressure*Ma/(CompressabilityFactor*R*fluidTemperature)*(1-x_v*(1-Mv/Ma));


#Viscocity

mu_w=1.12*10^(-5.)*(fluidTemperature/350.)^(1.15);

mu_a=1.716*10^(-5.)*(fluidTemperature/273.)^(0.666);

phi_ww=(1./(8.^(1./2.)))*((1+Mv/Mv)^(-
0.5))*(1+((mu_w/mu_w)^(0.5))*((Mv/Mv)^(0.25)))^(2.);

phi_wa=(1./(8.^(1./2.)))*((1+Mv/Ma)^(-
0.5))*(1+((mu_w/mu_a)^(0.5))*((Ma/Mv)^(0.25)))^(2.);

phi_aa=(1./(8.^(1./2.)))*((1+Ma/Ma)^(-
0.5))*(1+((mu_a/mu_a)^(0.5))*((Ma/Ma)^(0.25)))^(2.);

phi_aw=(1./(8.^(1./2.)))*((1+Ma/Mv)^(-
0.5))*(1+((mu_a/mu_w)^(0.5))*((Mv/Ma)^(0.25)))^(2.);

mu_g=x_v*mu_w/(x_v*phi_ww+(1-x_v)*phi_wa)+(1-x_v)*mu_a/(x_v*phi_aw+(1-
x_v)*phi_aa);


#Thermal Conductivity
```

```
    k_w=0.0181*(fluidTemperature/300.)^(1.35);
    k_a=0.0241*(fluidTemperature/273.)^(0.81);
    k_g=x_v*k_w/(x_v*phi_ww+(1-x_v)*phi_wa)+(1-x_v)*k_a/(x_v*phi_aw+(1-x_v)*phi_aa);

#-Mass Transfer coefficient determinations-
TStar=fluidTemperature/((97*356)^0.5);
collisionIntegral=1.06036/(TStar^0.15610)+0.193/(e^(0.47635*TStar))+1.03587/(e^(1.52996*
TStar))+2*1.76474/(e^(3.89411*TStar));
MolecularDiffusivity =
(0.001858*(fluidTemperature^(3./2.))*(((1/(Ma*10^3))+(1/(Mv*10^3)))^(0.5))/((pressure/1013
25.)*(collisionDiameter^(2.))*collisionIntegral))/10000; #m^2/s

Sc=mu_g/(rho_g*MolecularDiffusivity);#Diffusivity is used here however knudsen diffusion is
ignored and only molecular diffusivity used here, see adsorption assignment 5
Re=rho_g*velocity*dp/mu_g;

Sh=2.0+1.1*Sc^(1./3.)*(Re)^(0.6);#superficial velocity used here

#CASE cal_mode OF
    #WHEN initialize:
      #MassTransferCoefficient=Sh*molecularDiffusivity/(dp*inletGasConcentration);
    #WHEN calculate:
      #IF gasConcentration = 0 THEN
         #MassTransferCoefficient=0;
      #Else
         MassTransferCoefficient=((m/(Sh*molecularDiffusivity/(dp)))+(1/km))^(-
1);#MassTransferCoefficient=Sh*molecularDiffusivity/(dp*gasConcentration); #units of m/s
      #End
#END
```

D_L = (molecularDiffusivity/epsilon)*(20 + 0.5*Re*Sc) ;#(velocity*dp)*(20/(Re*Sc) + 0.5) ; #Ruthven section 7 eq 7.7/7.11-> not working because of div by 0 at boundary conditions. Flower used something similar. not sure where she got it from = Wakao

#-Particle Fluid Heat Transfer Coefficient Determination-#

humidity=Mv*x_v/(Mv*x_v+Ma*(1-x_v));

Cp_a=R*(3.55+0.575E-3*fluidTemperature-0.016E5*fluidTemperature^(-2));

Cp_w=R*(3.470+1.450E-3*fluidTemperature+0.121E5*fluidTemperature^(-2));

Cp_g=x_v*Cp_w+(1-x_v)*Cp_a;

Pr=mu_g*Cp_g/k_g;

Nu_0=2+0.478*(Re^0.48)*(Pr^0.45)+0.012*(Re^1.1)*(Pr^0.78);

beta_=(3.92*epsilon^2+0.34*epsilon+10.3)-12*e^(0.38*(epsilon^2)*log(Pr))+(3.13*epsilon^3-5.43*epsilon^2)*log(Re);

Nu_p=Nu_0*((Re*Pr)^(1-epsilon))*epsilon^(-beta_); #nusselt number between fluid and adsorbent

HeatTransferCoefficient_solid_fluid=Nu_p*k_g/dp;

#HeatTransferCoefficient_solid_fluid=k_g*4.36/Dc;

#Nuw = 12.5 + 0.048*Re;

#HeatTransferCoefficient_fluid_wall = k_g*Nuw/dc;

#Rao = (wallTemperature - ambientTemperature)*g*(D_o^3)/ambientTemperature/nuamb/Alphamb ;

#Nuo = 0.68 + (0.67*(Rao^(1/4)))/(1 + (0.492/Pramb)^(9/16))^(4/9) ;

#HeatTransferCoefficient_wall_ambient = kamb*Nuo/D_o ;

Mf=(1-x_v)*Ma+x_v*Mv;#Molecular weight of the fluid kg/mol

```
Mm=Mm0*exp(qm/(R*fluidTemperature));

C=C0*exp(dHc/(R*fluidTemperature));

K=K0*exp(dHk/(R*fluidTemperature));


q=Mm*C*K*(p_H2O/pressure)/((1-K*(p_H2O/pressure))*(1-
K*(p_H2O/pressure)+C*K*(p_H2O/pressure)));

$amountAdsorbed=(MassTransferCoefficient*a)*(q-amountAdsorbed);

$solidTemperature= (HeatTransferCoefficient_solid_fluid*a/(rho_s*Cps))*(fluidTemperature-
solidTemperature)+(-deltaH/Cps)*$amountAdsorbed;

FOR z:= 0|+  TO L|- DO


   $gasConcentration(z) =  D_L(z)*PARTIAL(gasConcentration(z),Axial,Axial) #diffusive


                  -(velocity/epsilon)*PARTIAL(gasConcentration(z),Axial) #convective


                  -(rho_s/(rho_g(z)/Mf(z)))*((1-epsilon)/epsilon)*$amountAdsorbed(z);
#consumption




   $fluidTemperature(z) =
(k_g(z)/(Cp_g(z)/Mf(z))/rho_g(z)/epsilon)*PARTIAL(fluidTemperature(z),Axial,Axial)
#diffusive


                  -(velocity/epsilon)*PARTIAL(fluidTemperature(z),Axial) #convective


                  -(rho_s/(rho_g(z)/Mf(z)))*(deltaH/Cp_g(z))*((1-
epsilon)/epsilon)*$amountAdsorbed(z) #generative


                  -(rho_s/(rho_g(z)/Mf(z)))*(Cps/(Cp_g(z)))*((1-
epsilon)/epsilon)*$solidTemperature(z); #convective fluid and solid
```

```
                #- 2*HeatTransferCoefficient_fluid_wall(z)*(fluidTemperature(z) -
wallTemperature(z))/epsilon/ri/rho_g(z)/Cp_g(z); #convective fluid and wall




   #$wallTemperature(z) =   kw*PARTIAL(wallTemperature(z),Axial,Axial)/(rhow*Cpw)
#conductive


                #+ 2*ri*HeatTransferCoefficient_fluid_wall(z)/((ro^2-
ri^2)*rhow*Cpw)*(fluidTemperature(z)-wallTemperature(z)) #convective fluid and wall


                #- 2*ro*HeatTransferCoefficient_wall_ambient(z)/((ro^2-
ri^2)*rhow*Cpw)*(wallTemperature(z)-ambientTemperature); #convective wall and ambient'

END


ConcentrationBreakthrough = gasConcentration(L)/InletConcentration;

#IF ConcentrationBreakthrough > 1 THEN

   #gasConcentration(L) = inletGasConcentration;

   #ConcentrationBreakthrough = 1;

#Else

   #ConcentrationBreakthrough = gasConcentration(L)/inletGasConcentration;

   #outletConcentration = gasConcentration(L);

#END


outletConcentration = gasConcentration(L);

TemperatureBreakthrough = fluidTemperature(L)-273.15;
```

```
$ESD =
(volumetricFlowrate/60/1000*rho_g(L))*(cp_g(L)/Mf(L)/1000)*(TemperatureBreakthrough-
inletTemperature)/3600/((PI/4)*(Dc^2)*L);




INITIAL

# Equations

FOR z := 0|+ TO L|- DO

    gasConcentration(z) = 0 ;

    fluidTemperature(z) = initialTemperature;

    #wallTemperature(z) = (initialTemperature+ambientTemperature)/2 ;

END


solidTemperature = initialTemperature;

amountAdsorbed = 0 ;

ESD = 0;

#q=0;

INITIALISATION_PROCEDURE init

START

  cal_mode := initialize ;

END

NEXT

  JUMP_TO

    cal_mode := calculate ;

  END

END
```

A.2. TES Manual

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2023/06/16

A.2.1. Experimental Setup

Outlined in Figure A.1below is the process flow diagram of the experimental setup used to collect data. Air is sourced from the building supply and enters at the left of the diagram. The building supply air enters the system at a pressure of 30.7psia. 2 AliCat mass flow controllers (MFC) control the inlet flow of the dry and wet air streams to achieve a desired relative humidity at the column inlet. The pressure at the outlet of the MFC's can range between 18-25 psia. The Dry air stream first enters R-100 which is an adsorptive air dehumidification column before mixing with the wet air stream. The adsorbent used to dry the dry air stream is drierite. The wet air stream bubbles through V-101 which is a humid air generation tank that is assisted by ultrasonic foggers to aid in generating humid air at higher flowrates. Immediately following the humidifying tank the wet air stream enters V-102, a water knockout vessel. The purpose of this vessel is to remove excess water from the stream to ensure no condensation and/or excess water buildup down stream of the humidifying vessel. Once excess water is knocked out in V-102 the wet air stream mixes with the dry air stream in mixing chamber V-103 to achieve the desired relative humidity. V-103 is fitted with temperature, relative humidity sensors as well as a pressure gauge. These sensors allow the calculation of water concentration in the column inlet stream for modeling purposes. Following the mixing chamber the flow is either sent through the column or through the bypass for flow development depending on the progress of the experiment. The flow direction is controlled by a 3-way diverting valve hooked up to the Lab-View software. The adsorption column R-101 is outfitted with fiberglass insulation, column skin temperature probes, as well as inlet and outlet temperature probes to ensure accurate measurement of inlet and outlet temperature differences. In addition to the sensors, R-101 is also equipped with 2 manual valves (VLV-103 and VLV-104) to ensure isolation from the rest of the system after regeneration is complete and cooling us underway. Finally, the flow enters a mock "room" V-104. V-104 is outfitted with temperature and relative humidity sensors to monitor the progress of the experiment before flow exhausts to the atmosphere.

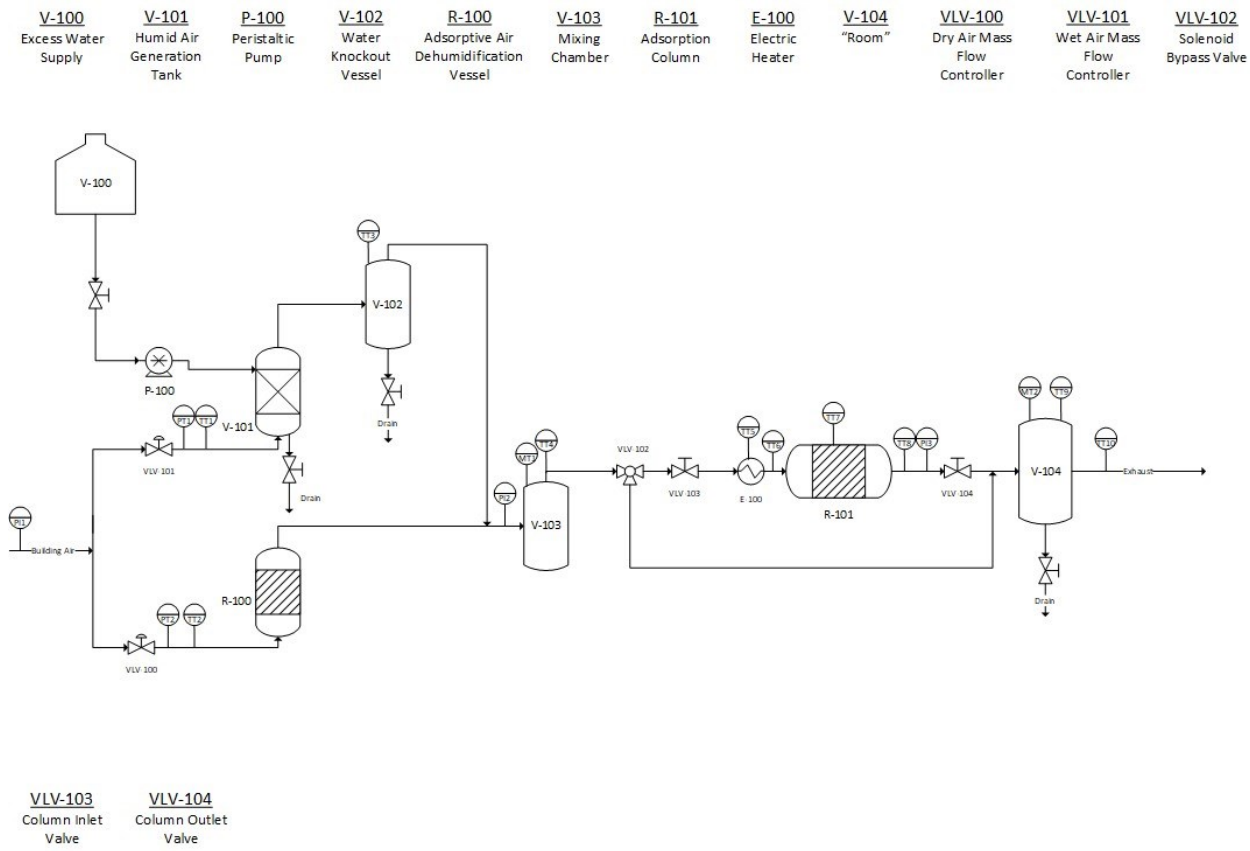
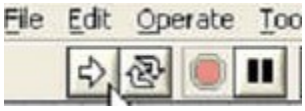


Figure A.1: PFD of TES Experimental Setup

A.2.2.1. Opening LabView

- At the top left of the screen click the run arrow and wait 10 seconds for calibration to finish
  - 

- The data file will be saved as the current date in the Glass Reactor folder.
- Begin the experiment.

A.2.2.2. Packing the Column

- Remove the insulating jacket and unravel the fiberglass insulation tape.
- Remove the temperature probe from the skin of the column.
- Remove the column module.
- Empty the column and clean with hot water.
- Dry the column using compressed air supplied from the building.
- Insert fiberglass wool into each end of the column caps if there is no metal mesh in the caps.
- Weigh the dry empty column and record the weight.
- Fill the column while ensuring even distribution of adsorbent through the column.
- Weigh the filled column and calculate weight of adsorbent added.
- Secure the column caps.

A.2.2.3. Installing the Column

- Replace temperature probe on the column skin.
- Wrap fiberglass tape around the piping.
- Install the insulating jacket.

A.2.2.4. Starting Regeneration

- Set the mode to Manual (Mode 0).
- Set the wet air flow to 0.
- Set the dry air flow to the desired operating flowrate.
- Close the bypass and open the column valve to allow dry air to pass through the column.

A.2.2.5. Starting High Temperature Regeneration

- Once the RH at the outlet of the column (RH probe 3) reaches around 6-7RH, high temperature regeneration can begin.
- Set the temperature of the column to your desired regeneration temperature and send the Set Point (SP) to the temperature controller.

- Allow the RH to decrease some more using high temperature regeneration for a few hours until the outlet temperature and relative humidity is stable for at least 1 hour.

A.2.2.6. Stopping Regeneration

- Open the bypass and close the column valve.
- Set the new temperature set point to 0 (room temperature).
- Set dry gas flowrate to 0.
    - Wet gas flowrate should still be 0.
- Once system pressure reaches 0, close the bypass.
- Allow the system to cool before starting adsorption.
    - The column inlet and outlet temperatures should be equal and at room temperature before starting adsorption.

A.2.2.7. Starting Adsorption

- Allow the system to cool before starting adsorption.
    - The column inlet and outlet temperatures should be equal and at room temperature before starting adsorption.
- Set the total air flowrate and the RH desired to adsorb at.
- Set the mode to RH setting mode (Mode 3)
    - This will open the bypass.
- Allow the RH in the mixing chamber to steady on desired RH.
- Change the mode to adsorption mode once RH is stable (Mode 4).
    - Bypass will close.
    - Immediately open column valve (green valve)
- System is now adsorbing.
- Allow the system to reach equilibrium before regeneration begins again.
    - Equilibrium is achieved once outlet RH value is stable and the difference between the inlet and outlet thermocouples reaches approximately the difference before adsorption starts while the system is at thermal equilibrium

A.2.2.8. Ending/Pausing the Experiment

- Set the mode to manual mode (Mode 0).
- Open the bypass and close the column valve (green valve).
- Set the dry and wet flowrates to 0.
- Allow the systems pressure to reach 0.
- Close the bypass.
    - Column valve should still be closed.
- Click the big red stop button to stop data logging/LabView Software if desired.
- Always make sure the heater control box is turned off when not in use

A.2.3. Data Analysis

There are many outputs from LabView. The outputs that are of most importance is the inlet/outlet relative humidity, inlet/outlet temperature, and the dry/wet volumetric flowrates. The energy storage density (units of kWh/m^3) can be calculated knowing the heat released due to adsorption and the volume of adsorbent used. Many correlations which depend on temperature and pressure are applied to determine the ESD. Since there is a temperature change and humidity change across the adsorbent bed the average temperature and relative humidity across the bed is used. To determine the heat released due to adsorption equation (A.1) is applied.

$$Q_{adsorption} = \int_0^t \dot{m}\, C_p (T_{out} - T_{in}) dt \qquad (A.1)$$

Equation (A.1) depicts the total amount of heat released to the air resulting from the temperature change of adsorption. It is important to note that the resulting temperature change due to adsorption should be corrected by subtracting the average temperature difference across the column before adsorption has started for approximately 10 data points in excel (this corrects the data for thermocouple error, this value is also useful for determining when to end adsorption runs). This ensures that differences/inaccuracies in temperature probes are accounted for. Once the heat released due to adsorption is determined, the ESD can be calculated as shown below in equation (A.2).

$$ESD = \frac{Q_{adsorption}}{V} \qquad (A.2)$$

The first step in calculating the energy storage density is to determine the mass flow of moist air entering the column. The mass flow can be calculated by applying equation A.3 as shown below.

$$\dot{m} = \dot{V} \cdot \rho_{MA} \cdot \frac{1\, m^3}{1000L} \cdot \frac{1\, min}{60s} \qquad (A.3)$$

Where $\dot{V}$ is the total volumetric flowrate in LPM, and ρ_{MA} is the density of moist air in kg/m^3. Equation (A.3) assumes that the volumetric flow reported by LabView is in units of LPM. If the units are not in LPM the necessary unit conversions should be applied and the

division of 60,000 can be omitted. If the volumetric flowrate reported is in SLPM, this should be corrected to LPM first by applying the following equation.

$$\dot{V}_{LPM} = \dot{V}_{SLPM}\left(\frac{T}{298.15K}\right)\left(\frac{14.696}{P_{psi}}\right) \quad (A.4)$$

To determine the mass flow first the density of moist air must be evaluated. Shown in equation (A.5) is the formula for the density of moist air proposed by the NIST (National Institute of Standards and Technology) CIPM (Committee on Weights and Measures) which is valid for the ranges $600\ hPa \leq P \leq 1100 hPa$ and $15°C \leq T \leq 27°C$ [36].

$$\rho_{MA} = \frac{PMW_{DA}}{ZRT}\left[1 - x_v\left(1 - \frac{MW_W}{MW_{DA}}\right)\right] \quad (A.5)$$

Where P is the pressure in Pa, MW_{DA} is the molecular weight of dry air in g/mol, MW_W is the molecular weight of water in g/mol, T is the temperature in K, R is the ideal gas constant in J/mol/K, Z is the compressibility factor and x_v is the mole fraction of water vapour.

Before the density can be determined the mole fraction of water in the air and the compressibility factor must be determined. First the mole fraction of water in air can be evaluated by applying equation (A.6).

$$x_v = RH\frac{P_{sv}(T)}{P} \quad (A.6)$$

Where $P_{sv}(T)$ is the water saturation vapour pressure at a specific temperature in Pa, P is the total pressure in the system in Pa, and RH is the relative humidity of the system which spans from 0-1 and shows how much water is in the air compared to saturation. The ratio of the saturation vapour pressure to the total pressure shows the mole fraction of water in the air at saturation. However, Since the system is not always operated at saturation this value must be corrected by multiplying by the relative humidity to correct for how much water is actually in the air compared to saturation. Before this equation can be applied the saturation vapour pressure of water must be determined as shown by equation (A.7) [36].

$$P_{sv} = 1\ Pa \cdot \exp\left(AT^2 + BT + C + \frac{D}{T}\right) \quad (A.7)$$

Where T is in K, P_{sv} is in Pa, and A-D are fitting constants as shown in Table A.1 below.

Table A.1: Fitting Constants for Saturation Vapour Pressure Calculations

A	1.2378847x10^{-5} K^{-2}
B	-1.9121316 x10^{-2} K^{-1}
C	33.93711047
D	-6.3431645 x10^{3} K

Next, the compressibility factor can be found through another correlation as shown in equation (A.8) below [36].

$$Z = 1 - \frac{P}{T} \cdot [a_0 + a_1 T + a_2 T^2 + (b_0 + b_1 T)x_v + (c_0 + c_1 T)x_v^2] + \frac{P^2}{T^2} \cdot (d + e x_v^2) \qquad (A.8)$$

Where P is in Pa, and T is in K, and the fitting constants are shown below in Table A.2.

Table A.2: Fitting Constants for Compressibility Factor Calculations

a_0	1.58123 x10^{-6} K Pa^{-1}
a_1	-2.9331 x10^{-8} Pa^{-1}
a_2	1.1043 x10^{-10} K^{-1} Pa^{-1}
b_0	5.707x10^{-6} K Pa^{-1}
b_1	-2.051 x10^{-8} Pa^{-1}
c_0	1.9898 x10^{-4} K Pa^{-1}
c_1	-2.376 x10^{-6} Pa^{-1}
d	1.83 x10^{-11} K^{2} Pa^{-2}
e	-0.765 x10^{-8} K^{2} Pa^{-2}

Finally, the density of moist air can be determined by applying equation (A.5). Next, the heat capacity of moist air must be found by weighting the dry air heat capacity and the heat capacity of water by the humidity as shown in equation (A.9).

$$C_{p,MA} = (1 - H)C_{p,DA} + HC_{p,w} \qquad (A.9)$$

Where the humidity H is in kg water/kg total, $C_{p,DA}$ is in kJ/kg DA/K, and $C_{p,w}$ is in kJ/kg water/K. The humidity can be determined by weighting the mass of water over the total mass of fluid in the system as shown by equation (A.10) below.

$$H = \frac{MW_w x_v}{MW_w x_v + MW_a(1 - x_v)} \qquad (A.10)$$

The heat capacities of dry air and water were each obtained by correlations in Smith et al.'s Introduction to Chemical Engineering Thermodynamics 7th edition as shown below in equation (A.11) where the temperature is in K [20].

$$C_p = R(A + BT + CT^2 + DT^{-2}) \qquad (A.11)$$

The fitting constants for the heat capacity of dry air is presented below in Table A.3 and for water in Table A.4.

Table A.3: Fitting Constants for Dry Air Heat Capacity

A	3.355
B	0.575×10^{-3}
C	…
D	-0.016×10^{5}

Table A.4: Fitting Constants for Heat Capacity of Water Vapour

A	3.470
B	1.450×10^{-3}
C	…
D	0.121×10^{5}

Finally, now that everything has been determined the energy storage density can be evaluated by performing numerical integration of the heat released curve over the course of the adsorption time. To do this Simpsons 1/3rd rule is applied as shown below.

$$\int_0^t Q_{adsorption}\, dt = \frac{1}{3}\frac{t_{i+1} - t_{i-1}}{2}\left(Q(t_{i-1}) + 4Q(t) + Q(t_{i+1})\right) \qquad (A.12)$$

Once the total amount of heat released is determined the ESD can be evaluated by applying equation (A.2) and converting to kWh by dividing by 3600. Alternatively, to integrating the data set one can calculate the total mass that flew through the system at each data point by multiplying the flowrate by the time change between each data point as shown by equation (A.13) below.

$$m = \dot{V} \cdot \rho_{MA} \cdot (t_i - t_{i-1}) \cdot \frac{1\ m^3}{1000L} \cdot \frac{1\ min}{60s} \qquad (A.13)$$

This enables the ability to simply add each datapoint together to determine the total heat released instead of integrating the heat of adsorption curve.

A.2.4. References

[1] NASA, "Carbon Dioxide," NASA, February 2024. [Online]. Available: https://climate.nasa.gov/vital-signs/carbon-dioxide/?intent=121. [Accessed 04 04 2024].

[2] Core Writing Team, H. Lee and J. Romero, "Climate Change 2023: Synthesis Report," IPCC, Geneva, Switzerland, 2023.

[3] V. Masson-Delmotte, P. Zhai, H. Portner, D. Roberts, J. Skea, P. Shukla, A. Pirani, W. Moufouma-Okia, C. Peann, R. Pidcock, S. Connors, J. Matthews, Y. Chen, X. Zhou, M. Gomis, E. Lonnoy, T. Maycock, M. Tignor and T. Waterfields, "IPCC, 2018: Summary for Policymakers," Cambridge University Press, New York, 2018.

[4] Natural Resources Canada, "Natural Resources Canada," Government of Canada, 2020. [Online]. Available: https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/showTable.cfm?type=CP&sector=res&juris=ca&rn=7&year=2020&page=3. [Accessed 02 04 2024].

[5] H. Yang, C. Wang, L. Tong, S. Yin, L. Wang and Y. Ding, "Salt Hydrate Adsorption Material-Based Thermochemical Energy Storage for Space Heating Application: A Review," *Energies,* vol. 16, no. 6, p. 2875, 2023.

[6] H. Yang, C. Wang, Y. Zhang, B. Nie, L. Tong, S. Yin, L. Wang and Y. Ding, "Experimental Investigation of thermochemical energy storage system based on MgSO4-silica gel for building heating: adsorption/desorption performance testing and system optimization," *Energy Conversion and Management,* vol. 301, p. 118000, 2024.

[7] C. Strong, Y. Carrier and F. H. Tezel, "Experimental optimization of operating conditions for an open bulk-scale silica gel/water vapour adsorption energy storage system," *Applied Energy,* vol. 312, p. 118533, 2022.

[8] A. Banaei and A. Zanj, "A Review on the Challenges of Using Zeoloite 13X as Heat Storage Systems for the Residential Sector," *energies,* vol. 14, no. 23, p. 8062, 2021.

[9] M. Green, "Community power," *Nature Energy,* no. 1, 2016.

[10] D. Lefebvre, P. Amyot, B. Ugur and F. H. Tezel, "Adsorption Prediction and Modeling of Thermal Energy Storage Systems: A Parametric Study," *Industrial & Engineering Chemistry Research,* vol. 55, no. 16, pp. 4760-4772, 2016.

[11] D. O. Cooney, "Numerical Investigation of Adiabatic Fixed-Bed Adsorption," *Industrial & Engineering Chemistry Process Design and Development,* vol. 13, no. 4, pp. 368-373, 1974.

[12] D. M. Ruthven, Principles of Adsorption and Adsorption Processes, John Wiley & Sons, Inc, 1984.

[13] N. Wakao and T. Funazkri, "Effect of fluid dispersion coefficients on particle-to-fluid mass transfer coefficients in packed beds," *Chemical Engineering Science,* vol. 33, pp. 1375-1384, 1978.

[14] J. O. Hirschfelder, C. F. Curtiss and R. B. Bird, Molecular Theory of Gases and Liquids, John Wiley & Sons, Inc., 1964.

[15] O. Singh and A. W. Joshi, "Effective potential for water vapour," *Pramana,* vol. 15, no. 5, pp. 407-412, 1980.

[16] A. Picard, R. S. Davis, M. Glaser and K. Fujii, "Revised formula for the density of moist air (CIPM-2007)," *Metrologia,* vol. 45, pp. 149-155, 2008.

[17] S. A. Kumar, "Estimating the Effect of Moist Air on Natural Convection Heat Transfer in Electronics Cooling," Qpedia, Bangalore, India, 2008.

[18] M. Sultan, I. I. El-Sharkawy, T. Miyazaki, B. B. Saha, S. Koyama, T. Maruyama, S. Maeda and T. Nakamura, "Insights of water vapor sorption onto polymer based sorbents," *Adsorption,* vol. 21, pp. 205-215, 2015.

[19] P. V. Danckwerts, "Continuous Flow Systems. Distribution of Residence Times," *Chemical Engineering Science,* vol. 2, p. 3857, 1953.

[20] J. M. Smith, H. C. Van Ness and M. M. Abbott, Introduction to Chemical Engineering Thermodynamics seventh edition, New York: McGraw Hill, 2005.

[21] E. Scuiller, P. Dutournie, M. Zbair and S. Bennici, "Thermo-physical properties measurements of hygroscopic and reactive material (zeolite 13X) for open adsorptive heat storage operation," *Journal of Thermal Analysis and Calorimetry,* vol. 147, pp. 12409-12416, 2022.

[22] Z. Qi and A. B. Yu, "A new correlation for heat transfer in particle-fluid beds," *International Journal of Heat and Mass Transfer,* vol. 181, p. 121844, 2021.

[23] C. Strong, Y. Carrier and H. F. Tezel, "Experimental optimization of operating conditions for an open bulk-scale silica gel/water vapour adsorption energy storage system," *Applied Energy,* vol. 312, p. 118533, 2022.

[24] Core Writing Team, H. Lee and J. Romero, "Climate Change 2023: Synthesis Report," IPCC, Geneva, Switzerland, 2023.

[25] D. M. Ruthven, Principles of Adsorption and Adsorption Processes, John Wiley & Sons, Inc, 1984.

[26] "HVAC Direct," HVAC Direct, [Online]. Available: https://hvacdirect.com/sizing-air-conditioner-and-heater.html?srsltid=AfmBOoqbRcEQy6Q85KB0A0bGd4OQKqODEGwDRyS7yIUxmVpLayWWD9gC. [Accessed 19 08 2024].

[27] "Cubic Feet Per Minute (CFM)," American Standard Heating & Air Conditioning, [Online]. Available: https://www.americanstandardair.com/resources/glossary/cfm/. [Accessed 15 09 2024].

[28] "Humidifier Calculation," Vacker, [Online]. Available: https://tools-dashboard.vackerglobal.com/humidifier-calculation. [Accessed 10 08 2024].

[29] "Building Performance Research Centre," Carleton Universiy, [Online]. Available: https://carleton.ca/bprc/people/cher-house/. [Accessed 19 08 2024].

[30] "100,000 BTU Multi-Speed Single Stage Goodman Gas Furnace," HVAC Direct, [Online]. Available: https://hvacdirect.com/goodman-100-000-btu-80-afue-multi-speed-single-stage-gas-furnace-gces801005cn.html. [Accessed 19 08 2024].

[31] "Goodman 2.5 Ton 13.4 SEER2 Air Conditioner Condenser," HVAC Direct, [Online]. Available: https://hvacdirect.com/2-5-ton-13-4-seer2-goodman-air-conditioner-condenser-gsxn3n30106.html. [Accessed 19 08 2024].

[32] "Vacuum Tube Solar Collector Kit," Hydro Solar Innovative Energy, [Online]. Available: https://hydrosolar.ca/products/xianke-xkph58-srcc-certified-evacuated-tube-solar-collector?srsltid=AfmBOooV7CjiU2NDgdDAbBmpLcvu6sHjdfowz2DTkjEWqGeHlIsRwFzI. [Accessed 19 08 2024].

[33] "A.O. Smith 9477 1000 CFM 2 Speed Centrifugal Blower," Amazon, [Online]. Available: https://www.amazon.ca/Smith-9477-Volts-Centrifugal-Blower/dp/B007ATNCPO. [Accessed 19 08 2024].

[34] "Goodman HE12MB Humidifier - CleanComfort," Canada HVAC, [Online]. Available: https://canadahvac.com/product/goodman-he12mb-humidifier-cleancomfort/. [Accessed 19 08 2024].

[35] "Adsorbent Zeolite 13x molecular sieve for CO2 removal 13x molecular sieve beads dehydration and desulphurization," Alibaba, [Online]. Available: https://www.alibaba.com/product-detail/Adsorbent-zeolite-13x-Molecular-Sieve-For_60701038274.html?spm=a2700.7724857.0.0.2b08742a4fdAtE. [Accessed 19 08 2024].

[36] A. Picard, R. S. Davis, M. Glaser and K. Fujii, "Revised formula for the density of moist air (CIPM-2007)," *Metrologia,* vol. 45, pp. 149-155, 2008.

B. Appendix

Griffin Worboy

University of Ottawa

Department of Biological and Chemical Engineering

2024/08/20

B.1.1. adsorptionModel.f90

```
Module adsorptionModel

    USE omp_lib

    IMPLICIT NONE


    PRIVATE

    PUBLIC :: initializeModel, solveTimeStep, cleanUpSimulation


    !Constants

    REAL, PARAMETER :: R = 8.3145 !J/mol*K or Pa*m**3/mol*K

    DOUBLE PRECISION, PARAMETER :: PI = 4.D0*DATAN(1.d0)


    !Solution Parameters

    REAL(8) :: END_TIME

    INTEGER :: NUM_SPACIAL_INTERVALS

    INTEGER :: NUM_DERIVATIVES = 4

    REAL(8) :: INIT_TIME_STEP = 5.0

    DOUBLE PRECISION :: MAX_TE = 0.001d0

    INTEGER :: MAX_ITTERATION = 100000000

    CHARACTER(10) :: simulator = "RK45"


    !Column Characteristics

    REAL(8) :: COLUMN_DIAMETER!m, 0.0335
```

```
REAL(8) :: COLUMN_LENGTH!m,0.0713
REAL(8) :: dz
REAL :: VOID_FRACTION!m^3 open spaces/m^3 column

!Operating Conditions/Initial Conditions
REAL :: VOLUMETRIC_FLOWRATE!LPM
REAL :: PRESSURE!Pa
REAL :: T_FLUID_INLET!K
REAL :: T_FLUID_INIT!K
REAL :: INLET_RH!K
REAL(8) :: VELOCITY
REAL(8) :: INLET_CONCENTRATION

!Fluid Characteristics
REAL :: AIR_MW = 28.96546*10.**(-3.) !kg/mol air
REAL :: WATER_MW = 18.01528*10.**(-3.) !kg/mol water
REAL :: COLLISION_DIAMETER = 0.5*(3.617+2.725)

!Solid Characteristics
REAL :: PARTICLE_DIAMETER!m
!REAL :: qm!mol/kg
REAL(8) :: b0!1/Pa
REAL :: as!1/m
REAL :: SOLID_HEAT_CAP!J/Kg*K
```

```
        REAL :: DENSITY_SOLID!Kg/m3
        REAL :: HEAT_OF_ADS!J/mol
        DOUBLE PRECISION :: Mm0 = 12.025908651
        DOUBLE PRECISION :: qm = 2852.562025
        DOUBLE PRECISION :: C0 = 0.227163506
        DOUBLE PRECISION :: dHc = 9294.968815
        DOUBLE PRECISION :: K0 = 1.604923623
        DOUBLE PRECISION :: dHk = -1537.830778

        DOUBLE PRECISION :: slope = 331.302d0!for overall mass transfer coefficient fitted from gPROMS
        DOUBLE PRECISION :: ks = 3.13904E-06!for overall mass transfer coefficient fitted from gPROMS

        !Initialization of important arrays and parameters
        !DOUBLE PRECISION, DIMENSION(:),allocatable :: concentration, Tf, Ts, q,z
        DOUBLE PRECISION, DIMENSION(:,:),allocatable :: state
        INTEGER :: zStep
        REAL(8) :: startTime, endTime

        CONTAINS
        !-----------------------------------------Accessors, Mutators, Init, and Solve-----------------------------------!
```

```
        subroutine
initializeModel(numberSpacialIntervals,systemPressure,inletFluidTemperature,initialReact
orTemp,
inletRelativeHumidity,voidFraction,columnDiameter,columnLength,flowrate,particleDiam
eter,maxAdsCap,b,specificSA,solidHeatCap,densitySolid,heatAdsorption,inletHR)
            integer, intent(in) :: numberSpacialIntervals
            real, intent(in) :: inletHR,inletRelativeHumidity, inletFluidTemperature,
systemPressure, initialReactorTemp, voidFraction,
flowrate,particleDiameter,maxAdsCap,specificSA,solidHeatCap,densitySolid,heatAdsorpti
on
            real(8), intent(in) :: columnDiameter, columnLength, b
            real :: inletConc

            NUM_SPACIAL_INTERVALS = numberSpacialIntervals

            PRESSURE = systemPressure*101325
            T_FLUID_INIT = initialReactorTemp+273.15
            T_FLUID_INLET = inletFluidTemperature+273.15
            INLET_RH = inletRelativeHumidity/100.
            if(flowrate==0) then
               inletConc=0
            else
               inletConc = (inletHR * flowrate /WATER_MW)/((((1.-
inletHR)*flowrate)/AIR_MW)+((inletHR)*flowrate)/WATER_MW)

            end if
```

```
        INLET_CONCENTRATION  = inletConc
        VOLUMETRIC_FLOWRATE = (flowrate /
densityMoistAirSingle(DBLE(T_FLUID_INLET), DBLE(inletConc)))*1000./60. !L/min

        COLUMN_DIAMETER = columnDiameter
        COLUMN_LENGTH = columnLength
        VOID_FRACTION = voidFraction

        dz = COLUMN_LENGTH / (NUM_SPACIAL_INTERVALS-1)
        VELOCITY = VOLUMETRIC_FLOWRATE * 4. /(PI *
(COLUMN_DIAMETER**2.) * 60000.)
        !INLET_CONCENTRATION =
INLET_RH*waterSatVapPress(DBLE(T_FLUID_INLET))/PRESSURE

        PARTICLE_DIAMETER = particleDiameter
        !qm = maxAdsCap
        b0 = b
        as = specificSA
        SOLID_HEAT_CAP = solidHeatCap
        DENSITY_SOLID = densitySolid
        HEAT_OF_ADS = heatAdsorption

!ALLOCATE(concentration(NUM_SPACIAL_INTERVALS),Tf(NUM_SPACIAL_INTE
RVALS),Ts(NUM_SPACIAL_INTERVALS),q(NUM_SPACIAL_INTERVALS),z(NUM
_SPACIAL_INTERVALS))
```

```
        ALLOCATE(state(NUM_DERIVATIVES,NUM_SPACIAL_INTERVALS))
        !Setting Initial Condiitons
        state(1,:) = T_FLUID_INIT
        state(2,:) = 0.d0
        state(3,:) = 0.d0
        state(4,:) = T_FLUID_INIT

        !CALL
init(simulator,4,NUM_SPACIAL_INTERVALS,INIT_TIME_STEP,state)

    end subroutine
    subroutine cleanUpSimulation()
        DEALLOCATE(state)
        WRITE(*,*) 'Deallocated adsorptionModel, now ddeallocating runge kutta'
        !call cleanUp()
    end subroutine
    function
solveTimeStep(currentTime,endTime,inletRH,inletHR,inletTemp,inletPress,inletFlowrate)
        double precision, intent(in) :: currentTime, endTime
        real, intent(in) :: inletRH,inletHR,inletTemp,inletPress,inletFlowrate
        double precision,dimension(5) :: solveTimeStep
        double precision,dimension(2) :: outlet
        double precision :: conc, temp_out,inletConc,RH_out,outletFlowrate,HR_out
        T_FLUID_INLET = inletTemp+273.15
        INLET_RH = inletRH/100.
```

```
        PRESSURE = inletPress*101325.

        inletConc = (inletHR * inletFlowrate /WATER_MW)/((((1.-
inletHR)*inletFlowrate)/AIR_MW)+((inletHR)*inletFlowrate)/WATER_MW)

        VOLUMETRIC_FLOWRATE = (inletFlowrate /
densityMoistAirSingle(DBLE(T_FLUID_INLET), inletConc))*1000./60.

        VELOCITY = VOLUMETRIC_FLOWRATE * 4. /(PI *
(COLUMN_DIAMETER**2.) * 60000.)

        INLET_CONCENTRATION =
INLET_RH*waterSatVapPress(DBLE(T_FLUID_INLET))/PRESSURE

        END_TIME = endTime


        outlet =
EulerTimeStep(currentTime,endTime,inletRH,inletTemp,VOLUMETRIC_FLOWRATE)!s
olve(derivativeArrays,currentTime,END_TIME,MAX_TE,MAX_ITTERATION,validate)

        RH_out = (PRESSURE*outlet(1)/waterSatVapPress(outlet(2)))*100.

        outletFlowrate = (VELOCITY * PI * (COLUMN_DIAMETER/2.)**2.) *
densityMoistAirSingle(outlet(2), outlet(1))*60.*60.

        temp_out = outlet(2)-273.15

        HR_out = outlet(1) * WATER_MW/fluidMolecularWeightSingle(outlet(1))

        solveTimeStep =
[RH_out,HR_out,temp_out,DBLE(PRESSURE)/101325.,outletFlowrate]

        !WRITE(*,*) 'Simulation took ', (endTime - startTime), 's to run'


    end function solveTimeStep

    function
EulerTimeStep(currentTime,endTime,inletRH,inletTemp,inletVolumetricFlowrate)
```

```
        implicit none
        double precision, intent(in) :: currentTime, endTime
        real, intent(in) :: inletRH,inletTemp,inletVolumetricFlowrate
        double precision,dimension(2) :: EulerTimeStep
        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
nextTemp,nextConc,nextAds,nextTs,dTsdt,dqdt,savedTs,savedAds
        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
mu_MA,rho_MA,Mf,Cp_MA,k_g,Re,Dm,Sc,km,DL,h,q_eq
        double precision :: time,dt,cTime

        time = currentTime
        dt=0.01d0

        nextTemp = state(1,:)
        nextConc = state(2,:)
        nextAds = state(3,:)
        nextTs = state(4,:)

        mu_MA = gasViscocity(nextTemp,nextConc)
        rho_MA = densityMoistAir(nextTemp,nextConc)
        Mf = (1-nextConc)*AIR_MW+nextConc*WATER_MW
        Cp_MA = heatCapacityMoistAir(nextTemp,nextConc)
        k_g = fluidThermalConductivity(nextTemp,nextConc)
        Re = rho_MA * VELOCITY * PARTICLE_DIAMETER / mu_MA
        Dm = molecularDiffusivity(nextTemp)
```

```
Sc = mu_MA/(rho_MA * Dm)
km = massTransferCoefficient(Re,Sc,Dm)
DL = (Dm/VOID_FRACTION)*(20.+0.5*Re*Sc)
h = heatTransferCoefficient(nextTemp,nextConc,mu_MA,Re,Cp_MA)
do while(time<endTime)



   !nextAds = nextAds+dt*dqdt


   !savedTs=nextTs
   !savedAds=nextAds
   !cTime= time
   !do while(cTime<time+dt)
   q_eq = equilibriumAdsorptionCapacity(nextConc,nextTemp)
   dqdt = changeAmtAdsWRTTime(km,q_eq,nextAds)
   nextAds = nextAds+(dt)*dqdt
   dTsdt = changeSldTmpWRTTime(nextTemp,nextTs,dqdt,h)
   nextTs = nextTs+(dt)*dTsdt


   nextConc = nextConc +
(dt)*changeGasConcentrationWRTTime(time,nextConc,DL,rho_MA,dqdt,Mf)
```

```
            !cTime = cTime + dt
        !end do
        !dTsdt=(nextTs-savedTs)/dt
        !dqdt=(nextAds-savedAds)/dt
        nextTemp = nextTemp +
dt*changeFluidTempWRTTime(nextTemp,Cp_MA,rho_MA,k_g,Mf,dqdt,dTsdt)
        time = time +dt
    end do

    state(1,:) = nextTemp
    state(2,:) = nextConc
    state(3,:) = nextAds
    state(4,:) = nextTs
    EulerTimeStep =
[nextConc(NUM_SPACIAL_INTERVALS),nextTemp(NUM_SPACIAL_INTERVALS)]

end function
!-------------------------------------------------Governing Equations-----------------------
-------------------------------!
function derivativeArrays(time,currentState,row,col) result(derArr)
    implicit none
    integer, intent(in) :: row, col
    double precision, dimension(row,col), intent(in) :: currentState
    double precision, intent(in) :: time
```

```
        double precision, dimension(NUM_SPACIAL_INTERVALS) :: TFluid,
fluidConcentration,amtAdsorbed,TSolid
        double precision, dimension(NUM_SPACIAL_INTERVALS) ::
dqdt,dTsdt,dcdt,dTfdt,mu_MA,rho_MA,RE,Dm,Sc,Mf,Cp_MA,k_g,DL,km,q_eq,h
        double precision, dimension(row,col) :: derArr
        TFluid = currentState(1,:)
        fluidConcentration = currentState(2,:)
        amtAdsorbed = currentState(3,:)
        TSolid = currentState(4,:)

        !!$omp parallel sections shared(Tfluid,fluidConcentration)
        mu_MA = gasViscocity(Tfluid,fluidConcentration)
        rho_MA = densityMoistAir(TFluid,fluidConcentration)
        Dm = molecularDiffusivity(TFluid)
        Mf = fluidMolecularWeight(fluidConcentration)
        Cp_MA = heatCapacityMoistAir(TFluid,fluidConcentration)
        k_g = fluidThermalConductivity(TFluid,fluidConcentration)
        q_eq = equilibriumAdsorptionCapacity(fluidConcentration,TFluid)
        !!$omp end parallel sections

        Re = reynoldsNumber(rho_MA,mu_MA)
        Sc = schmidtNumber(rho_MA,mu_MA,Dm)

        !!$omp parallel sections
        DL = axialDispersion(Re,Sc,Dm)
```

```
            km = massTransferCoefficient(Re,Sc,Dm)
            h = heatTransferCoefficient(TFluid,fluidConcentration,mu_MA,Re,Cp_MA)
            !!$omp end parallel sections

            !!$omp parallel sections
            dqdt = changeAmtAdsWRTTime(km,q_eq,amtAdsorbed)
            dTsdt = changeSldTmpWRTTime(TFluid,TSolid,dqdt,h)
            dcdt =
changeGasConcentrationWRTTime(time,fluidConcentration,DL,rho_MA,dqdt,Mf)
            dTfdt =
changeFluidTempWRTTime(TFluid,Cp_MA,rho_MA,k_g,Mf,dqdt,dTsdt)
            !!$omp end parallel sections


            derArr(1,:) = dTfdt
            derArr(2,:) = dcdt
            derArr(3,:) = dqdt
            derArr(4,:) = dTsdt


        end function
        function validate(currentState,row,col) result(updatedState)
            integer, intent(in) :: row,col
            double precision,dimension(row,col),intent(inout) :: currentState
            double precision,dimension(row,col) :: updatedState
```

```
            integer :: m
            do m = 1, col
               if(currentState(2,m)>INLET_CONCENTRATION) then
                  currentState(2,m) = INLET_CONCENTRATION
               end if
               if(currentState(2,m)<0.d0) then
                  currentState(2,m) = 0.d0
               end if
            end do

            updatedState=currentState
         end function
         pure function schmidtNumber(rho_MA,mu_MA,Dm)
            implicit none
            double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
rho_MA,mu_MA,Dm
            double precision,dimension(NUM_SPACIAL_INTERVALS) :: schmidtNumber
            schmidtNumber = mu_MA/(rho_MA * Dm)
         end function

         pure function reynoldsNumber(rho_MA,mu_MA)
            implicit none
            double precision,dimension(NUM_SPACIAL_INTERVALS), intent(in) ::
rho_MA,mu_MA
            double precision,dimension(NUM_SPACIAL_INTERVALS) :: reynoldsNumber
```

```
        reynoldsNumber = rho_MA * VELOCITY * PARTICLE_DIAMETER / mu_MA
    end function

    pure function waterSatVapPress(TFluid)
        implicit none
        double precision, intent(in) :: TFluid
        double precision :: waterSatVapPress

        waterSatVapPress= &
        1*exp((1.2378847*10.**(-5.)*TFluid**(2.)+(-1.9121316*10.**(-
2.)*TFluid)+33.93711047+(-6.3431645*10.**(3.))/TFluid))
    end function

    pure function p_H2O(fluidConcentration)
        implicit none
        double precision,dimension(NUM_SPACIAL_INTERVALS), intent(in) ::
fluidConcentration
        double precision,dimension(NUM_SPACIAL_INTERVALS) :: p_H2O

        p_H2O = fluidConcentration * PRESSURE
    end function

    pure function densityMoistAir(TFluid, fluidConcentration)
        implicit none
```

```
        double precision,dimension(NUM_SPACIAL_INTERVALS), intent(in) ::
TFluid, fluidConcentration

        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
compressibilityFactor, densityMoistAir

        compressibilityFactor = 1 - PRESSURE / TFluid * (1.58123 * 10. ** (-6.) + (-
2.9331 * 10. ** (-8.)) * TFluid + (1.1043 *&

            10. ** (-10.))* TFluid ** (2.) + (5.707 * 10. ** (-6.) + (-2.051 * 10. ** (-8.) *
TFluid)) * fluidConcentration + &

            (1.9898 * 10. ** (-4.) + (-2.376 * 10. ** (-6.)) * TFluid) * fluidConcentration
** (2.)) + &

                pressure ** (2.) / (TFluid ** (2.)) * (1.83 * 10. ** (-11.) + (-0.765 * 10.** (-
8) * fluidConcentration ** (2.)))


        densityMoistAir=PRESSURE * AIR_MW / (compressibilityFactor * R * TFluid)
* (1 - fluidConcentration * (1-WATER_MW/AIR_MW))

    end function

    pure function densityMoistAirSingle(TFluid, fluidConcentration)

        implicit none

        double precision, intent(in) :: TFluid, fluidConcentration

        double precision :: compressibilityFactor, densityMoistAirSingle

        compressibilityFactor = 1 - PRESSURE / TFluid * (1.58123 * 10. ** (-6.) + (-
2.9331 * 10. ** (-8.)) * TFluid + (1.1043 *&

            10. ** (-10.))* TFluid ** (2.) + (5.707 * 10. ** (-6.) + (-2.051 * 10. ** (-8.) *
TFluid)) * fluidConcentration + &

            (1.9898 * 10. ** (-4.) + (-2.376 * 10. ** (-6.)) * TFluid) * fluidConcentration
** (2.)) + &
```

```
                pressure ** (2.) / (TFluid ** (2.)) * (1.83 * 10. ** (-11.) + (-0.765 * 10.** (-
8) * fluidConcentration ** (2.)))


            densityMoistAirSingle=PRESSURE * AIR_MW / (compressibilityFactor * R *
TFluid) * (1 - fluidConcentration * (1-WATER_MW/AIR_MW))
        end function
        pure function heatCapacityMoistAir(TFluid, fluidConcentration)
            implicit none
            double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
fluidConcentration, TFluid
            double precision,dimension(NUM_SPACIAL_INTERVALS) :: humidity, Cp_a,
Cp_w, heatCapacityMoistAir
            humidity = AIR_MW * fluidConcentration / (WATER_MW * fluidConcentration
+ AIR_MW * (1 - fluidConcentration))
            Cp_a = R * (3.55 + 0.575E-3 * TFluid - 0.016E5 * TFluid ** (-2))
            Cp_w = R * (3.470 + 1.450E-3 * TFluid + 0.121E5 * TFluid ** (-2))
            heatCapacityMoistAir = humidity * Cp_w + (1 - humidity) * Cp_a
        end function


        pure function molecularDiffusivity(TFluid)
            implicit none
            double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) :: TFluid
            double precision,dimension(NUM_SPACIAL_INTERVALS) :: Tstar,
collisionIntegral, molecularDiffusivity
```

```
        TStar = TFluid / ((97 * 356) ** 0.5);  ! need leonard jones parameter of water, 2x air used here
                collisionIntegral = 1.06036 / (TStar ** 0.15610) + 0.193 / exp(TStar * 0.4763) + 1.03587 / &
                        (exp(TStar * 1.52996)) + 2 * 1.76474 / (exp(TStar * 3.89411))  ! adsorption assignment 5 solutions
        molecularDiffusivity = (0.001858 * (TFluid ** (3. / 2.)) * &
                        (((1 / (AIR_MW * 10. ** 3.)) + (1 / (WATER_MW * 10. ** 3.))) ** (0.5)) / &
                                ((PRESSURE / 101325.) * (COLLISION_DIAMETER ** (2.)) * collisionIntegral)) / 10000.
        end function


        pure function gasViscocity(TFluid, fluidConcentration)
          implicit none
          double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) :: TFluid,fluidConcentration
          double precision, dimension(NUM_SPACIAL_INTERVALS) :: mu_w,mu_a,phi_ww,phi_wa,phi_aa,phi_aw,gasViscocity


          mu_w=1.12*10**(-5.)*(TFluid/350.)**(1.15)
          mu_a=1.716*10**(-5.)*(TFluid/273.)**(0.666)
          phi_ww=(1./(8.**(1./2.)))*((1+WATER_MW/WATER_MW)**(-0.5))*(1+((mu_w/mu_w)**(0.5))*((WATER_MW/WATER_MW)**(0.25)))**(2.)
          phi_wa=(1./(8.**(1./2.)))*((1+WATER_MW/AIR_MW)**(-0.5))*(1+((mu_w/mu_a)**(0.5))*((AIR_MW/WATER_MW)**(0.25)))**(2.)
```

```
        phi_aa=(1./(8.**(1./2.)))*((1+AIR_MW/AIR_MW)**(-
0.5))*(1+((mu_a/mu_a)**(0.5))*((AIR_MW/AIR_MW)**(0.25)))**(2.)
        phi_aw=(1./(8.**(1./2.)))*((1+AIR_MW/WATER_MW)**(-
0.5))*(1+((mu_a/mu_w)**(0.5))*((WATER_MW/AIR_MW)**(0.25)))**(2.)
        gasViscocity = fluidConcentration*mu_w/(fluidConcentration*phi_ww+(1-
fluidConcentration)*phi_wa)+(1-fluidConcentration)*&
                mu_a/(fluidConcentration*phi_aw+(1-fluidConcentration)*phi_aa)
    end function


    pure function fluidThermalConductivity(TFluid, fluidConcentration)
        implicit none
        double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) :: TFluid,
fluidConcentration
        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
k_w,k_a,mu_w,mu_a,phi_ww,phi_wa,phi_aa,phi_aw,fluidThermalConductivity
        k_w = 0.0181 * (TFluid / 300.) ** (1.35)
        k_a = 0.0241 * (TFluid / 273.) ** (0.81)
        mu_w = 1.12 * 10 ** (-5.) * (TFluid / 350.) ** (1.15)
        mu_a = 1.716 * 10 ** (-5.) * (TFluid / 273.) ** (0.666)
        phi_ww = (1. / (8. ** (1. / 2.))) * ((1 + WATER_MW / WATER_MW) ** (-0.5))
* &
            (1 + ((mu_w / mu_w) ** (0.5)) * ((WATER_MW / WATER_MW) **
(0.25))) ** (2.)
        phi_wa = (1. / (8. ** (1. / 2.))) * ((1 + WATER_MW / AIR_MW) ** (-0.5)) * &
            (1 + ((mu_w / mu_a) ** (0.5)) * ((AIR_MW / WATER_MW) ** (0.25))) **
(2.)
```

```
        phi_aa = (1. / (8. ** (1. / 2.))) * ((1 + AIR_MW / AIR_MW) ** (-0.5)) * &
            (1 + ((mu_a / mu_a) ** (0.5)) * ((AIR_MW / AIR_MW) ** (0.25))) ** (2.)
        phi_aw = (1. / (8. ** (1. / 2.))) * ((1 + AIR_MW / WATER_MW) ** (-0.5)) * &
            (1 + ((mu_a / mu_w) ** (0.5)) * ((WATER_MW / AIR_MW) ** (0.25))) **
(2.)
        fluidThermalConductivity = fluidConcentration * k_w / (fluidConcentration *
phi_ww + (1 - fluidConcentration) * phi_wa) + &
            (1 - fluidConcentration) * k_a / (fluidConcentration * phi_aw + (1 -
fluidConcentration) * phi_aa)
    end function

    pure function fluidMolecularWeight(fluidConcentration)
        implicit none
        double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
fluidConcentration
        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
fluidMolecularWeight
        fluidMolecularWeight = (1-
fluidConcentration)*AIR_MW+fluidConcentration*WATER_MW
    end function
    pure function fluidMolecularWeightSingle(fluidConcentration)
        implicit none
        double precision,intent(in) :: fluidConcentration
        double precision :: fluidMolecularWeightSingle
        fluidMolecularWeightSingle = (1-
fluidConcentration)*AIR_MW+fluidConcentration*WATER_MW
```

```
        end function
        pure function equilibriumAdsorptionCapacity(fluidConcentration,TFluid)
            implicit none
            double precision,dimension(NUM_SPACIAL_INTERVALS), intent(in) ::
fluidConcentration, TFluid
            double precision,dimension(NUM_SPACIAL_INTERVALS) ::
equilibriumAdsorptionCapacity
            double precision,dimension(NUM_SPACIAL_INTERVALS) :: Mm,C,K,RH
            Mm=Mm0*exp(qm/(R*TFluid))
            C=C0*exp(dHc/(R*TFluid))
            K=K0*exp(dHk/(R*TFluid))
            RH = p_H2O(fluidConcentration)/PRESSURE
            equilibriumAdsorptionCapacity = Mm*C*K*RH/((1-K*RH)*(1-
K*RH+C*K*RH))
        end function

        pure function axialDispersion(Re,Sc,Dm)
            implicit none
            double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
Re,Sc,Dm
            double precision,dimension(NUM_SPACIAL_INTERVALS) :: axialDispersion
            axialDispersion = (Dm/VOID_FRACTION)*(20.+0.5*Re*Sc)
        end function

        pure function massTransferCoefficient(Re,Sc,Dm)
```

```
        implicit none
        double precision,dimension(NUM_SPACIAL_INTERVALS), intent(in) ::
Re,Sc,Dm
        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
massTransferCoefficient, Sh
        Sh = 2.0+1.1*Sc ** (1. / 3.) * Re ** 0.6
        massTransferCoefficient =
(slope/(Sh*Dm/(PARTICLE_DIAMETER))+1/ks)**(-1)
    end function

    pure function heatTransferCoefficient(TFluid,fluidConcentration,muG,Re,CpMA)
        implicit none
        double precision,dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
TFluid,fluidConcentration,muG,Re,CpMA
        double precision,dimension(NUM_SPACIAL_INTERVALS) ::
k_g,Pr,Nu_0,beta,Nu_p,heatTransferCoefficient
        k_g=fluidThermalConductivity(TFluid,fluidConcentration)
        Pr = muG * CpMA / k_g
        Nu_0 = 2 + 0.478 * (Re ** 0.48) * (Pr ** 0.45) + 0.012 * (Re ** 1.1) * (Pr **
0.78)
        beta = (3.92 * VOID_FRACTION ** 2 + 0.34 * VOID_FRACTION + 10.3) - 12
*exp(0.38 * (VOID_FRACTION ** 2) * log10(Pr)) &
            + (3.13 * VOID_FRACTION ** 3 - 5.43 * VOID_FRACTION ** 2) *
log10(Re)
        Nu_p = Nu_0 * ((Re * Pr) ** (1 - VOID_FRACTION)) * VOID_FRACTION **
(-beta)
```

```
        heatTransferCoefficient = Nu_p * k_g /PARTICLE_DIAMETER
    end function

    function changeAmtAdsWRTTime(km,q_eq,amtAdsorbed)
        implicit none
        double precision, dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
amtAdsorbed,km,q_eq
        double precision, dimension(NUM_SPACIAL_INTERVALS) ::
changeAmtAdsWRTTime
        !equilibrium adsiorption capacity is returning mmol/g shouldnt it be mmol

        changeAmtAdsWRTTime = (km*as)*(q_eq-amtAdsorbed) !modified due to bad
units with original eq
    end function

    function changeSldTmpWRTTime(TFluid,TSolid,dqdt,h)
        implicit none
        double precision, dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
TFluid,TSolid,dqdt,h
        double precision, dimension(NUM_SPACIAL_INTERVALS) ::
changeSldTmpWrTTime

        changeSldTmpWRTTime =
(h*as/(DENSITY_SOLID*SOLID_HEAT_CAP))*(TFluid-TSolid)+(-
HEAT_OF_ADS/SOLID_HEAT_CAP)*dqdt
    end function
```

```
        function
changeGasConcentrationWRTTime(time,fluidConcentration,DL,rho_g,dqdt,F_MW)
            implicit none
            integer :: i
            double precision, dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
fluidConcentration,dqdt,DL,rho_g,F_MW
            double precision, intent(in) :: time
            double precision :: smoothFunc
            double precision, dimension(NUM_SPACIAL_INTERVALS) ::
changeGasConcentrationWRTTime


            smoothFunc = 1.d0/(1.d0 + exp(-30*time + 4))




            !changeGasConcentrationWRTTime = DL *
matmul(FDMMatrix(:,:,1),fluidConcentration) + (VELOCITY / VOID_FRACTION) * &
                                    !matmul(FDMMatrix(:,:,2),fluidConcentration) - &
                !(DENSITY_SOLID / (rho_g / F_MW)) * ((1 - VOID_FRACTION) /
VOID_FRACTION) * dqdt


            !Boundary Conditions
            !changeGasConcentrationWRTTime(1) = smoothFunc *
cBC1(fluidConcentration(2),DL(2),dqdt(2))
```

```
changeGasConcentrationWRTTime(1)=(DL(1)*(smoothFunc*INLET_CONCENTRATIO
N-2*fluidConcentration(1)&
                +fluidConcentration(1+1))/dz**2-
(VELOCITY/VOID_FRACTION)*(fluidConcentration(1+1)-&
                smoothFunc*INLET_CONCENTRATION)/(2*dz) -
(DENSITY_SOLID/(rho_g(1)/&
                    F_MW(1)))*((1-
VOID_FRACTION)/VOID_FRACTION)*dqdt(1))
        !Main derivative loop - spacial approximated by central finite difference
        do i = 2, NUM_SPACIAL_INTERVALS-1
          changeGasConcentrationWRTTime(i)=(DL(i)*(fluidConcentration(i-1)-
2*fluidConcentration(i)&
                +fluidConcentration(i+1))/dz**2-
(VELOCITY/VOID_FRACTION)*(fluidConcentration(i+1)-&
                fluidConcentration(i-1))/(2*dz) -(DENSITY_SOLID/(rho_g(i)/&
                    F_MW(i)))*((1-
VOID_FRACTION)/VOID_FRACTION)*dqdt(i))
        end do

!changeGasConcentrationWRTTime(NUM_SPACIAL_INTERVALS)=cBC2(changeGasC
oncentrationWRTTime(NUM_SPACIAL_INTERVALS-1))
        i = NUM_SPACIAL_INTERVALS
        changeGasConcentrationWRTTime(i)=(DL(i)*(fluidConcentration(i-1)-
2*fluidConcentration(i)&
                +fluidConcentration(i))/dz**2-
(VELOCITY/VOID_FRACTION)*(fluidConcentration(i)-&
```

```
                fluidConcentration(i-1))/(2*dz) -(DENSITY_SOLID/(rho_g(i)/&
                    F_MW(i)))*((1-
VOID_FRACTION)/VOID_FRACTION)*dqdt(i))

    end function

    function changeFluidTempWRTTime(TFluid,CpMA,rho_g,k_g,Mf,dqdt,dTsdt)
        implicit none
        integer :: i
        double precision, dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
TFluid,CpMA,dqdt,dTsdt
        double precision, dimension(NUM_SPACIAL_INTERVALS),intent(in) ::
rho_g,Mf,k_g
        double precision, dimension(NUM_SPACIAL_INTERVALS) ::
changeFluidTempWRTTime

        !changeFluidTempWRTTime = (k_g/(CpMA/Mf)/rho_g/VOID_FRACTION) *
matmul(FDMMatrix(:,:,1),TFluid) - (VELOCITY/VOID_FRACTION) *&
                    !matmul(FDMMatrix(:,:,2),TFluid) -
(DENSITY_SOLID/(rho_g/Mf))*(SOLID_HEAT_CAP/(CpMA))*&
```

```
                    !((1-VOID_FRACTION)/VOID_FRACTION) * dTsdt -
(DENSITY_SOLID/(rho_g/Mf))*&
                    !(HEAT_OF_ADS/CpMA)*((1-
VOID_FRACTION)/VOID_FRACTION) *dqdt
        !Boundary Conditions
        !changeFluidTempWRTTime(1) =
tBC1(TFluid(2),fluidConcentration(2),rho_g(2),k_g(2),CpMA(2))

changeFluidTempWRTTime(1)=((k_g(1)/(CpMA(1)/Mf(1))/rho_g(1)/VOID_FRACTION)
*(T_FLUID_INLET-2*TFluid(1)+TFluid(1+1))/&
                dz**2 -(VELOCITY/VOID_FRACTION)*(TFluid(1+1)-
T_FLUID_INLET)/(2*dz) &
                -
(DENSITY_SOLID/(rho_g(1)/Mf(1)))*(SOLID_HEAT_CAP/(CpMA(1)))*((1-
VOID_FRACTION)/VOID_FRACTION) * dTsdt(1) &
                -
(DENSITY_SOLID/(rho_g(1)/Mf(1)))*(HEAT_OF_ADS/CpMA(1))*((1-
VOID_FRACTION)/VOID_FRACTION) *dqdt(1))
        !Main derivative loop - spacial approximated by forward upwind scheme
        do i =2, NUM_SPACIAL_INTERVALS - 1

changeFluidTempWRTTime(i)=((k_g(i)/(CpMA(i)/Mf(i))/rho_g(i)/VOID_FRACTION)*(
TFluid(i-1)-2*TFluid(i)+TFluid(i+1))/&
                dz**2 -(VELOCITY/VOID_FRACTION)*(TFluid(i+1)-TFluid(i-
1))/(2*dz) &
                -
(DENSITY_SOLID/(rho_g(i)/Mf(i)))*(SOLID_HEAT_CAP/(CpMA(i)))*((1-
VOID_FRACTION)/VOID_FRACTION) * dTsdt(i) &
```

```
                -(DENSITY_SOLID/(rho_g(i)/Mf(i)))*(HEAT_OF_ADS/CpMA(i))*((1-
VOID_FRACTION)/VOID_FRACTION) *dqdt(i))


            end do
            !changeFluidTempWRTTime(1) = changeFluidTempWRTTime(2)

!changeFluidTempWRTTime(NUM_SPACIAL_INTERVALS)=tBC2(changeFluidTemp
WRTTime(NUM_SPACIAL_INTERVALS-1))
            i = NUM_SPACIAL_INTERVALS

changeFluidTempWRTTime(i)=((k_g(i)/(CpMA(i)/Mf(i))/rho_g(i)/VOID_FRACTION)*(
TFluid(i-1)-2*TFluid(i)+TFluid(i))/&
                dz**2 -(VELOCITY/VOID_FRACTION)*(TFluid(i)-TFluid(i-1))/(2*dz)
&
                -
(DENSITY_SOLID/(rho_g(i)/Mf(i)))*(SOLID_HEAT_CAP/(CpMA(i)))*((1-
VOID_FRACTION)/VOID_FRACTION) * dTsdt(i) &
                -(DENSITY_SOLID/(rho_g(i)/Mf(i)))*(HEAT_OF_ADS/CpMA(i))*((1-
VOID_FRACTION)/VOID_FRACTION) *dqdt(i))
        end function
        FUNCTION max(arr)
            double precision, dimension(:),intent(in) :: arr
            double precision :: max
            integer :: i,j
            max = 0.0001
               do i = 1, NUM_SPACIAL_INTERVALS
```

```
            if(arr(i)>max) then
               max = arr(i)
            end if

        end do
    END FUNCTION
END Module adsorptionModel
```

B.1.2. Type299.f90

```
SUBROUTINE Type299

    USE trnsysfunctions

    USE trnsysconstants

    USE adsorptionModel


!DEC$ATTRIBUTES DLLEXPORT :: TYPE299


    IMPLICIT NONE

    Integer :: controlSignal

    Double Precision :: Timestep,Time !Current time and current time step

    real :: Tf_inlet,RH_inlet,inletFlowrate,TSetpoint,inletHR,inletPressure!inlet fluid temperature, concentration

    real :: HR_init,Tf_init,adsCap,voidFrac !initial parameters <-Maybe make volumetric flowrate variable?

    real :: particleDiameter,specificSA, cpSolid,solidDensity,heatAdsorption !Initial parameters

    real(8) :: b,diameterColumn,columnLength

    Double Precision :: RH_out,HR_out,TF_out,Press_out,Flowrate_out

    double precision,dimension(5) :: timeStepSolution

    integer :: numberSpacialIntervals = 75

    !Get the Global Trnsys Simulation Variables

    Time = getSimulationTime()*60.d0*60.d0

    Timestep = getSimulationTimeStep()*60.d0*60.d0
```

```
!Set the Version Number for This Type
If (getIsVersionSigningTime()) Then
   Call setTypeVersion(17)
   Return
End If
!Do All of the “Last Call” Manipulations That May Be Required
If (getIsLastCallofSimulation()) Then
   Call cleanUpSimulation()
   !Deallocating variables, closing files, cleaning up, etc..
   Return
End If

!Perform Any "End of Timestep" Manipulations That May Be Required
If (getIsEndOfTimestep()) Then

   Return
End If

!Do All of the "Very First Call of the Simulation” Manipulations Here
If (getIsFirstCallofSimulation() ) Then
   ! Tell the TRNSYS Engine How This Type Works
   Call setNumberofParameters(12)
   Call setNumberofInputs(7)
```

```
    Call setNumberofDerivatives(0)
    Call setNumberofOutputs(5)
    Call setIterationMode(1)
    Call setNumberStoredVariables(0,0)
    ! Set the Correct Input and Output Variable Types
    Call setInputUnits(1,'PC1') !Percentage, RH
    Call setInputUnits(2,'DM1') !kg water/kg total, Humidity Ratio
    Call setInputUnits(3,'TE1') !Temperature C, Inlet Temp
    Call setInputUnits(4,'PR4') !atm, Pressure
    Call setInputUnits(5,'MF1') !kg/h, Total Flowrate
    Call setInputUnits(6,'TE1') !Temperature C, Setpoint Temp
    Call setInputUnits(7,'CF1') !Control signal

    Call setInputUnits(1,'PC1') !Percentage, RH
    Call setInputUnits(2,'DM1') !kg water/kg total, Humidity Ratio
    Call setInputUnits(3,'TE1') !Temperature C, Outlet Temp
    Call setInputUnits(4,'PR4') !atm, Pressure
    Call setInputUnits(5,'MF1') !kg/h, Total Flowrate

    ! Return control to the kernel
    Return
End If

!Do All of the First Timestep Manipulations Here - There Are No Iterations at
```

```
! the Intial Time
If (getIsStartTime()) Then
    !Gets inlet values
    RH_inlet = getInputValue(1) !%
    inletHR = getInputValue(2) !kg water/kg total
    Tf_inlet = getInputValue(3) !C
    inletPressure = getInputValue(4) !atm
    inletFlowrate = getInputValue(5) !kg/h
    TSetPoint = getInputValue(6) !C
    controlSignal = getInputValue(7) !-

    ! Read in the Values of the Parameters from the Input File
    HR_init = getParameterValue(1)
    Tf_init = getParameterValue(2)
    diameterColumn = getParameterValue(3)
    columnLength = getParameterValue(4)
    voidFrac = getParameterValue(5)
    adsCap = getParameterValue(6)
    b = getParameterValue(7)
    heatAdsorption = getParameterValue(8)
    particleDiameter = getParameterValue(9)
    specificSA = getParameterValue(10)
    cpSolid = getParameterValue(11)
    solidDensity = getParameterValue(12)
```

```
! Check the Parameters for Problems

If ((HR_init < 0.).or.(HR_init>1.)) Call FoundBadParameter(1,'Fatal','The initial humidity ratio must be between 0 and 1.')

If (Tf_init < -273.15) Call FoundBadParameter(2,'Fatal', 'The initial fluid Temperature must be positive.')

If (inletPressure < 0.) Call FoundBadParameter(3,'Fatal', 'The pressure must be positive.')

If (diameterColumn < 0.) Call FoundBadParameter(4,'Fatal', 'The column diameter must be positive.')

If (columnLength < 0.) Call FoundBadParameter(5,'Fatal', 'The column length must be positive.')

If ((voidFrac < 0.).or.(voidFrac>1.)) Call FoundBadParameter(6,'Fatal', 'The void fraction must be between 0 and 1.')

If (adsCap < 0.) Call FoundBadParameter(7,'Fatal', 'The adsorption capacity must be positive.')

If (b < 0.) Call FoundBadParameter(8,'Fatal', 'The b parameter must be positive.')

!If (heatAdsorption < 0.) Call FoundBadParameter(9,'Fatal', 'The heat of adsorption must be positive.')

If (particleDiameter < 0.) Call FoundBadParameter(10,'Fatal', 'The particle diameter must be positive.')

If (specificSA < 0.) Call FoundBadParameter(11,'Fatal', 'The specific surface area must be positive.')

If (cpSolid < 0.) Call FoundBadParameter(12,'Fatal', 'The solid heat capacity must be positive.')

If (solidDensity < 0.) Call FoundBadParameter(13,'Fatal', 'The solid density must be positive.')
```

```
If (ErrorFound()) Return


Call
initializeModel(numberSpacialIntervals,inletPressure,Tf_inlet,Tf_init,RH_inlet,voidFrac,di
ameterColumn,columnLength,inletFlowrate,particleDiameter,adsCap,b,specificSA,cpSolid,
solidDensity,heatAdsorption,HR_init)


! Set the Initial Values of the Outputs

timeStepSolution =
solveTimeStep(time,time+timeStep,RH_inlet,inletHR,TF_inlet,inletPressure,inletFlowrate)
!make time step solution output temp in C, humidity ratio in kg w/kg total, rh (%),flowrate
in kg/h,pressure in atm

RH_out = timeStepSolution(1)

HR_out = timeStepSolution(2)

Tf_out = timeStepSolution(3)

Press_out = timeStepSolution(4)

Flowrate_out = timeStepSolution(5)


!Call setOutputValue(1,0.)

!Call setOutputValue(2,0.)

!Call setOutputValue(3,Tf_inlet)

!Call setOutputValue(4,inletPressure)

!Call setOutputValue(5,inletFlowrate)

Call setOutputValue(1,RH_out)

Call setOutputValue(2,HR_out)
```

```
    Call setOutputValue(3,Tf_out)
    Call setOutputValue(4,Press_out)
    Call setOutputValue(5,Flowrate_out)
    ! Return control to the kernel
    Return
  End If

  !ReRead the Parameters if Another Unit of This Type Has Been Called Last
  If (getIsReReadParameters()) Then
    HR_init = getParameterValue(1)
    Tf_init = getParameterValue(2)
    diameterColumn = getParameterValue(3)
    columnLength = getParameterValue(4)
    voidFrac = getParameterValue(5)
    adsCap = getParameterValue(6)
    b = getParameterValue(7)
    heatAdsorption = getParameterValue(8)
    particleDiameter = getParameterValue(9)
    specificSA = getParameterValue(10)
    cpSolid = getParameterValue(11)
    solidDensity = getParameterValue(12)

    If ((HR_init < 0.).or.(HR_init>1.)) Call FoundBadParameter(1,'Fatal','The initial
concentration must be between 0 and 1.')
```

```
        If (Tf_init < -273.15) Call FoundBadParameter(2,'Fatal', 'The initial fluid
Temperature must be positive.')

        !If (systemPressure < 0.) Call FoundBadParameter(3,'Fatal', 'The pressure must be
positive.')

        If (diameterColumn < 0.) Call FoundBadParameter(4,'Fatal', 'The column
diameter must be positive.')

        If (columnLength < 0.) Call FoundBadParameter(5,'Fatal', 'The column length
must be positive.')

        If ((voidFrac < 0.).or.(voidFrac>1.)) Call FoundBadParameter(6,'Fatal', 'The void
fraction must be between 0 and 1.')

        If (adsCap < 0.) Call FoundBadParameter(7,'Fatal', 'The adsorption capacity must
be positive.')

        If (b < 0.) Call FoundBadParameter(8,'Fatal', 'The b parameter must be positive.')

        !If (heatAdsorption < 0.) Call FoundBadParameter(9,'Fatal', 'The heat of
adsorption must be positive.')

        If (particleDiameter < 0.) Call FoundBadParameter(10,'Fatal', 'The particle
diameter must be positive.')

        If (specificSA < 0.) Call FoundBadParameter(11,'Fatal', 'The specific surface area
must be positive.')

        If (cpSolid < 0.) Call FoundBadParameter(12,'Fatal', 'The solid heat capacity must
be positive.')

        If (solidDensity < 0.) Call FoundBadParameter(13,'Fatal', 'The solid density must
be positive.')

        If (ErrorFound()) Return


    End If

    !Get the Current Inputs to the Model
```

```
RH_inlet = getInputValue(1) !%
   inletHR = getInputValue(2) !kg water/kg total
   Tf_inlet = getInputValue(3) !C
   inletPressure = getInputValue(4) !atm
   inletFlowrate = getInputValue(5) !kg/h
   TSetPoint = getInputValue(6) !C
   controlSignal = NINT(getInputValue(7)+0.1) !-

!Check the Inputs for Validity
If ((RH_inlet > 100.).or.(RH_inlet < 0.)) Call foundBadInput(1,'Fatal', 'The inlet relative humidity must be between 0 and 100.')
If (Tf_inlet < -273.15) Call foundBadInput(2,'Fatal','The inlet temperature cannot be negative')
If (inletFlowrate < 0.) Call foundBadInput(3,'Fatal','The inlet flowrate cannot be negative')
If (TSetpoint < 0.) Call foundBadInput(4,'Fatal','The setpoint temperature cannot be negative')
If ((controlSignal > 1.).or.(controlSignal < 0.)) Call foundBadInput(5,'Fatal','The heater control signal must be between 0 and 1.')
If (ErrorFound() ) Return

!Perform All of the Calculations Here
If (inletFlowrate <= 0.0) Then
   ! No flow
   Call setOutputValue(1,0.)
```

```
      Call setOutputValue(2,0.)
      Call setOutputValue(3,Tf_inlet)
      Call setOutputValue(4,inletPressure)
      Call setOutputValue(5,0.)

    Else
      ! Check inlet temperature and control function
      If ((Tf_inlet < TSetpoint).and.(controlSignal == 1.d0) ) Then !
        ! Heater on

        if((ABS(RH_out-RH_inlet)/RH_inlet<0.005).and.(time .ne.0.)) then
          RH_out = RH_inlet
          HR_out = inletHR
          Tf_out = Tf_inlet
          Press_out = inletPressure
          Flowrate_out = inletFlowrate
        else
          timeStepSolution =
solveTimeStep(time,time+timeStep,RH_inlet,inletHR,TF_inlet,inletPressure,inletFlowrate)
          if(timeStepSolution(1) > 100) then
            RH_out = 100.
          else
            RH_out = timeStepSolution(1)
          end if
```

```
            HR_out = timeStepSolution(2)
            Tf_out = timeStepSolution(3)
            Press_out = timeStepSolution(4)
            Flowrate_out = timeStepSolution(5)
          end if

        Else
          ! Heater off
          Tf_out = Tf_inlet
          HR_out = inletHR
          Press_out = inletPressure
          RH_out = RH_inlet
          Flowrate_out = 0.0
        EndIf
        ! Set outputs
        Call setOutputValue(1,RH_out)
        Call setOutputValue(2,HR_out)
        Call setOutputValue(3,Tf_out)
        Call setOutputValue(4,Press_out)
        Call setOutputValue(5,Flowrate_out)
      EndIf
      Return
END SUBROUTINE Type299
```