Heat and Mass Transfer Characteristics of Desiccant Polymers

by

JoAnna Christen Staton, B.S.

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Dr. Elaine P. Scott, Chair
Dr. Ronald G. Kander
Dr. James R. Thomas

May 1998

Blacksburg, Virginia

Heat and Mass Transfer Characteristics of Desiccant Polymers

Approved by
Supervising Committee:

____

____

____

Heat and Mass Transfer Characteristics of Desiccant Polymers

JoAnna Christen Staton, M.S.

Virginia Polytechnic Institute and State University, 1998

Advisor: Elaine P. Scott

Abstract

Desiccant-enhanced air conditioning equipment has exhibited both the capability to improve humidity control and the potential to save energy costs by lowering the latent energy requirement of the supply air stream. The resulting increasing popularity of desiccant-enhanced air conditioning systems has sparked new interest in the search for a better, more efficient desiccant material. The ultimate goal of this research was to develop a material that, when applied to an existing air-to-air heat exchanger, would achieve the necessary heat *and* mass transfer in a single process, thus transforming a sensible heat exchanger into a *total enthalpy exchanger*.

This study focuses on the development and determination of appropriate polymeric desiccant materials for use in different heat and mass transfer applications. Various candidate materials were initially studied. It was decided that polyvinyl alcohol best met the pre-determined selection criteria. After the focus material was chosen, numerical models representing two heat and mass transfer applications were created. One-dimensional numerical models were developed for the performance studies of a rotary wheel total enthalpy exchanger. A two-dimensional numerical model was developed for the performance studies of a fixed plate total enthalpy exchanger as well. Material characterization tests were performed to collect material property information required by the numerical models.

Sensible, latent, and total efficiencies gathered from both the rotary wheel total enthalpy exchanger and the fixed plate total enthalpy exchanger models indicate potential uses for some candidate polyvinyl alcohol materials.

Acknowledgments

Thank you to Des Champs Laboratories, Inc. and the Center for Innovative Technology for funding this work. Thank you to Ben Tritt for serving as our project manager.

I would like to thank my advisor, Dr. Elaine Scott. I am grateful for the guidance she provided me throughout this project, and for the way in which she made herself available whenever I had a concern. Thank you, Dr. Scott, for your encouragement.

I would also like to thank my committee members, Dr. Ron Kander and Dr. J. R. Thomas for providing helpful suggestions and resources during the course of this research. Special thanks to Dr. Kander for serving as the MSE faculty advisor on this project and for giving me 'crash' courses in polymer science.

I would like to thank Jennifer Howard for providing me with material property and characterization data for my numerical models and for letting me come into the polymer lab to help her run tests. Thanks to Linda Vick for initial contacts and uptake testing.

Thank you to all the Heat Transfer Lab Rats for providing endless hours of distraction and for putting up with my big mouth. I will miss working with all you guys.

Appreciation is extended to Mahesh Balakrishnan and Hélène Cornélis, who both contributed to the project's preliminary literature review.

Finally, I would like to thank my family for all their support. I'm done, Pete! Yipee!

JoAnna Christen Staton

Virginia Polytechnic Institute and State University

May 1998

Contents

List of Tables

List of Figures

Nomenclature

A_c = cross-sectional area

a =thickness

C =specific heat

D_{ab} =diffusion coefficient

D =depth of dehumidifier

f =mass fraction of desiccant in dehumidifier

H =enthalpy

h =heat transfer coefficient

H_v =heat of vaporization

k =conductivity

K_y =mass transfer coefficient

M =mass

P =perimeter

Q =heat of adsorption

T =Temperature

t =time

Y =humidity ratio

$Y_{w_{max}}$ =moisture content of air at 100 percent relative humidity

W =moisture content

W_{max} =maximum moisture loading of desiccant material

V =velocity

x =distance along the length of the desiccant bed

X =sensitivity coefficient

y =distance along desiccant bed thickness

Greek:

α =thermal diffusivity

ρ =density

η =efficiency

σ =uncertainty

Subscripts:

a =moist air

bc =boundary condition

d =desiccant

da =dry air

i =member in a series

ic =initial condition

l =latent

n =position step index in x direction

m =position step index in y direction

$mass$ =mass moisture

max =maximum

pro =process (supply)

reg =regenerative (exhaust)

s =sensible

tot =total

v =vapor

w =desiccant bed

Superscripts

k =time step index

$*$ =normalized with respect to maximum capacity

Chapter 1

Introduction

Controlling temperature and humidity within a conditioned space is important for a wide variety of applications. Since the 1920's desiccant materials have been used for dehumidification purposes including process improvement and product protection. In the past twenty years the use of desiccant dehumidification has expanded into new venues including hospitals, hotels, ice rinks, and supermarkets. Still more recently, the possibilities of applying desiccant technologies to office and residential buildings has been explored.

The wide acceptance of desiccant technology, coupled with the expanding markets for these materials has created a competitive production environment. The air conditioning industry is continually looking for an improved desiccant material. Specifically, research within the field of air conditioning technology has focused upon the development of new desiccant materials capable of removing both sensible and latent energy within a single process. A vast number of desiccant applications currently in operation use desiccant technology in tandem with a heat exchanger; in this situation sensible and latent energy are processed at different stages within the air conditioning cycle. The ultimate goal of this research was to develop a material that, when applied to either an existing heat exchanger or a novel heat exchanger design, would achieve the necessary heat *and* mass transfer in a single process, thus transforming a sensible heat exchanger into a *total enthalpy exchanger*.

Traditionally, the desiccant market has been dominated by ceramic materials. The popularity of these materials can be attributed to their low cost and high availability in addition to their ability to easily take on water, a result of their porous construction. Notwithstanding, ceramics do lack certain desirable characteristics in desiccant applications.

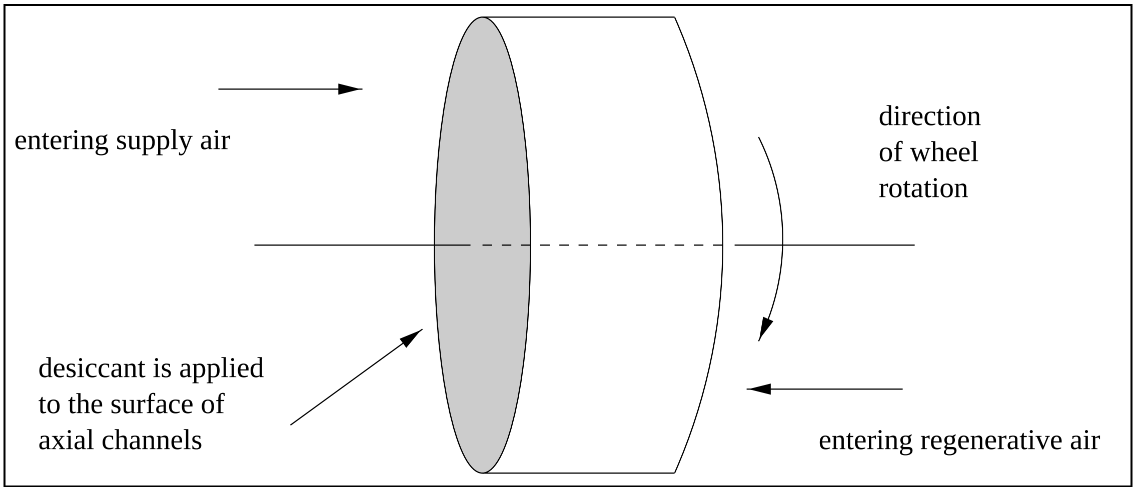

Figure 1.1 Representative Rotary Wheel Total Enthalpy Exchanger Environment

For instance, these materials tend to degrade as they continuously cycle through sorption stages and consequently need to be replaced often. Ceramics also frequently require high temperatures for effective desorption. The primary goal of this project was the development and/or selection of a desiccant material, superior to ceramic materials currently in operation, for use in total energy exchanger applications.

For the past two years, the Mechanical Engineering and the Materials Science Engineering Departments at Virginia Tech have been working cooperatively with Des Champs Laboratories, Inc. to determine appropriate polymeric materials for use in different air-to-air heat and mass transfer applications. Two applications were studied: a rotary wheel total enthalpy exchanger and a fixed plate total enthalpy exchanger. Due to modeling considerations discussed later, it was decided that the rotary wheel total enthalpy exchanger would be the first system modeled followed by a study of the fixed plate total enthalpy exchanger. Representations of both the rotary wheel total enthalpy exchanger and a fixed plate total enthalpy exchanger are provided in Fig. 1.1 and Fig. 1.2 respectively.

1.1 Objectives

Several objectives were formulated to address the overall goal of the research project: the development and validation of an improved desiccant for use in various total enthalpy exchangers. These objectives are to:

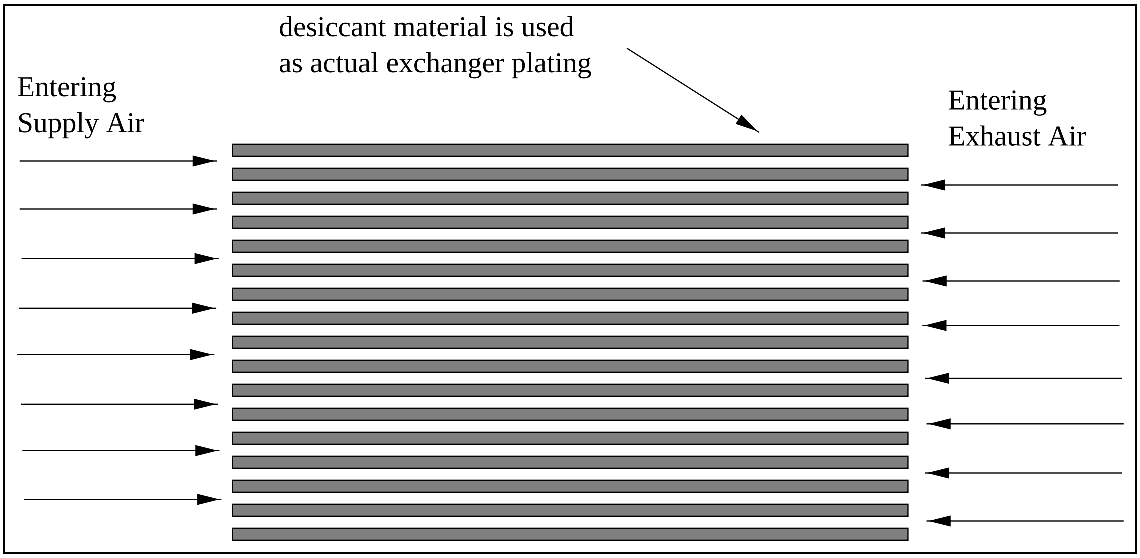

Figure 1.2 Representative Fixed Plate Total Enthalpy Exchanger Environment

1. Select a polymer desiccant suitable to incorporate into the existing air conditioning applications mentioned above.

2. Determine material properties required in the modeling of each application (were evaluated within the Materials Science Engineering Department's labs).

3. Develop the mathematical models used to analyze desiccant material performance.

4. Enhance and augment mathematical models to accommodate alternate materials when needed.

In the early stages of the study, preliminary sorption and desorption tests were performed on several candidate polymer materials in an effort to determine a material of focus to be studied throughout the research. These tests were conducted to qualitatively determine both the maximum moisture capacity, by percent mass uptake, of various desiccants as well as the uptake and desorption time responses. The results of these tests were then used to comparatively evaluate the field of candidate materials. Using information from these results in addition to information gathered regarding material availability, material cost, and material flexibility, the selection of polyvinyl alcohol as the focus polymer desiccant was made, thus marking the completion of Objective 1.

Once polyvinyl alcohol was chosen as the candidate polymer desiccant, it was necessary to determine what form of polyvinyl alcohol would be appropriate to use in each application. This is outlined in the second objective. Polyvinyl alcohol comes in various forms including powders, films, gels, and foams. Due to the ease with which a small-channeled rotary wheel could be coated with a film desiccant, polyvinyl alcohol film was selected as the desiccant coating for the rotary wheel, the first application to be modeled. The analysis presented in Chapter 5 later lead to the investigation of another coating, polyvinyl alcohol open-cell foam, for the rotary wheel total enthalpy exchanger. Results from the foam coated rotary wheel total enthalpy exchanger analysis coupled with polyvinyl alcohol foam's structural characteristics contributed to the decision to use this material in the evaluation of the fixed plate total enthalpy exchanger as well.

Objective 3 was accomplished via a heat and mass transfer analysis of each material-specific application. With the implementation of a control volume technique, difference equations were developed which were in turn used in the derivation of the governing differential equations for heat and mass transfer both within the desiccant and the air stream. These finite difference equations were solved using a quasi-central finite difference scheme. Assuming the conditions outlined in Chapter 4, the rotary wheel applications were treated as one-dimensional problems while the counter-flow plate exchanger was represented as a two-dimensional problem. The temperature and moisture contents were calculated at the inlet and exit air streams of both the supply and exhaust flows and sensible, latent, and total efficiencies were determined for the particular system.

Each numerical model was used to examine exchanger efficiency under various conditions. Boundary and initial conditions, desiccant material thickness, air stream velocity, exchanger depth, and channel spacing are examples of the parameters that were changed throughout the analysis of each total enthalpy exchanger. The effects of such parameter changes on performance efficiencies were observed within each model.

After first developing the one-dimensional programs and performing several modeling analyses, it was concluded that the film would not be an acceptable desiccant material for the rotary wheel application. Due to the non-porous structure of the film, a strictly absorbing material, the surface area of the coating was found to be inadequate in transferring the desired amount of moisture from one air stream to another; the latent efficiency of the polyvinyl alcohol film coated wheel was undesirably low. Modeling of the polyvinyl alcohol foam coated application as well as a polyvinyl alcohol/ceramic composite coated system revealed an increased latent efficiency; these materials were proposed to be better desiccant materials for the rotary wheel total enthalpy exchanger.

The two-dimensional analysis of the fixed plate exchanger took place in the second half of the study. The modeling efforts corresponding to the counter flow plate exchanger program were independent of the rotary wheel total enthalpy exchanger models. Analysis of polyvinyl alcohol foam fixed plate exchangers exhibited acceptable sensible and latent efficiency results. Although structurally rigid, selectively permeable, open-cell polyvinyl alcohol foam was considered for both applications, the transfer phenomena were exchanger specific.

The analysis of the models conducted in this study suggest that two polyvinyl alcohol materials, polyvinyl alcohol open-cell foam and polyvinyl alcohol/silica gel/molecular sieve composite, can be successfully used in total energy exchange applications. With cases resulting in total efficiencies exceeding 65% and latent efficiencies of over 50%, both polyvinyl alcohol foam and polyvinyl alcohol/ceramic composite were shown to perform well as desiccants in rotary wheel total enthalpy exchanger systems. In addition, with a total efficiency of greater than 80%, polyvinyl alcohol foam was shown to be efficient in fixed plate total enthalpy exchanger systems. With appropriate optimization of exchanger design, these two polyvinyl alcohol materials have the potential to perform as well as desiccant materials used in current rotary wheel and fixed plate total energy exchanger applications.

Chapter 2

Literature Review

Since the early 1920's water absorption systems have become increasingly popular and are currently used in several industries. More recently, since the 1980's, desiccant-based air conditioning systems have been used in process and comfort-based applications (Anon., 1996) and have been considered for use in residential applications (Jurinak et al., 1984). Air conditioning systems based on desiccants have the potential to save energy costs by lowering the latent energy requirement of the supply air stream. A reduction in latent energy removal could eliminate the need for environmentally unfriendly refrigerants. Desiccant systems can also improve humidity control.

The main motive behind this study was to develop a water absorption system that replaces the current refrigerant-based air conditioning systems. Such a system would require development of newer, better materials as well as the property evaluation of these new materials. Commercial production, viability and safety issues would also need to be studied for each new material. Lastly, development of models that best capture each desiccant material's steady-state behavior would be required. All of these objectives were met while keeping in mind that development of a desiccant material capable of being directly applied to existing air conditioning equipment was most preferred.

The review that follows focuses on the materials that are currently used as desiccants, existing hydrophilic polymers and their applications, and laboratory based material property evaluation techniques. In addition, various numerical models that best capture the heat and mass transfer of existing systems are explored.

2.1 Total Energy Exchangers in Air Conditioning Applications

Traditionally, especially with the use of ceramic desiccants, proper conditioning of supply air requires a two phase process: mass and heat within the air stream are treated (transferred) separately as exemplified by Anon. (1997). This two phase process could be attributed to the limiting factor of desiccant moisture uptake rate. If a desiccant material has a slow moisture uptake rate, then the desiccant wheel must rotate slowly, at typical speeds of 6-10 rph. Rotational speeds this slow require the implementation of another wheel, a sensible heat exchanger, to help transfer the sensible heat from one air stream to another. Sensible heat exchangers can rotate at rates as fast as 60 rpm.

To illustrate, consider a conditioning cycle using summer-time conditions; the supply flow (warm, humid air) is being conditioned to be cooler and drier while the exhaust flow (a regenerative stream) is receiving the heat and moisture and exhausting it to outdoor air. This two phase conditioning process is represented in Fig. 2.1. As supply air enters the conditioning equipment, it encounters a desiccant dehumidifier which adjusts the moisture content in the air. The removal of moisture increases the supply air temperature, a result of the desiccant's heat of adsorption. The temperature of the supply air is adjusted as it passes through the sensible heat exchanger. Finally, the supply air temperature is refined with the use of the cooling coil. As the exhaust air enters to perform regenerative functions, the air must first pass through a heater in order to elevate the temperature of the regenerative air so that, when passed though the channels of the rotary humidifier, it will drive off the water held by the desiccant wheel.

If a desiccant material with an increased rate of absorption and desorption and a low desorption temperature was incorporated into air conditioning equipment similar to the equipment discussed above, the sensible heat exchanger component could be combined with the desiccant dehumidifier to produce a total enthalpy exchanger. The implementation of a total enthalpy exchanger introduces possibilities for increased energy transfer efficiency, reduced material costs, a decrease in the number of components required to properly condition the air, and a decline in the demand put on refrigerant-based components within the conditioner.

Consider once more a conditioning cycle in summer-time conditions using a total

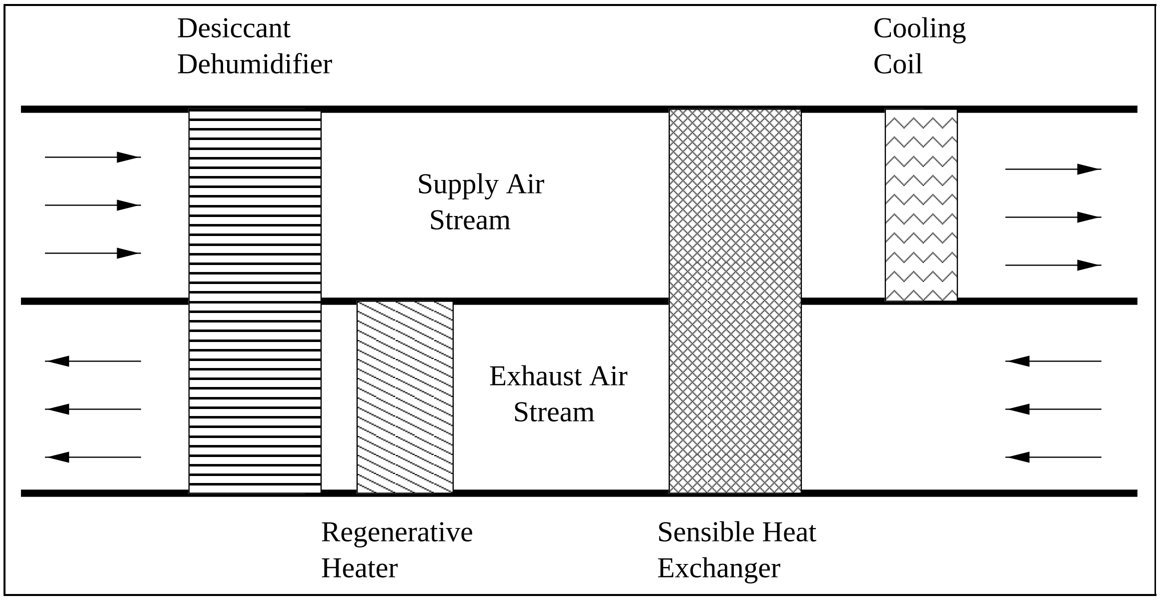

Figure 2.1 Schematic of Two Step Air Conditioning Process

enthalpy exchanger, as illustrated in Fig. 2.2. Supply air entering the conditioning equipment would encounter the total enthalpy exchanger. At this stage, both latent and sensible efficiency would be transferred with one piece of equipment. As the exhaust air enters to perform regenerative functions, the air may not need to pass through a regenerative heater; the desiccant could have a very low desorption temperature. Instead, the exhaust air flows straight to the rotary total enthalpy exchanger, where the exhaust air would drive off the moisture from the desiccant material.

2.2 Overview of Traditional and Potential Desiccants

While surveying the relevant literature for this research, it became clear that a variety of materials had been previously researched for desiccant applications. Polymer materials were among the desiccants studied. However, the literature reviewed revealed little effort and study into the area of polymer desiccants within energy exchanger applications.

2.2.1 Traditional Desiccant Materials

Industrial air conditioning products still employ traditional desiccants that have been used in the market for decades. A list of these materials is given in Table 2.1.

A comprehensive overview of existing open-cycle desiccant technology employing

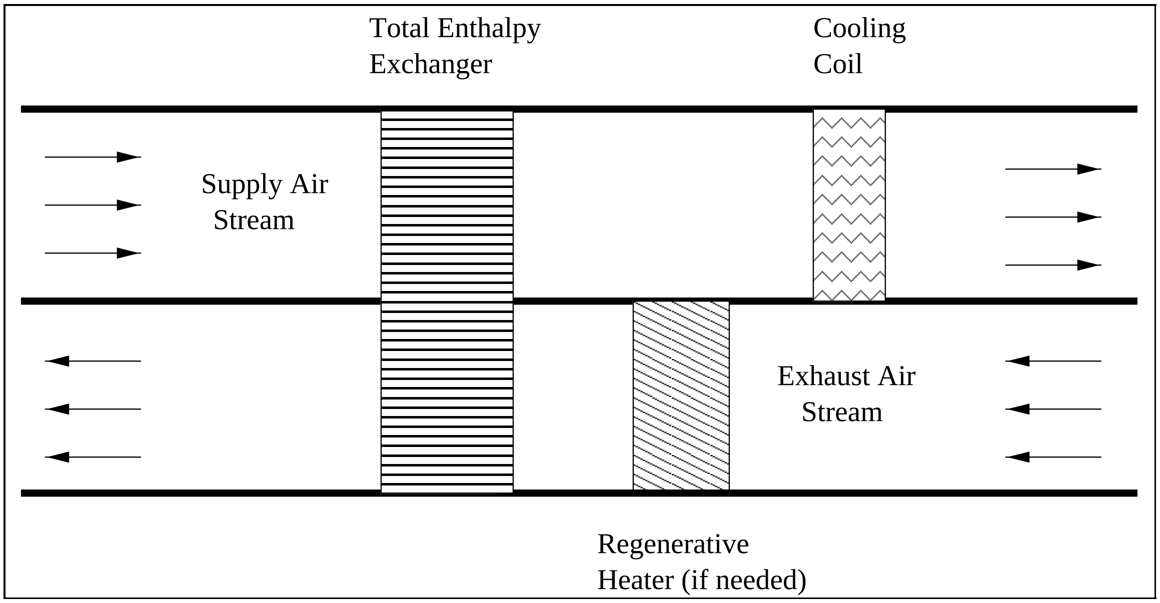

Figure 2.2 Schematic of One Step Air Conditioning Process

some of the materials listed in Table 2.1 is presented by Collier et al. (1982). Solar regenerated desiccant-cooling systems are the main focus of this particular research. Two proposals are introduced for conceivable dehumidifier improvement including the development of new desiccant materials suitable for regeneration at low temperatures. Collier et al. (1982) continue by listing five desirable characteristics for these new materials.

1. Mechanical and chemical stability

2. Large maximum moisture capacity

Table 2.1 Traditional Desiccants and Their Noted Characteristics

Desiccant	Characteristics
Activated Alumina	Aluminum Oxide is highly porous and has a very large surface area. It resists shock and abrasion, does not shrink, swell, distort or disintegrate
Silica Gel	Most popular desiccant material. It can selectively absorb compounds of different molecular weights
Molecular Sieve	Synthetically produced zeolites characterized by pores and cavities of extremely uniform size
Activated Charcoal	Usually coconut shell based. It has an extremely high surface area
Clay	Once dried, can sometime be used as a desiccant

3. High sorption rates at low vapor pressures

4. Low heats of adsorption

5. Ideal Isotherm shape

2.2.2 Polymer Desiccant Materials

Polymers are large molecules consisting of repeating chemical units, or mers. One polymer molecule, or 'macro-molecule', can consist of 50 to 3000 mers. Polymers are capable of being cross-linked, which is the condition of the polymer chains being bonded together to form a network. The bonds responsible for cross-linking are covalent bonds. Once cross-linked, polymers will neither dissolve when put in solution nor flow when heated.

Super-Absorbent Polymers

Using information gathered during this review, a listing of hydrophilic polymers believed to have potential as desiccants was compiled. Super-absorbent polymers (SAP) are hydrogels, water-insoluble hydrophilic homopolymers or copolymers, which are able to swell and absorb 10 to 1000 times their own weight in water (Pó, 1994) and (Kudela, 1984). SAPs are cross-linked in order to avoid dissolution. During his patent survey, Pó (1994) discovered three main classes of SAP's.

1. Cross-linked polyacrylates and polyacrylamides

2. Cellulose- or starch-acrylonitrile graft copolymers

3. Cross-linked maleic anhydride copolymers

Other hydrophilic polymers such as polyethylene oxide (Pó, 1994), polyvinyl alcohol (Kudela, 1984), polyurethane (Lomax, 1990), and poly(N-vinylpyrrolidone) (Pó, 1994) have been used as hydrogels. Pó (1994) and Kudela (1984) recognize that these materials need first to undergo cross-linking or blending to prevent dissolution. Pó (1994) presents the two main methods for cross-linking polymers. Kudela (1984) presents a comprehensive evaluation of the different classes of hydrogels as well as methods of polymerization and processing.

Several authors focused their studies on particular SAPs, or classes of SAPs. For instance, Masuda and Iwata (1990) studied highly absorbent polyacrylate polymers and report excellent reversible absorption- drying cycles under high and low humidity levels, respectively. Likewise, Miyata et al. (1995) focused on the examination of the moisture sorption characteristics of hydroxyethylcellulose (HEC) graft copolymers. This study touched upon structural investigations into the effects of moisture content on visco-elasticity as well as the material's absorption kinetics.

Additionally, Harland (1993) presents the notion of synthetic and bio-degradable ultimate superabsorbents, again characterized by their absorbency. Ultimate superabsorbents are material systems containing an empty cell surrounded by a semi-permeable expandable material. An osmotic agent is placed within the cell which forces the systems to expand under wetting conditions. To exemplify such materials, both poly(acrylic acid)-based systems (synthetic)and polysaccharides (bio-degradable) were used. The moisture absorbency of these ultimate SAP's can exceed 44 times that of the original weight of the polymer.

Absorption capacity, absorption rate, and gel strength usually characterize the properties of SAPs. These properties can be evaluated by different methods; however, there exists an American Society for Testing and Materials (ASTM) standard to evaluate the absorption of water by molded plastics when immersed (ASTM 1981). This test is primarily used as the control test for the verification of a product's uniformity. The degree of water absorption is very dependent on the cross-linking density of the systems (Pó, 1994) and (Kulicke et al., 1989)

The most important property of hydrogels is their high affinity for water (Kudela, 1984). The capability of hydrogels to attract water can be attributed to the presence of groups like hydroxyl groups which are highly water-soluble. Even when saturated, hydrogels can remain insoluble and structurally stable due to the three-dimensional cross-linked network.

Lower Critical Solution Temperature Polymers

Allcock (1996) presents the synthesis and properties of several water-soluble polyphosphazenes and their hydrogels. One of these phosphazene polymers, poly[bis(methoxyethoxyethoxy) phosphazene] (MEEP), has unusual solution properties in water; it displays a lower critical solution temperature (LCST). That is, it is soluble below a specific temperature, but it becomes insoluble at temperature above this point. MEEP has a a very low glass transition temperature. The glass transition temperature is the temperature at which the amorphous portions of the polymer begin to flow. MEEP can be readily cross-linked by exposure to gamma rays or UV light. Cross-linked MEEP swells in water to form stable hydrogels, the water content of which is a function of the degree of cross-linking. Allcock (1996) sites a series of other polymers with LCSTs.

2.3 Present Polymer Applications

Applications of SAPs range from absorbent infant and personal care products (Harland, 1993) to contact lenses (Pó, 1994) to waterproof, breathable fabrics (Lomax, 1990). SAPs have also been used as desiccant materials in refrigeration systems of buildings (Pó, 1994). Masuda and Iwata (1990) present their work on the dewatering effect of such polymers on different materials like coal, clay, activated sludge, and metal plating sludge. In these cases, the rate of absorption is dependent upon the material's form rather than polymer composition. In other words, the fine powder is preferred as it gives the highest surface area possible.

Recognizing the future potential and changing environmental needs, Carrier Corp. has launched an environmentally responsible water-based air conditioning system called Hydroflow (Anon., 1994). The goals of this venture are to save energy, installation and maintenance costs, and most importantly, to help the environment. The John's Hopkins School of Medicine has followed suit and has implemented a desiccant system to air condition the entire facility. DiBlasio (1995) discusses the savings that resulted from this installation. In initial capital costs alone, the facility saved \$1,000,000 over the next alternative air conditioning system and about \$2,580,000 in four years in heating and cooling costs.

Harriman (1994) published an article about the fundamentals of a commercial desiccant system which provides a good exposure to the current trends of desiccant technology. Desiccants attract moisture because of the difference in vapor pressure between the surface and the air stream. Rotational speeds of the rotary wheel desiccant systems are typically slow (6 to 10 rph). The slow diffusion of water molecules essentially dictate the slow speed of rotation. This brings forth the question of whether water diffusion can be accelerated using special, synthetic or natural substances, which in turn would speed up the humidity control process. While the development of such an ideal material is still in progress, considerable efforts have been devoted to the studying of characteristics of laboratory-synthesized polymer materials.

2.4 Existing Polymer Characterization Techniques

Several polymer characterization techniques researched during this study dealt with the determination of the maximum moisture capacities and adsorption rates of various desiccant materials. Kinetics of diffusion have also been researched by several scientists and are reported in this section. Information was also obtained regarding the evaluation of effective thermal conductivities in multi-phase composite desiccant materials.

2.4.1 Experimental Methods of Characterization

Cai et al. (1995) studied the water absorption on polymer thin films by using a quartz crystal resonator. Employing UV-initiated polymerization, the hydrophilic groups of poly(methyl methacrylate) polymers were synthesized. The films of the polymers were then deposited on quartz crystal resonators. As moisture was absorbed and desorbed, the frequency response of the quartz crystals changed accordingly. Mendizabal et al. (1995) investigated the water absorbing characteristics of hydrogels of monoitaconate eaters, 2-ethoxyethyl monoitaconate and methyl monoitaconate synthesized from itaconic acid. The effect of copolymer composition, amount of cross-linking agent, electrolyte present, and pH of the solution on water absorption rate and maximum degree of swelling of the hydrogel were also investigated.

Han et al. (1995) investigated the water sorption and diffusion characteristics of poly (4,4 prime-oxydiphenylene pyromellitimide) (PMDA-ODA), a representative polyimide, by varying the humidity, method of synthesis, and film thickness at ambient temperatures. Moisture uptake within the films was measured with an electronic microbalance. Han et al. (1995) observed the water diffusion in the films to be nearly Fickian. An increase in the mass diffusion coefficient corresponding to an increase in the humidity level was also noted. Rao et al. (1995) investigated the moisture diffusion characteristics of reinforced plastics and composites. They concluded that all specimens exhibited Fickian diffusion. They also concluded that an increase in the effective surface area resulted in a larger diffusion coefficient.

Berthold et al. (1994) studied the sorption characteristics of different wood polymers as a function of relative humidity. The effect of ions and counter-ions present in various wood polymers containing carboxylic and sulfonic acid groups on moisture uptake was studied. In the same vein, Anderna et al. (1995) evaluated the water sorption process of laboratory synthesized alkali salts of anionic polymers and copolymers, which arguably have the potential to be the next generation of desiccants. The authors suggest criteria for assessing water absorption performance and compare the results with that of commercially available polymer desiccants.

Passiniemi (1995) proposed a simple method to determine the diffusion coefficient of water in polyaniline. Since water content affects the electrical conductivity of a material, measurements of a material's change in conductivity indicate the rate at which the material takes on water. The diffusion coefficient of water can be evaluated from this absorption rate. The Cai et al. (1995) work, which monitors the frequency response of a hydrophilic polymer-coated quartz crystal resonator, provides an alternate way of measuring the amount of water absorbed, hence the diffusion coefficient.

2.4.2 Theoretical Methods of Characterization

Investigations have been carried out into the nature of the diffusion process at the desiccant/air interface as well. Ishihara and Matsumoto (1995) conducted a theoretical study of water vaporization from a super-saturated surface to a cold air stream. The combination of a super-saturated surface/cold air interface sets the stage for the potential formation of fog. If fog does form within the boundary layer, the resulting low vapor concentration results in a high mass transfer coefficient and a low heat transfer coefficient, resulting in a moderate increase in mass transfer and a remarkable decrease in the heat transfer. The results from the study by Ishihara and Matsumoto (1995) imply that fog-forming conditions must be avoided for effective heat and mass transfer.

Ketelaars et al. (1995) checked the validity of using the diffusion coefficient obtained from the macroscopic behavior of the sample, referred to as the apparent coefficient, against the actual coefficient determined experimentally. They conclude that for porous substances the apparent coefficient agreed with the actual coefficient value only under some prescribed conditions.

In response to efforts in composite desiccant material development, Sarwar and Majumdar (1995) present a model for the thermal conductivity of two-phase composite porous media, such as absorbing desiccant bed with an inert material. The model assumes knowledge of individual component thermal conductivities and volume fractions. The effect of water content, porosity, and particle size on thermal conductivity are discussed. The work completed by Sarwar and Majumdar (1995) is partially based on an older study performed by Luikov et al. (1968) which presents a model for the evaluation of the thermal conductivity of porous systems both in solid and powder form. This model was developed for a wide range of temperatures and gaseous pore pressures.

2.5 Desiccant-Enhanced Energy Exchanger Modeling

Models have also been developed to capture and predict the steady state behavior of desiccants and cooling systems. This review covers models for both solid and liquid desiccant-enhanced conditioning systems. Models discussed in this section include system configurations of rotary wheel dehumidifiers as well as fixed plate exchangers.

Kettleborough and Waugaman (1995) analyzed the thermal performance of a liquid desiccant cooling cycle using a computer program modeling two indirect evaporative coolers. It was concluded that the liquid desiccant flow rates that affect the efficiency of the desiccant tower are critical to the overall performance of this air conditioning system. Otterbein (1995) researched liquid-based desiccant applications as well. His paper *"A Theory for Heat Exchangers with Liquid-Desiccant-Wetted Surfaces"* introduces a theory that uses a wall boundary condition that forces the moist air to be at an arbitrary relative humidity that is less than 100%. The theory introduces a 'new' brine-bulb temperature used in liquid desiccant analysis. The theory basically brings the dry-surface, solid desiccant-based, and wet-surface, liquid desiccant-based heat exchangers together. Consequently, geometrically similar dry- and wet-surface heat exchangers can be compared relative to their performance.

Jain et al. (1995) evaluated various solid desiccant cycles for air conditioning in a hot and humid climate using computer simulation. Using performance data from commercially available air conditioning equipment, including a rotary wheel desiccant dehumidifier, various cooling scenarios were developed and analyzed. Ventilation, recirculation, and Dunkle cycles were evaluated. Jain et al. (1995) concluded that the Dunkle cycle, a ventilation/recirculation hybrid cycle, performed the best over a wide range of hot and humid outdoor conditions.

Banks (1971) developed a model which specifically looks at heat and adsorbate transfer in fluid flow through a porous medium . The Banks (1971) model assumes perfectly straight and parallel channels through which a fluid moves at constant velocity. This model further assumes completely reversible desorption phenomena. Banks (1971) solved coupled non-linear heat and mass transfer equations via the method of characteristics, a mathematical approach yielding a solution in terms of characteristics and waves. Close and Banks (1971) verified this earlier model with a silica-gel-water-air system.

Chant and Jeter (1995) investigated the use of parabolic concentration profiles, which use effective diffusivities, for rotary desiccant wheels. An implicit steady-state solution method is presented and the overall efficiency of the model is discussed. Zheng and Worek (1993) also examined rotary wheel desiccant behavior. A coupled heat and mass transfer finite-differencing scheme is introduced. Zheng et al. (1995) extended this earlier work to investigate the effect of rotational speed on the dehumidification performance of a rotary dehumidifier. Their most recent work involved the effect of desiccant sorption properties, the heat and mass transfer characteristics, and the size of the rotary dehumidifier on the desiccant performance. Zheng et al. (1995) emphasized the importance of rotational speed on the overall performance of the rotary desiccant system.

Research in the field of fixed plate exchangers has traditionally focused on the dynamic modeling of heat transfer between various working fluids in both counterflow and cross-flow configurations. The dynamic response of plate heat exchangers is an area of interest due to the numerous industrial applications in which plate exchangers are controlled for optimum performance (Sharifi et al., 1995). Sharifi et al. (1995) investigated such control capabilities with an extended 'cinematic' model which uses a backwards-differencing convective term. Models are executed according to explicit, implicit, and Crank-Nicolson discretization methods and compared.

The 'cinematic' model upon which the work of Sharifi et al. (1995) was based was developed by Lakshmanan and Potter (1990). In their paper *" Dynamic Simulation of Plate Heat Exchangers"*, Lakshmanan and Potter (1990) reported on their use of the 'cinematic' model to estimate both steady-state and dynamic temperature profiles. Dynamic plate exchanger response was also studied by Khan et al. (1988). In their research, the plate exchanger response was measured with respect to inlet flow and temperature changes. Khan et al. (1988) commented on frequent assumptions made when modeling plate exchanger response. These assumptions include constant fluid properties and heat transferred exclusively by convective means.

Chapter 3

Material Selection and Evaluation

The material selection process of the research proved to be of great importance. Experimental material characterization and property evaluation tests were conducted by graduate student Jennifer Howard within the Materials Science Engineering Department of Virginia Tech. The author was able to assist Ms. Howard in performing several of the experiments described in this chapter.

The design of a new total enthalpy exchanger is highly dependent on thermophysical properties such as mass and thermal diffusivities, as well as structural properties including stiffness. The candidate desiccant material must adsorb as well as desorb in a timely fashion. In addition, the material is required to transfer heat quickly. High surface area-to-volume ratios, structural flexibility, cost, and availability are examples of additional considerations that were continually addressed throughout this project. Descriptions of various material property characterization techniques are presented here as well.

3.1 Candidate Materials

The first stage of the material selection process involved several studies conducted over a field of seven candidate materials. These materials included a variety of water absorbing polymers, lower critical solution temperature polymers, as well as traditional water absorbing ceramics. Brief descriptions of the seven candidate desiccants are presented below.

- **Cross-linked polyacrylamide copolymer:** used as a soil additive to keep moisture content more constant. Known to maintain strength when swollen and readily absorbs and desorbs water. The particular material studied was called TerraSorb AG manufactured by Industrial Services International, Inc.

- **Polyacrylate:** acrylic acid cross-linked with sodium acetate, the material in diapers and other similar hygiene products. Known for selective absorbency as a result of the pH of the water.

- **Polyvinyl alcohol:** porous open-cell sheet foam capable of absorbing up to twelve times its weight in water. Also available in powder and liquid forms. Foam versions used as sponges in cosmetic and industrial applications.

- **Polyethylene glycol:** copolymer exhibiting LCST characteristics with water in lower molecular weights. Lower molecular weight materials are in liquid form. These can be used as thickeners in cosmetics and food products.

- **Polypropylene glycol:** copolymer exhibiting LCST characteristics with water in lower molecular weights. Lower molecular weight materials are in liquid form. Propylene glycol is produced for use in approved food, cosmetic, personal care and pharmaceutical applications.

- **Silica Gel:** ceramic desiccant popular for use in dehumidification applications. Obtained for comparison with the polymer desiccants.

- **Calcium sulfate:** ceramic desiccant popular for use in dehumidification applications. Obtained for comparison with the polymer desiccants.

3.2 Selection Criteria

It was decided that the candidate desiccant to best meet the predetermined selection criteria would be chosen as the material to be incorporated into the total energy exchanger designs. Initially, the selection criteria addressed concerns such as equilibrium moisture capacity, quick moisture absorption rates, and ability to desorb quickly at ambient temperatures. Commercial availability, low toxicity, and safety hazards were also considered.

Polyacrylamide $-(CHCH_2)_n-$ $O=C-NH_2$	Polyacrylate $-(CHCH_2)_n-$ $O=C-O$	Polyvinyl Alcohol $-(CHCH_2)_n-$ OH	Polyethylene Glycol $-(CH_2CH_2-O)_n-$
Polypropylene Glycol $-(CHCH_2-O)_n-$ CH_3	Silica Gel SiO_2	Calcium Sulfate $Ca — SO_4$	

Figure 3.1 Representation of Candidate Chemical Formulas

Qualitative experiments were conducted to explore each materials ability to absorb water and to determine each materials maximum sorption capacity. Similar experiments were also conducted to determine how quickly the material released the stored moisture (material desorption rates). At the same time, a general investigation was conducted into the availability of these desiccant materials from manufacturers. Once the field of candidate materials had been narrowed, an exploration was made into the toxicity and safety characteristics of the materials with the help of respective Material Safety Data Sheets (MSDS).

3.2.1 Adsorption Rate/Equilibrium Moisture Content

The seven candidate desiccant materials, with the exception of polypropylene glycol, which could only be obtained in a liquid form, were tested at high relative humidities to determine equilibrium moisture and to get an estimate of absorption rates. Using a water bath, a 95% relative humidity environment was created in a desiccator box, shown schematically in Fig. 3.2. The water absorption capacity was measured as a function of relative humidity for the pre-dried desiccant materials. A hygrometer mounted on the side of the desiccator box was used to verify the relative humidity. A brief outline of the procedures taken to conduct the absorption rate/equilibrium moisture content rate experiments follows.

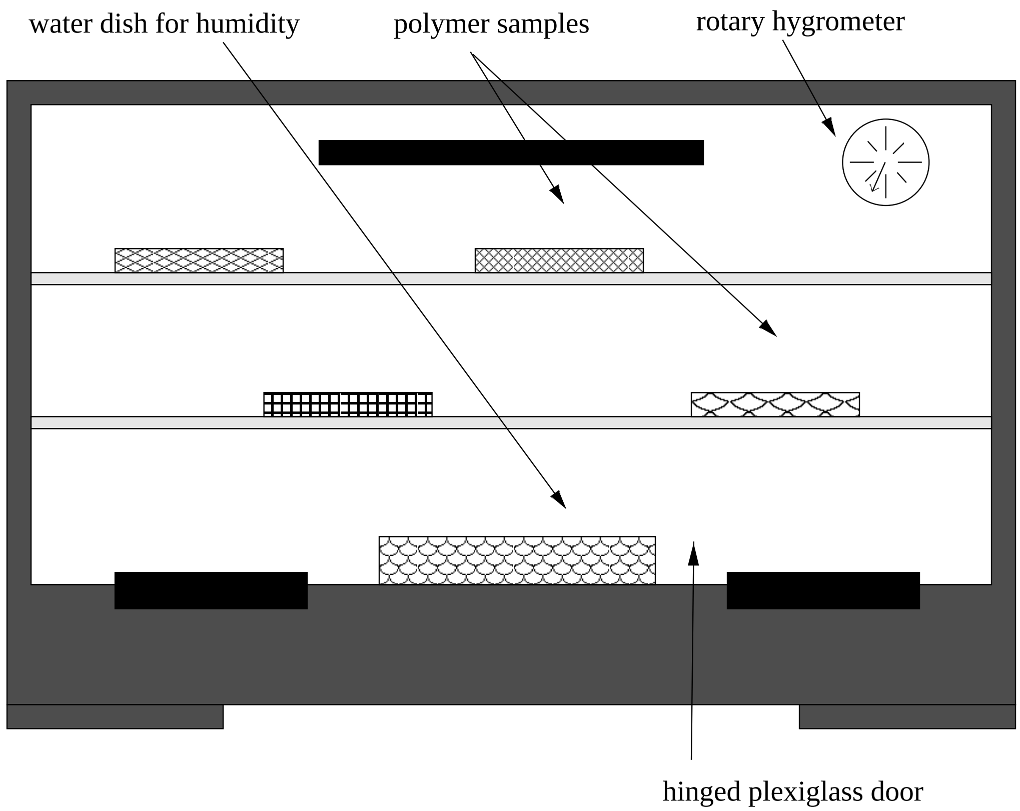

Figure 3.2 Representation of Desiccator Box, test equipment for initial uptake behavior and moisture content

Procedures for Absorption Rate/Equilibrium Moisture Content Experiments

1. Pre-dry polymer samples
   (a) Measure initial mass of samples
   (b) Place in dry box
   (c) Measure mass periodically
   (d) Drying is complete when mass is constant with time
2. Place water bath in desiccator box
3. Check for correct percent relative humidity with hygrometer
4. Plot equilibrium moisture absorption versus percent relative humidity at room temperature
   (a) Record mass of pre-dried samples

(b) Place samples in controlled humidity environment

(c) Measure mass of sample periodically

(d) Absorption is complete when sample has constant mass

5. Calculate and plot equilibrium percent water absorption vs. time at specified relative humidity

Results from the sorption experiments are presented in Fig. 3.3 and Table 3.1. It is important to keep in mind that, when discussing the adsorption rate characteristics, these initial experiments were conducted for comparative purposes only. More precise experiments were carried out later to determine actual adsorption rates. Considering this, a qualitative conclusion can be made regarding which material has the fastest absorption rate from these initial test results presented in Fig. 3.3. The material equilibrium moisture contents were also determined from the experiments shown in Fig. 3.3. The percent equilibrium moisture content by dry material weight is reported for each material in Table 3.1.

The qualitative test results indicated that the polyvinyl alcohol pre-fabricated sheeting sample absorbed moisture at the fastest rate and saturated at approximately 100% by dry weight. The polyacrylate and the polyacrylamide had slower absorption rates, and exhibited lower affinities for water with an approximate equilibrium weight uptakes of 80%-90%. It was found that the silica gel also absorbed less quickly and at a lower weight percent than the polymer materials.

It was important when analyzing the adsorption rate/equilibrium moisture content experiment results to keep the time rate of the desiccant application in mind. That is, it was imperative to consider how long the desiccant would be expected to adsorb moisture. Not only was the equilibrium moisture content important, but the time required for each material to reach its equilibrium moisture content is equally significant. In other words, if a certain material had the largest equilibrium moisture content, but required considerably more time to achieve this state with regards to other materials it might not necessarily be the best material.

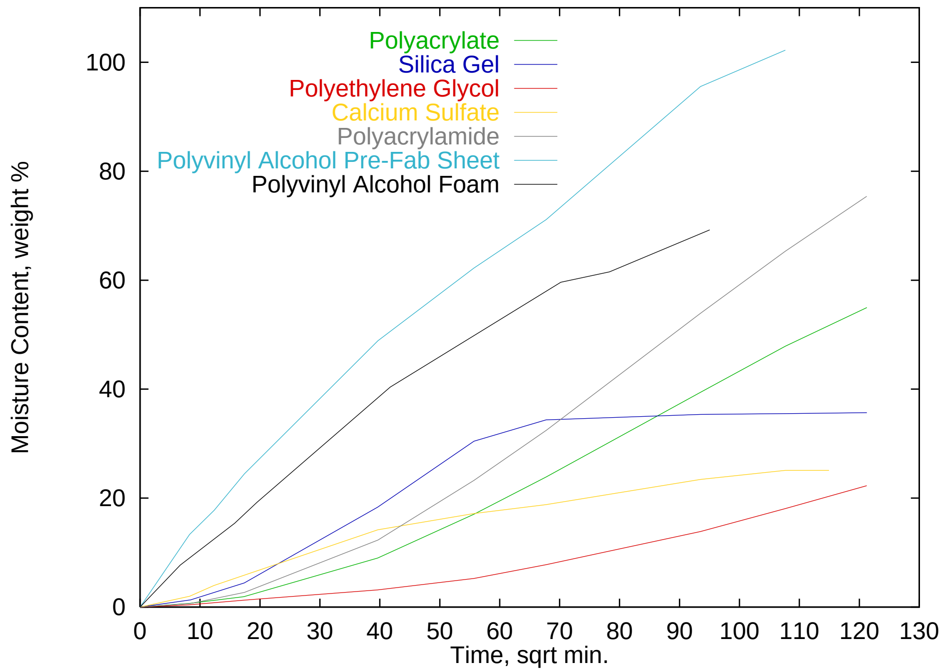

Figure 3.3 Desiccant Material Uptake Curves

3.2.2 Desorption Rate

Moisture desorption tests were conducted for the desiccant materials as well. Once the materials reached saturation in the adsorption/equilibrium moisture content experiments, the candidate materials were placed in a 35% humidity environment. In a routine similar to the absorption rate experiments previously described, desiccant material desorption rates were observed. Results of this test are presented in Fig. 3.4.

Table 3.1 Equilibrium Moisture Content Values for the Desiccant Material Candidates

Material	Equilibrium Moisture Content
Polyacrylate	78%
Polyacrylamide	90%
Polyvinyl alcohol pre-fabricated sheeting	103%
Polyvinyl alcohol foam	65%
Polyethylene glycol	33%
Silica Gel	37%
Calcium sulfate	20%

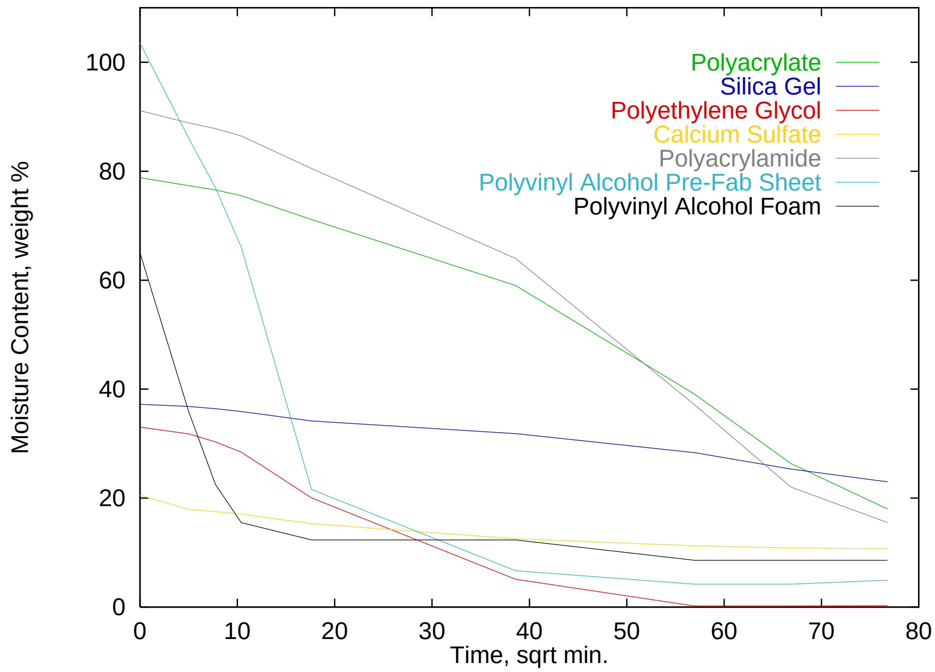

Figure 3.4 Desiccant Material Desorption Curves

3.2.3 Material Availability

During the material selection phase, members of the Materials Science Engineering Department made several industrial contacts regarding the commercial availability and feasibility of using the candidate materials in large scale energy exchanger manufacturing applications. MSE Department members also looked into material synthesis recipes in an effort to determine the possibility of producing materials in-house.

3.2.4 Material Safety

Toxicity and safety characteristics of three selected materials were explored by reviewing the respective Material Safety Data Sheets (MSDS). Summaries of the findings for each material are presented below.

P-Series Polypropylene Glycols

The polypropylene glycol P4000, P2000, and P1200 are considered to be low in acute oral toxicity, while the polypropylene glycol P425 is considered to be low to moderate in acute oral toxicity. Excessive ingestion of these substances could cause harm to the central nervous system. Any amounts ingested in association with normal industrial handling are considered harmless. Direct contact with eyes could cause mild temporary pain; the use of safety glasses is advised. These polymers are non-irritating to the skin.

The American Industrial Hygiene Association workplace environmental exposure level (AIHA WEEL) is reported as 10 mg/m^3. To avoid any respiratory protection requirements, appropriate ventilation should be provided to keep airborne levels below these exposure guidelines. The disposal of unused or uncontaminated materials should be handled be a licensed recycler, reclaimer, or incinerator.

These materials are stable under normal handling conditions. It is recommended that interaction with oxidizing materials and strong acids be avoided. Combustion could produce carbon dioxide and carbon monoxide along with unidentified organic compounds. Spills of theses polyglycols on hot fibrous insulations may lead to the lowering of the auto-ignition temperature (365 °F under normal conditions), possibly resulting in spontaneous combustion.

E-Series Polyethylene Glycols

The polyethylene glycol E4000, E200, E1450, and E8000 are considered to be low in acute oral toxicity. Excessive ingestion of these substances could cause harm to the renal system. Any amounts ingested in association with normal industrial handling are considered harmless. Direct contact with eyes could cause mild temporary pain; the use of safety glasses is advised. Under prolonged exposure, these polymers may be slightly irritating to the skin. There are studies showing that toxic, even fatal levels of polyethylene glycols could be absorbed through damaged skin, as in burn wounds. The use of gloves should be advised.

The AIHA WEEL is reported as 10 mg/m^3. To avoid any respiratory protection requirements, appropriate ventilation should be provided to keep airborne levels below these exposure guidelines. The disposal of unused or uncontaminated materials should be handled be a licensed recycler, reclaimer, or incinerator.

These materials are stable under normal handling conditions. It is recommended that interaction with oxidizing materials and strong acid be avoided. Decomposition can occur at elevated temperatures; hazardous decomposition products could include aldehydes, ketones, organic acids, and polymer fragments. The national fire protection association (NFPA) gives these materials a flammability rating of 1. This indicates that the material must be pre-heated before ignition will occur.

Polyvinyl Alcohol (PVOH) Granules/Powder

Polyvinyl alcohol is considered to be relatively non-toxic by ingestion. Any amounts in association with normal industrial handling are considered harmless. However, it is advised to be careful of stirring up dust since excessive concentrations acquired via inhalation could cause discomfort. Direct contact with eyes is reported to be non-irritating. PVA is slightly irritating to the skin; it is not a sensitizer, but certain additives may cause sensitization dermatitis.

There are no OSHA, ACGIH, or NIOSH occupational exposure limits established for this substance. Due to the physical properties of the powdered material, the use of respirators during the handling of this material is recommended. In addition, explosion-proof local exhaust ventilation should be provided to keep airborne levels below these exposure guidelines. The disposal of unused or uncontaminated materials should be handled be a licensed recycler, reclaimer, or incinerator.

These materials are stable under normal handling conditions. Therefore, interaction with strong oxidizing materials, excessive heat, sparks, or open flame should be avoided. The national fire protection association (NFPA) gives these materials a flammability rating of 2. This indicates that the material must be moderately heated or exposed to relatively high ambient temperatures before ignition will occur. Thermal decomposition products could include toxic carbon oxides.

3.2.5 Desiccant Selection Summary

After all of the initial material selection experiments had been completed and reviewed, it was determined that the candidate polymer desiccant polyvinyl alcohol would become the desiccant materials of focus for the project. There were several features of the selection criteria that led to this decision. Most importantly, polyvinyl alcohol had the fastest rate of adsorption of all the material candidates; this material characteristic proved later to be directly related to mass diffusivity. The fact that desorption could be accomplished at low regenerative temperatures was also considered an advantage. In addition, film and foams, material forms of interest, were proven safe to use in heat recovery ventilation applications. Lastly, the flexibility of polyvinyl alcohol regarding material processibility along with the knowledge that manufacturers exist for this material and all of its forms provided another reason to conclude that polyvinyl alcohol should be chosen as the desiccant material of focus.

3.3 Polyvinyl Alcohol Material Properties

Once polyvinyl alcohol was selected as the desiccant material of focus, more extensive tests were conducted to determine material properties needed for the completion of the modeling phase discussed in Chapter 4. Heat transfer properties including thermal diffusivity and heat transfer coefficients needed to be determined. Mass transfer properties like Fick's mass diffusion coefficient and mass transfer coefficients, also needed to be established. Finally, it was concluded that physical and structural material properties such as density, pore size (if applicable), and stiffness would also need to be taken into consideration.

To assure more accurate property estimates, more detailed adsorption rate/equilibrium moisture content experiments were run in an environmental chamber. The Cole-Parmer Automatically Controlled Environmental Chamber ®, a 13 cubic foot chamber, offered control of temperature and humidity conditions within the enclosed environment. A self-contained, automatic, internal controller operated the humidifier and dehumidifier while displaying the current and set point humidity and temperature readings on two externally mounted LCD displays. Conditions within the chamber were kept uniform through the use of a circulating fan located along with a heater and AC outlet behind a perforated aluminum housing on the back wall of the chamber. An external humidifier and a $CaCO_4$ desiccator tube provided a method for humidity control. Neoprene gloves allowed desiccant test samples to handled within the controlled environment. A schematic of the environmental chamber is presented in Fig. 3.5. A computer program was written to record data resulting from the moisture uptake experiments in the environmental chamber. Using an electronic balance, the program recorded the changing weight as a function of time. Sample moisture uptake data obtained from the environmental chamber is included in Appendix B.

In addition to determining appropriate material properties, since polyvinyl alcohol was available in several forms, it was essential to establish which polyvinyl alcohol material forms would be used for the various energy transfer scenarios. Material selection within the realm of polyvinyl alcohol was an iterative process for each exchanger application. Since the

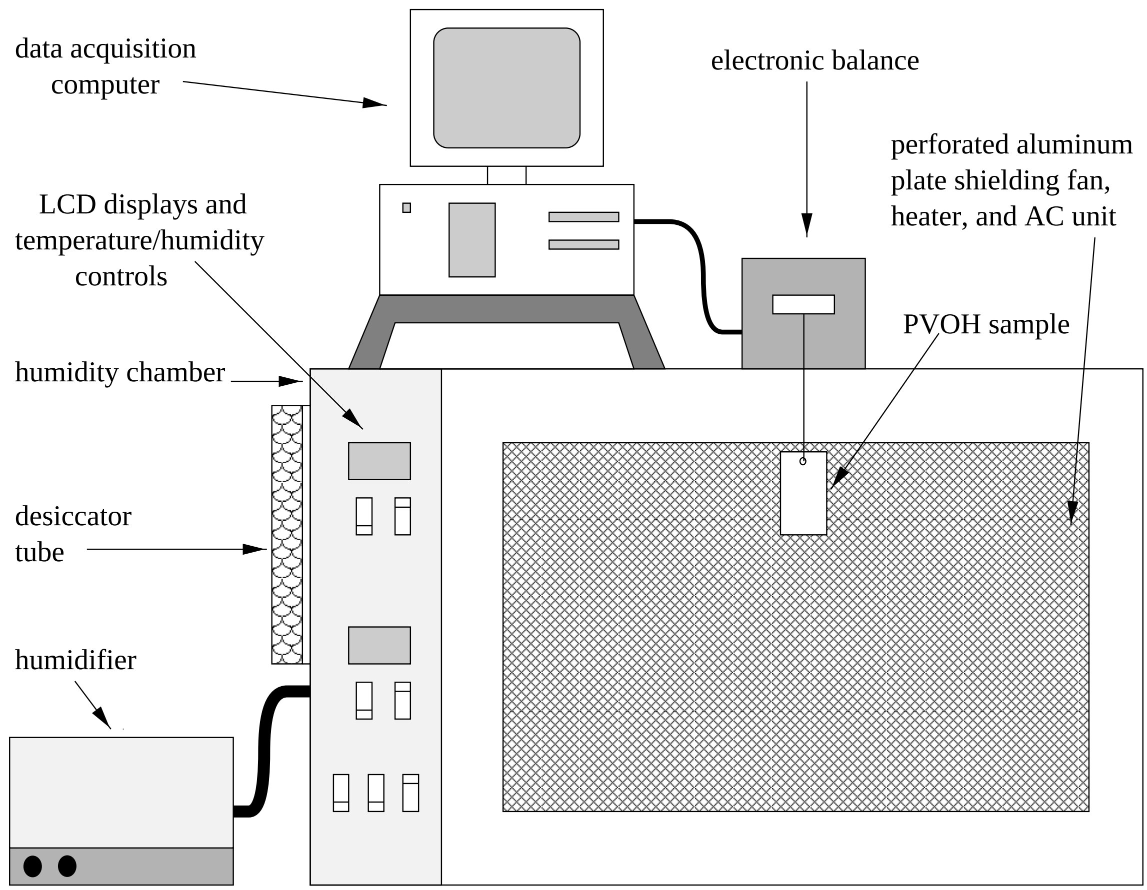

Figure 3.5 Humidity Chamber Schematic

polyvinyl alcohol flexible pre-fabricated sheeting exhibited the best behavior in the initial uptake test, it was decided that the first study would focus on implementing a polyvinyl alcohol coating onto an existing aluminum rotary wheel. However, the resulting model output, presented and discussed in Chapter 5, suggested an investigation into a form of polyvinyl alcohol that could provide an increased surface area. At this point, the use of polyvinyl alcohol open-cell foam was incorporated into the scope of the project. Polyvinyl alcohol foam was an attractive option because its porosity allows its use in both the rotary wheel total enthalpy exchanger and the fixed plate total enthalpy exchanger systems. In addition to using foam, the research eventually included the consideration of a desiccant coating comprised of a polyvinyl alcohol binder/silica gel/molecular sieve composite.

3.3.1 Heat Transfer Properties

Methods used in the evaluation of thermal properties are presented below.

Thermal Diffusivity

Thermal diffusivity was determined within the MSE Department by using the laser flash thermal diffusivity technique. The laser flash method is a transient technique for measuring the thermal diffusivity of materials. The front surface of the sample is irradiated with a pulse of energy from a high-power laser source. An infra-red detector measures the temperature change of the sample's back surface as a function of time. The thermal diffusivity of the material is obtained from the rise time of the measured temperature curve.

Specific Heat and Heat of Vaporization

Specific heat as well as material heat of vaporization were determined using a differential scanning calorimeter (DSC). This material characterization method yields peaks relating to endothermic and exothermic transitions in addition to showing changes in heat capacity. This method can also give quantitative information relating to enthalpic changes within the polymeric material. A DSC performs these measurements by comparing temperature differences between a sample material and a reference material, typically powdered alumina. Employing a servo system to supply energy at a varying rate to both the sample and the reference, the DSC records the energy that must be supplied against the average temperature in order to keep the sample and the reference at equal temperatures. When an exothermic or endothermic change occurs in a sample material, power (energy) is applied to one or both calorimeters to compensate for the energy change in the sample. Using plotted output, specific heat can be determined by measuring the energy difference associated with the slope before and after the material's glass transition temperature has been exceede. The heat of vaporization can be calculated by analyzing the area under the exothermic peak of the energy curve.

Heat Transfer Coefficient

Convection heat transfer coefficients were determined mathematically using the traditional method of evaluating effective airflow diameter, polyvinyl alcohol material conductivity, and effective Nusselt number. The methods used to evaluate the thermal properties discussed in this chapter are further outlined in Appendix A.

3.3.2 Mass Transfer Properties

Methods used in the evaluation of mass transfer properties are presented below.

Mass Diffusivity

Polyvinyl alcohol mass diffusivity as well as polyvinyl alcohol mass transfer coefficients were calculated using information gathered during the adsorption rate experiments. Sample data from these experiments is included in Appendix B. Diffusion in polymeric systems is passive if the driving force is purely a Brownian molecular motion, but diffusion can also be activated by external effects, either by the influence of the release medium by swelling or biodegradation, or by the effects of physical forces such as electrical, osmotic or convective forces. Fundamentally, diffusion is described by Fick's first law which describes the macroscopic transport of molecules by a concentration gradient:

$$\frac{\partial W}{\partial t} = D_{ab} \frac{\partial^2 W}{\partial y^2}. \tag{3.1}$$

An effective mass diffusion coefficient, D_{ab}, was calculated from an semi-infinite medium solution using the data from the environmental chamber. It is assumed that initially the concentration in the desiccant is zero, the surface concentration is constant, and the operating times of interest are short enough that the material can be modeled as a semi-infinite plate. The semi-infinite medium solution is represented as

$$W = W_{surface}[1 - erf(\frac{x}{2\sqrt{D_{ab}t}})]. \tag{3.2}$$

Mass Transfer Coefficient

The mass transfer coefficients were calculated by analyzing the the polyvinyl alcohol adsorption rate experiments as well. The mass transfer coefficients dictate how effectively a material will take on moisture at the material/air stream interface. By studying the change in slope of the moisture adsorption curves as the material moisture content changed, the mass transfer coefficient was extracted.

Methods for obtaining mass diffusivities and mass transfer coefficients are further outlined in Appendix A. Maximum moisture content was also of importance in the analysis as discussed in Chapter 4.

3.3.3 Physical and Structural Properties

Properties such as density were provided by the material manufacturer. Structural details including pore size, etc. were also supplied by the material manufacturer. These properties played an important role in evaluating the physical limitations of the energy exchanger applications with regard to the desiccant material. In addition, pore size delegated the desiccant surface area available to the conditioned airspace within the exchanger equipment.

3.3.4 Summary of Properties

Due to information gathered from the results of model runs presented in Chapter 5, three different forms of polyvinyl alcohol were ultimately studied. Polyvinyl alcohol film (a coating developed in the lab), polyvinyl alcohol foam, and the polyvinyl alcohol/ceramic composite were evaluated for use in a rotary wheel total enthalpy exchanger system while the performance of polyvinyl alcohol foam was estimated for use in a fixed plate total enthalpy exchanger. The methods covered in this chapter were used to determine material properties for these three forms of polyvinyl alcohol. These properties are summarized in Table 3.2.

Table 3.2 Estimated Properties for Polyvinyl alcohol Materials

Material Property	Polyvinyl alcohol Film	Polyvinyl alcohol Foam	Polyvinyl alcohol Composite
Density, ρ (kg/m^3)	1310.	100.	770.
Specific Heat, C_d $(kJ/kg \cdot K)$	1500.	1040.	1025.
Thermal Diffusivity, α (m/s)	1.5 x 10^{-6}	1.6 x 10^{-5}	2.6 x 10^{-7}
Thermal Conductivity, k $(W/m^2 \cdot K)$	0.29	1.44	0.20
Mass Diffusivity, D_{ab} (m^2/s)	2.0 x 10^{-13}	2.5 x 10^{-9}	7.0 x 10^{-7}
Mass Transfer Coefficient, K_y $(kg/m^2 \cdot s)$	order of 10^{-6}	order of 10^{-3}	order of 10^{-4}
Maximum Moisture Content, W_{max} $(kg_{water}/kg_{desiccant})$	0.156	0.07	0.15

Chapter 4

Theoretical Analysis

As stated previously in Chapter 1, this research originated with the idea to develop and model innovative desiccant materials for existing total energy exchange applications; specifically, total energy exchange applications associated with Heating Ventilation and Air Conditioning (HVAC) systems. The study focused on two classes of exchangers: rotary wheel total enthalpy exchangers and fixed plate total enthalpy exchangers. Since these two types of applications operate in two completely distinct ways, the modeling efforts were different for each type.

The rotary wheel total enthalpy exchanger operates as a revolving exchanger circulating between two opposing air streams. The physics of this mechanism dictate a close look into the surface phenomena taking place as the wheel rotates between contrasting conditions transferring heat and moisture. With regenerative surface effects of this kind, a lumped capacitance assumption can be used. This assumption allows for the rotary wheel total enthalpy exchanger model to be described by a one-dimensional model solution. With lumped capacitance effects occurring at the desiccant material surface, a one-dimensional solution can be formulated considering temperature and moisture effects along the length of the wheel channels.

In contrast to the rotary wheel total enthalpy exchanger the fixed plate total enthalpy exchanger operates in a stationary mode, as its name insinuates. This configuration involves a completely different group of transfer kinetics than those required in the rotary wheel total enthalpy exchanger situation. The immobile position of the fixed plate total enthalpy exchanger eliminates the possibility for a regenerative cycle on the surfaces of the desiccant plate. Therefore, to transfer heat and moisture, the plate must have some method of diffusing the heat and mass. Accounting for diffusion through the thickness of the working material introduces a second dimension to the problem; now we are not only interested in the temperature and moisture effects along the length of the plate, we are also interested in the conditions through the thickness of the desiccant polyvinyl alcohol material.

This chapter illustrates the two different numerical schemes, a one-dimensional and a two-dimensional method, used to solve for the performance estimates of the rotary wheel total enthalpy exchanger and the fixed plate total enthalpy exchanger respectively. In the following sections, the governing differential equations and the finite-difference equations will be introduced in an effort to outline the solution methods.

4.1 Mathematical Model of One Dimensional Rotary Wheel

This project resulted in the examination of two material forms, film and foam, of polyvinyl alcohol for use as a desiccant coating for the rotary wheel total enthalpy exchanger. Upon survey, it was found that, due to the structural differences, polyvinyl alcohol film and polyvinyl alcohol foam possess dissimilar sorption characteristics. For this reason, two one-dimensional rotary wheel total enthalpy exchanger models were created. More specifically, a program was written estimating the performance of polyvinyl alcohol film which takes on moisture via *absorption* while a separate program was developed to estimate the performance of polyvinyl alcohol foam which takes on moisture by both *adsorption* and *absorption*. Descriptions of these two dynamic moisture uptake mechanisms, absorption and adsorption, are addressed within the discussions of their respective models.

Despite the absorption/adsorption differences, there were several features mutually employed by both rotary wheel total enthalpy exchanger models. For instance, the models used like assumptions. The assumptions used during the development of the one-dimensional mathematical model of the rotary wheel total enthalpy exchanger are:

- Keeping in mind that typical rotary wheel total enthalpy exchangers using existing desiccant technology can accommodate over 100,00 channels in a 3 ft. diameter wheel, it was assumed that the air channel height in such a wheel is so minimal that no substantial gradient would exist within the height of the air stream.

- The use of a lumped capacitance scheme led to a similar assumption within the desiccant material; it was assumed that the desiccant layer upon the supporting structure is sufficiently thin (on the order of microns) that gradients within the thickness of the material would be negligible.

- In addition to the assumptions made with regard to the height of the air stream and the thickness of the material, it was also assumed that transfer of heat and mass axially by diffusion along the length of the dehumidifier, both in the air and in the material, was negligible.

- The model also presumed that the heat and mass transfer coefficients of the wheel were constant throughout the apparatus.

- All channels were assumed to be adiabatic.

The actual program codes used to model each rotary wheel total enthalpy exchanger system were also very similar. For instance, each program offered a choice of three wheel channel geometries to use when analyzing wheel product performance: triangular, hexagonal, and sinusoidal. Depictions of each channel geometry grouping, as they would appear in actual rotary wheel total enthalpy exchangers are presented in Fig. 4.1.

In addition, both rotary wheel total enthalpy exchanger models utilized the same logic in their solution methods. Similar finite difference schemes were also used for each model. The method of solution used to solve both rotary wheel total enthalpy exchanger models is discussed further in Section 4.1.5 while the development of the finite difference equations is outlined next in Section 4.1.1.

4.1.1 One Dimensional Equation Development

Although the performance estimates were ultimately obtained using a numerical scheme, both analytical and numerical equations were initially developed. The governing differential equations for the one-dimensional rotary wheel total enthalpy exchanger model were derived using a control volume technique. Simultaneously, using the theories of conservation of energy and conservation of mass, finite difference equations were derived.

A differential element, with length dx, was chosen and analyzed with respect to the principles of conservation of energy and conservation of mass. A schematic depicting the control volume analysis used to illustrate conservation of energy is represented in Fig. 4.2.

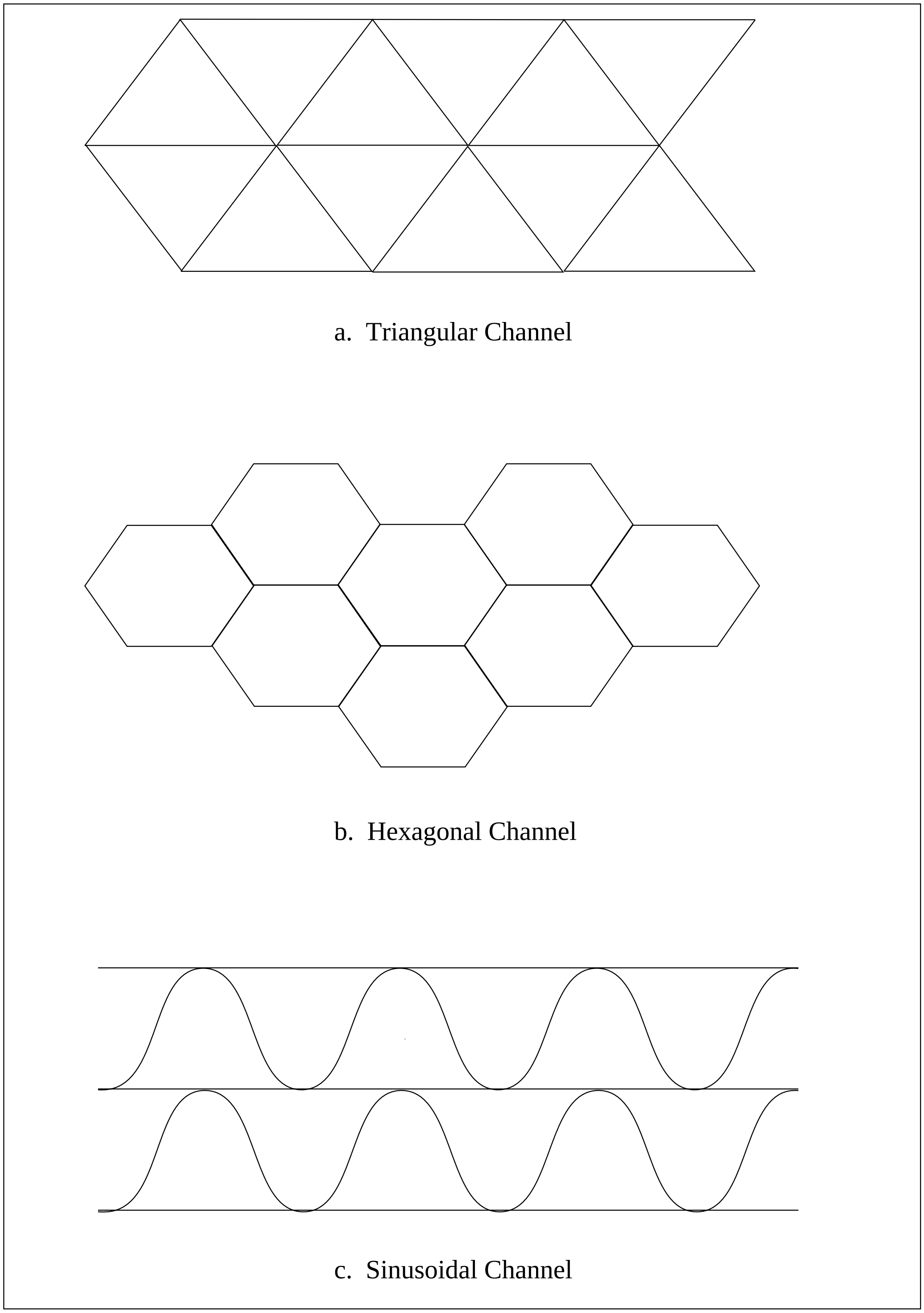

Figure 4.1 Channel Geometries as Incorporated into the Rotary Wheel Exchangers

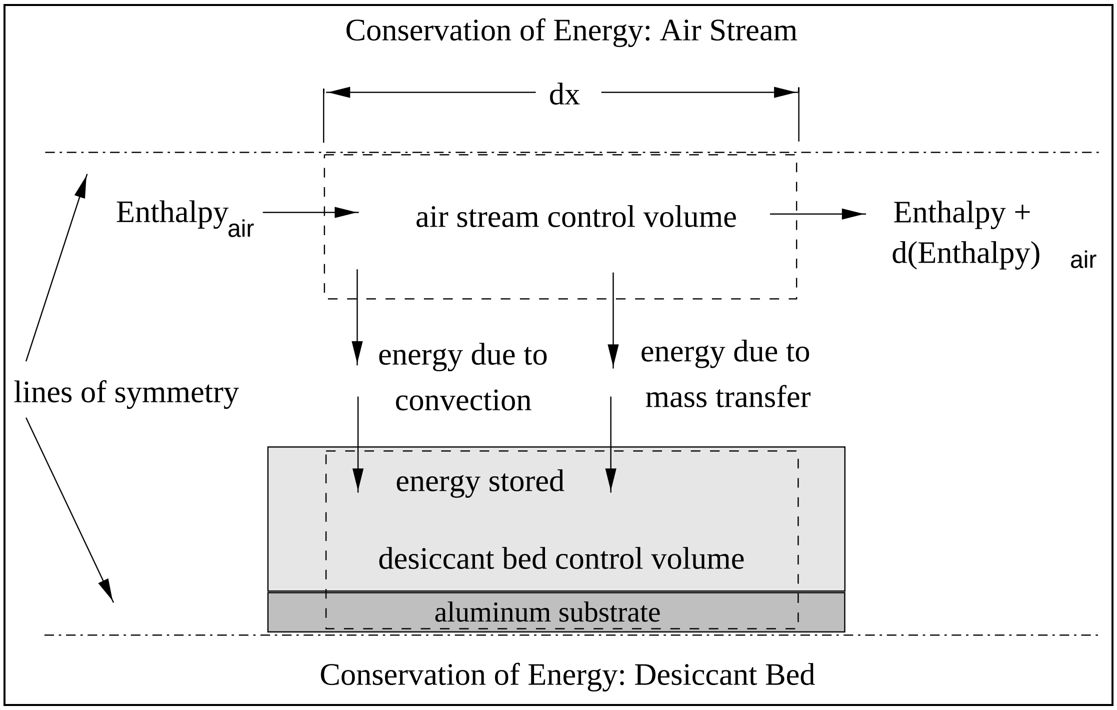

Figure 4.2 Control Volume Schematic Representing Conservation of Energy for Rotary Wheel Scenario

While considering the assumptions outlined above, examining an incremental distance of air stream/material interface within an incremental portion of time allowed for the development of differencing equations which could be translated to yield the governing differential equations presented later in this chapter.

Although the governing differential equations for both rotary wheel models were different,the same numerical formulation was used to develop the corresponding finite difference equations for both systems. This finite difference scheme is presented below. Parallel development of both the governing differential equations and the numerical finite difference equations is shown for the absorbing and adsorbing rotary wheel exchanger models in Sections 4.1.3 and 4.1.4, respectively.

4.1.2 Finite Difference Method

The method employed to develop the finite difference equations for both rotary wheel models consisted of a quasi-central-difference formulation. These difference equations were evaluated in time and space according to the techniques introduced by Zheng and Worek (1993). Using hypothetical variables A, B and C, the differencing scheme is outlined below.

Using a forward-differencing approach in time allowed the changing variables with respect to time to be written in an explicit fashion. In other words, the equation was written in a way that allowed for the explicit solution for variable A at time step $k+1$ by defining it in terms of known variables at time step k and positions $n+1$ and n. The equation associating a differential expression in time with an explicit expression in time is

$$\frac{\partial A}{\partial t} = (B + C) \quad \Leftrightarrow \quad \frac{(A^{\mathrm{k+1}}_{n+1} - A^{\mathrm{k}}_{n+1})}{\Delta t} = (B^{\mathrm{k}}_{n+1} + C^{\mathrm{k}}_{n+1}). \tag{4.1}$$

Similarly, it is possible to obtain a parallel expression in space using the backwards-differencing technique. At time step $k+1$, it is possible to view changing variables at node $n+1$ by defining the variables in terms of known variables at node n. The equation associating a differential expression in space with a backwards-difference expression in space is

$$\frac{\partial A}{\partial x} = (B + C) \quad \Leftrightarrow \quad \frac{(A^{\mathrm{k+1}}_{n+1} - A^{\mathrm{k+1}}_{n})}{\Delta x} = (B^{\mathrm{k+1}}_{n} + C^{\mathrm{k+1}}_{n}). \tag{4.2}$$

It is also possible to obtain these relationships by evaluating the time change of A at either time step $k + \frac{1}{2}$ or node $n + \frac{1}{2}$ with the central-difference formula. This unusual central-difference scheme, presented by Zheng and Worek (1993) and used in this study, presents a different method with which to evaluate variables changing in time and space. Using the central-difference concept, variables evaluated at time step $k + \frac{1}{2}$ and node $n + \frac{1}{2}$ will transform Eq. 4.1 into

$$\frac{(A^{\mathrm{k+1}}_{n+1} - A^{\mathrm{k}}_{n+1})}{\Delta t} = (B^{\mathrm{k}+\frac{1}{2}}_{n+1} + C^{\mathrm{k}+\frac{1}{2}}_{n+1}). \tag{4.3}$$

Likewise, Eq. 4.2 will be transformed as follows with the same technique:

$$\frac{(A^{\mathrm{k+1}}_{n+1} - A^{\mathrm{k+1}}_{n})}{\Delta x} = (B^{\mathrm{k+1}}_{n+\frac{1}{2}} + C^{\mathrm{k+1}}_{n+\frac{1}{2}}). \tag{4.4}$$

By averaging, Zheng and Worek (1993) developed definitions to finalize Eqs. 4.3 and 4.4:

$$B^{\mathrm{k+1}}_{n+\frac{1}{2}} = \frac{1}{2}(B^{\mathrm{k+1}}_{n+1} + B^{\mathrm{k+1}}_{n}) \quad , \quad C^{\mathrm{k+1}}_{n+\frac{1}{2}} = \frac{1}{2}(C^{\mathrm{k+1}}_{n+1} + C^{\mathrm{k+1}}_{n}) \tag{4.5}$$

$$B^{\mathrm{k}+\frac{1}{2}}_{n+1} = \frac{1}{2}(B^{\mathrm{k+1}}_{n+1} + B^{\mathrm{k}}_{n+1}) \quad , \quad C^{\mathrm{k}+\frac{1}{2}}_{n+1} = \frac{1}{2}(C^{\mathrm{k+1}}_{n+1} + C^{\mathrm{k}}_{n+1}). \tag{4.6}$$

Using Eqs. 4.5 and 4.6, the final centrally-differenced equations are presented below.

Although the left-hand-side of the following equation appears to be explicit with respect to time, the averaging technique used to determine the right-hand-side indicates a

Crank-Nicolson (central-difference) formulation.

$$\frac{(A^{\mathrm{k}+1}_{n+1} - A^{\mathrm{k}}_{n+1})}{\Delta t} = \frac{1}{2}[(C^{\mathrm{k}+1}_{n+1} + C^{\mathrm{k}}_{n+1}) + (B^{\mathrm{k}+1}_{n+1} + B^{\mathrm{k}}_{n+1})] \tag{4.7}$$

By the same means, the left-hand-side of this next equation appears to be backwards-differenced with respect to space, but the averaging technique used to determine the right-hand-side implies a central-difference formulation.

$$\frac{(A^{\mathrm{k}+1}_{n+1} - A^{\mathrm{k}+1}_{n})}{\Delta x} = \frac{1}{2}[(B^{\mathrm{k}+1}_{n+1} + B^{\mathrm{k}+1}_{n}) + (C^{\mathrm{k}+1}_{n+1} + C^{\mathrm{k}+1}_{n})] \tag{4.8}$$

The method described here was implemented into the rotary wheel performance solutions. By using pre-determined information at time step k and node n, a series of equations was generated for each system. The solution of unknowns at $k+1$ and $n+1$ was then accomplished via substitution. The equations used in each system solution are presented in the following sections.

4.1.3 Absorbing Rotary Wheel System of Equations

The absorbing rotary wheel total enthalpy exchanger employed polyvinyl alcohol film as the operating desiccant. This is a completely closed-cell absorbing material. Absolute absorbing substances exhibit strictly surface sorption characteristics. In other words, absorption is solely a surface phenomenon; as moisture comes into contact with the polyvinyl alcohol film surface, it assimilates into the desiccant material. Once the moisture is within the material it becomes part of the homogeneous desiccant coating; this occasionally results in material swelling. In the film-coated rotary wheel total enthalpy exchanger model, absorption and desorption are determined via a comparison of moisture *capacity* between the air stream and the desiccant bed. This moisture capacity is defined as the ratio of the actual moisture content ($W(x,t)$) to the maximum allowable moisture content (W_{max}). The desiccant bed moisture capacity and the air stream moisture capacity are defined as

$$W^* = \frac{W(x,t)}{W_{max}} \tag{4.9}$$

and

$$Y_a^* = \frac{Y_a(x,t)}{Y_{a_{max}}}, \tag{4.10}$$

respectively.

Four equations were used for the absorbing rotary wheel total enthalpy exchanger numerical analysis: conservation of energy within the air stream (Eq. 4.11), conservation of energy within the desiccant bed (Eq. 4.13), conservation of mass within the air stream (Eq. 4.15), and finally conservation of mass within the desiccant bed (Eq. 4.17). These equations, both in differential and finite-difference forms, represented the fundamental theories employed throughout the completion of the control volume analysis used to solve for the performance of polyvinyl alcohol film as a desiccant coating for a rotary wheel total enthalpy exchanger.

The first equation describes the effects of convective terms on the air temperature. Energy is transferred amongst the air and desiccant material by way of convective heat and mass transfer. This equation states that the enthalpy change in the air stream along the desiccant channel is a result of two convective phenomena. The first term on the right-hand-side represents the energy exchange associated with convective mass transfer; enthalpy associated with the transfer of water vapor is regulated by the mass transfer coefficient, K_y. The starred variables in this term represent moisture-content capacities of both the desiccant bed, W^*, and the air stream, Y_a^*. The second term on the right-hand-side represents the energy associated with the convective transfer of heat by the temperature difference between the air stream and the desiccant bed; this is regulated by the convective heat transfer coefficient, h.

$$\frac{\partial H_a}{\partial x} = \frac{PK_yH_v}{\rho V_a A_c}(W^* - Y_a^*) + \frac{Ph}{\rho V_a A_c}(T_w - T_a) \tag{4.11}$$

The finite difference form of this conservation of energy expression, using the differencing technique of Eq. 4.8, is

$$\begin{aligned}\frac{(C_{da} + Y_{a_n}^{\mathrm{k+1}}C_v)}{\Delta x}(T_{a_{n+1}}^{\mathrm{k+1}} - T_{a_n}^{\mathrm{k+1}}) = \\ \frac{PK_yH_v\Delta x}{2\rho V_a A_c}[(\frac{W_{n+1}^{\mathrm{k+1}}}{W_{max}} + \frac{W_n^{\mathrm{k+1}}}{W_{max}}) - (\frac{Y_{a_{n+1}}^{\mathrm{k+1}}}{Y_{a_{max}}} + \frac{Y_{a_n}^{\mathrm{k+1}}}{Y_{a_{max}}})] + \\ \frac{Ph\Delta x}{2\rho V_a A_c}[(T_{w_{n+1}}^{\mathrm{k+1}} + T_{w_n}^{\mathrm{k+1}}) - (T_{a_{n+1}}^{\mathrm{k+1}} + T_{a_n}^{\mathrm{k+1}})].\end{aligned} \tag{4.12}$$

The equation expressing the conservation of energy within the desiccant bed relates the same thermal transfer discussed above from the perspective of the desiccant bed. Similar to Eq. 4.11, the desiccant bed conservation of energy equation relates the effects of convective terms on the desiccant bed temperature. This equation considers the amount and type of substrate upon which the desiccant is mounted. Consideration of all materials contributing to bed composition is necessary for determination of the effective specific heat term which is embedded in the expression describing the change in enthalpy with respect to time, on the left-hand-side. As in Eq. 4.11, the energy exchange is governed by two convective phenomena: enthalpy exchange resulting from the transfer of water vapor regulated by the mass transfer coefficient, K_y, and the energy transfer resulting from the convective transfer of heat by the temperature difference between the desiccant bed and the air stream. These two transfer expressions are represented by the first and second terms of the right-hand-side, respectively:

$$\frac{\partial H_w}{\partial t} = \frac{PK_yH_vD_w}{M_w}(Y_a^* - W^*) + \frac{PhD_w}{M_w}(T_a - T_w). \tag{4.13}$$

The finite difference version of the conservation of energy within the desiccant bed, differenced according to Eq. 4.7, is

$$\begin{aligned}\frac{(1-f)C_s + f(C_d + C_vW_n^{\mathrm{k}+1})}{\Delta t}&(T_{w_{n+1}}^{\mathrm{k}+1} - T_{w_{n+1}}^{\mathrm{k}}) = \\ &\frac{PK_yH_vD_w}{M_w}[(\frac{Y_{a_{n+1}}^{\mathrm{k}+1}}{Y_{a_{max}}} + \frac{Y_{a_{n+1}}^{\mathrm{k}}}{Y_{a_{max}}}) - (\frac{W_{n+1}^{\mathrm{k}+1}}{W_{max}} + \frac{W_{n+1}^{\mathrm{k}}}{W_{max}})] + \\ &\frac{PhD_w}{M_w}[(T_{a_{n+1}}^{\mathrm{k}+1} + T_{a_{n+1}}^{\mathrm{k}}) - (T_{w_{n+1}}^{\mathrm{k}+1} + T_{w_{n+1}}^{\mathrm{k}})].\end{aligned} \tag{4.14}$$

The next relationship covers the behavior of the air with respect to moisture content as energy is transferred convectively to and from the desiccant material surface. Conservation of mass within the air stream is maintained by evaluating the differences in moisture capacity between the desiccant bed, W^*, and the air stream, Y_a^*. Regulated by the mass transfer coefficient K_y, the moisture capacity difference acting over the mass transfer surface area alone determines the change in moisture content Y_a with respect to distance along the desiccant channel. This is expressed by

$$\frac{\partial Y_a}{\partial x} = \frac{PK_y}{\rho V_a A_c}(W^* - Y_a^*). \tag{4.15}$$

In addition, the finite difference representation of Eq. 4.15 is

$$\frac{(Y_{a_{n+1}}^{\mathrm{k}+1} - Y_{a_n}^{\mathrm{k}+1})}{\Delta x} = \frac{PK_y}{2\rho V_a A_c}[(\frac{W_{n+1}^{\mathrm{k}+1}}{W_{max}} + \frac{W_n^{\mathrm{k}+1}}{W_{max}}) - (\frac{Y_{a_{n+1}}^{\mathrm{k}+1}}{Y_{a_{max}}} + \frac{Y_{a_n}^{\mathrm{k}+1}}{Y_{a_{max}}})]. \tag{4.16}$$

Lastly, the conservation of mass expression for the polyvinyl alcohol film-coated wheel depicts the manner in which the desiccant absorbs and desorbs water as it responds

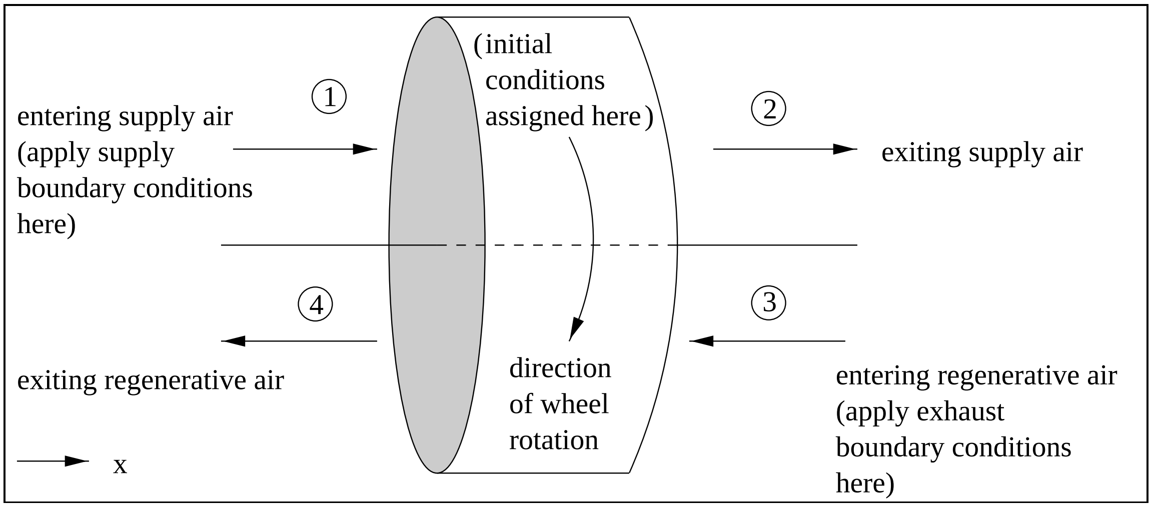

Figure 4.3 Representative Rotary Wheel Total Enthalpy Exchanger Environment

to the relative humidity of the convective air stream. In a manner complimentary to that of the conservation of mass in the air stream, the conservation of mass of the desiccant bed is also determined by the regulated differences in moisture capacity between the desiccant bed, W^*, and the air stream, Y_a^*. The effect of this difference is again controlled by the convective mass transfer coefficient K_y and influences the manner in which the moisture content of the desiccant bed changes with respect to time:

$$\frac{\partial W}{\partial t} = \frac{PK_yD_w}{fM_w}(Y_a^* - W^*). \tag{4.17}$$

The corresponding finite difference form of the conservation of mass expression is represented as

$$\frac{(W_{n+1}^{\mathrm{k}+1} - W_{n+1}^{\mathrm{k}})}{\Delta t} = \frac{PK_y\Delta tD_w}{2fM_w}[(\frac{Y_{a_{n+1}}^{\mathrm{k}+1}}{Y_{a_{max}}} + \frac{Y_{a_n}^{\mathrm{k}}}{Y_{a_{max}}}) - (\frac{W_{n+1}^{\mathrm{k}+1}}{W_{max}} + \frac{W_n^{\mathrm{k}+1}}{W_{max}})]. \tag{4.18}$$

The governing equations discussed above were subject to boundary and initial conditions during the solution process. As outlined in Section 4.1.5, the absorbing rotary wheel total enthalpy exchanger solution method consisted of two main components: analysis of the wheel rotation through the supply air flow and analysis of the rotation through the exhaust air flow. Consequently, two sets of boundary conditions were required. This is illustrated in Fig. 4.3. To allow for different, independent air conditions of the supply and exhaust air streams, different supply and exhaust boundary conditions must be imposed. For most realistic applications the exiting supply and inlet exhaust conditions would be dependent in some manner. However, since such dependence is highly application specific, independent conditions were used for the exhaust boundary.

When the passage of supply air over the surface of the desiccant channel is modeled, the solution satisfies the supply air boundary conditions

$$T_a(0,t) = T_{abc_{pro}} \tag{4.19}$$

and

$$Y_a(0,t) = Y_{abc_{pro}}. \tag{4.20}$$

When the passage of exhaust air over the surface of the desiccant channel is modeled, the solution satisfies the exhaust air boundary conditions

$$T_a(X,t) = T_{abc_{reg}} \tag{4.21}$$

and

$$Y_a(X,t) = Y_{abc_{reg}}. \tag{4.22}$$

The initial conditions applied to the governing equations, regardless of which air flow is being analyzed first, are

$$W(x,0) = W_{ic} \tag{4.23}$$

and

$$T_w(x,0) = T_{w_{ic}}. \tag{4.24}$$

4.1.4 Adsorbing Rotary Wheel System of Equations

The adsorbing rotary wheel total enthalpy exchanger employed polyvinyl alcohol foam as the operating desiccant. This desiccant coating is an open-cell absorbing and adsorbing material. Combinatory absorbing and adsorbing substances exhibit both surface sorption characteristics, associated with absorption, as well as surface adhesion characteristics associated with adsorption. As moisture comes into contact with the polyvinyl alcohol foam surface, it disperses into the foam's numerous pores, or open cells, and adheres to both the inner walls of the pores and to the top surface of the desiccant material. The increased surface area contributed by the structural qualities of the polyvinyl alcohol foam allow for increased absorption; after the moisture has been brought into contact with the desiccant surface, it is absorbed in a manner similar to that reported in the previous subsection. As is the case with the absorbing polyvinyl alcohol film, once the moisture is within the material it becomes part of the homogeneous desiccant coating; this occasionally results in material swelling. In the foam-coated rotary wheel total enthalpy exchanger model, sorption and desorption are determined via a comparison of moisture *contents* between the air stream and the pore air within desiccant bed. This moisture content is defined as a ratio of actual moisture content in the air to the mass of dry air in the channel. Moisture within the actual desiccant material is evaluated by assuming that the normalized moisture *capacity* of the air within the desiccant pores equals the normalized moisture *capacity* of the desiccant material.

As in the study of the film-coated rotary wheel total enthalpy exchanger model, the governing differential equations representing the physics associated with the transfer phenomena within the adsorbing rotary wheel model were derived from the control volume analysis. Five equations were used for the absorbing rotary wheel total enthalpy exchanger numerical analysis: conservation of energy within the air stream (Eq. 4.25), conservation of energy within the desiccant bed (Eq. 4.27), conservation of mass within the air stream (Eq. 4.29), conservation of mass within the desiccant bed (Eq. 4.31), and finally an equilibrium expression expressing the relationship between the moisture content of the pore air and the moisture content and temperature of the desiccant material (Eq. 4.33). These equations, presented below, represent the fundamental theories employed throughout the completion of the control volume analysis used to model for the performance of polyvinyl alcohol foam as a desiccant coating for a rotary wheel total enthalpy exchanger. Corresponding finite-difference equations are presented simultaneously in this section.

In the same manner as Eq. 4.11 describes the effects of convective terms on the air temperature of the polyvinyl alcohol film-coated rotary wheel total enthalpy exchanger, the following equation describes the effects of convective terms on the air temperature of the polyvinyl alcohol foam-coated rotary wheel total enthalpy exchanger. Energy is again transferred between the air and desiccant material by way of convective heat and mass transfer. The first term on the right-hand-side represents the energy associated with convective mass transfer; the enthalpy associated with the transfer of water vapor is regulated by the mass transfer coefficient, K_y. The convective phenomena are divided into two terms. In this case the moisture gradient is given in terms of a moisture content difference rather than the moisture capacity of the absorbing case. The imbalance between the moisture content of the pore air and the free stream air, coupled with K_y create a convective mass transfer effect. The second term on the right-hand-side is identical in form and purpose to the convective heat transfer term in Eq. 4.11. These two terms combine to represent the way in the enthalpy of the air stream varies along the length of the desiccant channel:

$$\frac{\partial H_a}{\partial x} = \frac{P K_y H_v}{\rho V_a A_c}(Y_w - Y_a) + \frac{Ph}{\rho V_a A_c}(T_w - T_a). \tag{4.25}$$

The finite difference form of Eq. 4.25 is

$$\begin{aligned}\frac{(C_{da} + Y_{a_n}^{\mathrm{k+1}} C_v)}{\Delta x}(T_{a_{n+1}}^{\mathrm{k+1}} - T_{a_n}^{\mathrm{k+1}}) = \\ \frac{P K_y H_v \Delta x}{2\rho V_a A_c}[(Y_{w_{n+1}}^{\mathrm{k+1}} + Y_{w_n}^{\mathrm{k+1}}) - (Y_{a_{n+1}}^{\mathrm{k+1}} + Y_{a_n}^{\mathrm{k+1}})] + \\ \frac{Ph\Delta x}{2\rho V_a A_c}[(T_{w_{n+1}}^{\mathrm{k+1}} + T_{w_n}^{\mathrm{k+1}}) - (T_{a_{n+1}}^{\mathrm{k+1}} + T_{a_n}^{\mathrm{k+1}})].\end{aligned} \tag{4.26}$$

As does it's absorbing counterpart, the equation characterizing the conservation of energy within the desiccant bed relates the same thermal transfer from the perspective of the desiccant bed. Again, the desiccant bed conservation of energy equation relates the effects of convective terms on the desiccant bed temperature. Like Eq. 4.13, the next equation considers the amount and type of substrate upon which the desiccant is mounted. Consideration of all materials contributing to bed composition is necessary for determination of the effective specific heat term which is embedded in the expression describing the change in enthalpy with respect to time. As in the air stream conservation of energy equation, enthalpy change within the bed with respect to time is the result of the combination of a convective mass transfer term, the first term on the right-hand-side, and a convective heat transfer term, the second term on the right-hand-side. As pointed out above, the adsorbing equations express the mass transfer in term of a difference in moisture content between the resident bed air and the free stream air. This yields

$$\frac{\partial H_w}{\partial t} = \frac{2 K_y H_v}{a_w \rho_w}(Y_a - Y_w) + \frac{2h}{a_w \rho_w}(T_a - T_w). \tag{4.27}$$

The finite difference version of the conservation of energy within the desiccant bed is expressed as:

$$\frac{(1-f)C_s + f(C_d + C_v W_n^{\mathrm{k+1}})}{\Delta t}(T_{w_{n+1}}^{\mathrm{k+1}} - T_{w_{n+1}}^{\mathrm{k}}) = \frac{PK_y H_v D_w}{M_w}[(Y_{a_{n+1}}^{\mathrm{k+1}} + Y_{a_{n+1}}^{\mathrm{k}}) - (Y_{w_{n+1}}^{\mathrm{k+1}} + Y_{w_{n+1}}^{\mathrm{k}})] + \frac{PhD_w}{M_w}[(T_{a_{n+1}}^{\mathrm{k+1}} + T_{a_{n+1}}^{\mathrm{k}}) - (T_{w_{n+1}}^{\mathrm{k+1}} + T_{w_{n+1}}^{\mathrm{k}})]. \quad (4.28)$$

Similar to the absorbing model, the following equation covers the behavior of the air with respect to moisture content as it is transferred convectively to and from the desiccant material surface. In this case, however, the conservation of mass within the air stream is upheld by evaluating the differences in moisture content between the air in the desiccant bed, and the air stream, $(Y_w - Y_a)$. Regulated by the mass transfer coefficient K_y, the moisture content difference expressed as the right-hand-side determines the change in moisture content Y_a with respect to distance along the desiccant channel.

$$\frac{\partial Y_a}{\partial x} = \frac{PK_y}{\rho V_a A_c}(Y_w - Y_a) \quad (4.29)$$

The corresponding finite difference equation for the above conservation of mass expression is represented as

$$\frac{(Y_{a_{n+1}}^{\mathrm{k+1}} - Y_{a_n}^{\mathrm{k+1}})}{\Delta x} = \frac{PK_y}{2\rho V_a A_c}[(Y_{w_{n+1}}^{\mathrm{k+1}} + Y_{wn}{}^{\mathrm{k+1}}) - (Y_{a_{n+1}}^{\mathrm{k+1}} + Y_{a_n}^{\mathrm{k+1}})]. \quad (4.30)$$

The conservation of mass expression for the polyvinyl alcohol foam-coated wheel relates the manner in which the desiccant absorbs and desorbs water. In other words, this expression represents the change in moisture content of the desiccant bed with respect to time. The difference between this adsorbing conservation of mass equation and the corresponding absorbing conservation of mass equation lies in the use of the moisture content differential instead of a moisture capacity differential. Nevertheless, the effect of this difference is still controlled by the convective mass transfer coefficient K_y, and it describes the manner in which the moisture content of the desiccant bed changes with respect to time.

$$\frac{\partial W}{\partial t} = \frac{2K_y}{a_w f \rho_d}(Y_a - Y_w) \quad (4.31)$$

In addition, the finite difference representation of Eq. 4.31 is

$$\frac{(W_{n+1}^{\mathrm{k+1}} - W_{n+1}^{\mathrm{k}})}{\Delta t} = \frac{PK_y \Delta t D_w}{2fM_w}[(Y_{a_{n+1}}^{\mathrm{k+1}} + Y_{a_n}^{\mathrm{k}}) - (Y_{w_{n+1}}^{\mathrm{k+1}} + Y_{w_n}^{\mathrm{k+1}})]. \quad (4.32)$$

Lastly, the equation of equilibrium accounts for the condition within the desiccant bed mentioned previously; an equal relationship between the moisture capacity of the air in the material pores and the moisture content within the desiccant material itself:

$$Y_w = f(W, T_w). \tag{4.33}$$

Correspondingly, the finite difference expression of Eq. 4.33 is

$$Y^{\text{k+1}}_{w_{n+1}} = W^{\text{k+1}}_{n+1}(\frac{Y^{\text{k+1}}_{wmax_{n+1}}}{W_{max}}). \tag{4.34}$$

As in Section 4.1.3, the governing equations discussed above were subject to boundary and initial conditions during the solution process. As outlined in Section 4.1.5, the adsorbing rotary wheel total enthalpy exchanger solution method was identical to that of the absorbing rotary wheel total enthalpy exchanger. As a result, the same boundary and initial conditions implemented in the absorbing rotary wheel total enthalpy exchanger model, Eqs. 4.19 - 4.24 , were used in the adsorbing rotary wheel total enthalpy exchanger solution.

4.1.5 Solution Method

Although they modeled different mass transfer techniques, both rotary wheel total enthalpy exchanger programs implemented the same logic in their solution methods. Integrating with respect to time as exiting channel conditions were calculated within incremental time steps simultaneously accounted for an integration across the rotary wheel face. This was the fundamental notion that drove each rotary wheel total enthalpy exchanger program. A flowchart, depicting logic applicable to both programs, demonstrates this point in Fig. 4.4. An itemized description of solution steps outlining the step sequence of the rotary wheel total enthalpy exchanger flowchart is presented below:

1. Stepping along the length of the wheel channel (x direction), the air temperature and moisture content are computed incrementally for the supply process, along with the temperature and moisture content of the bed surface during the supply rotation. This is illustrated in part a) of Fig. 4.5.

2. Half-way through the supply rotation, air and bed conditions are recorded and stored.

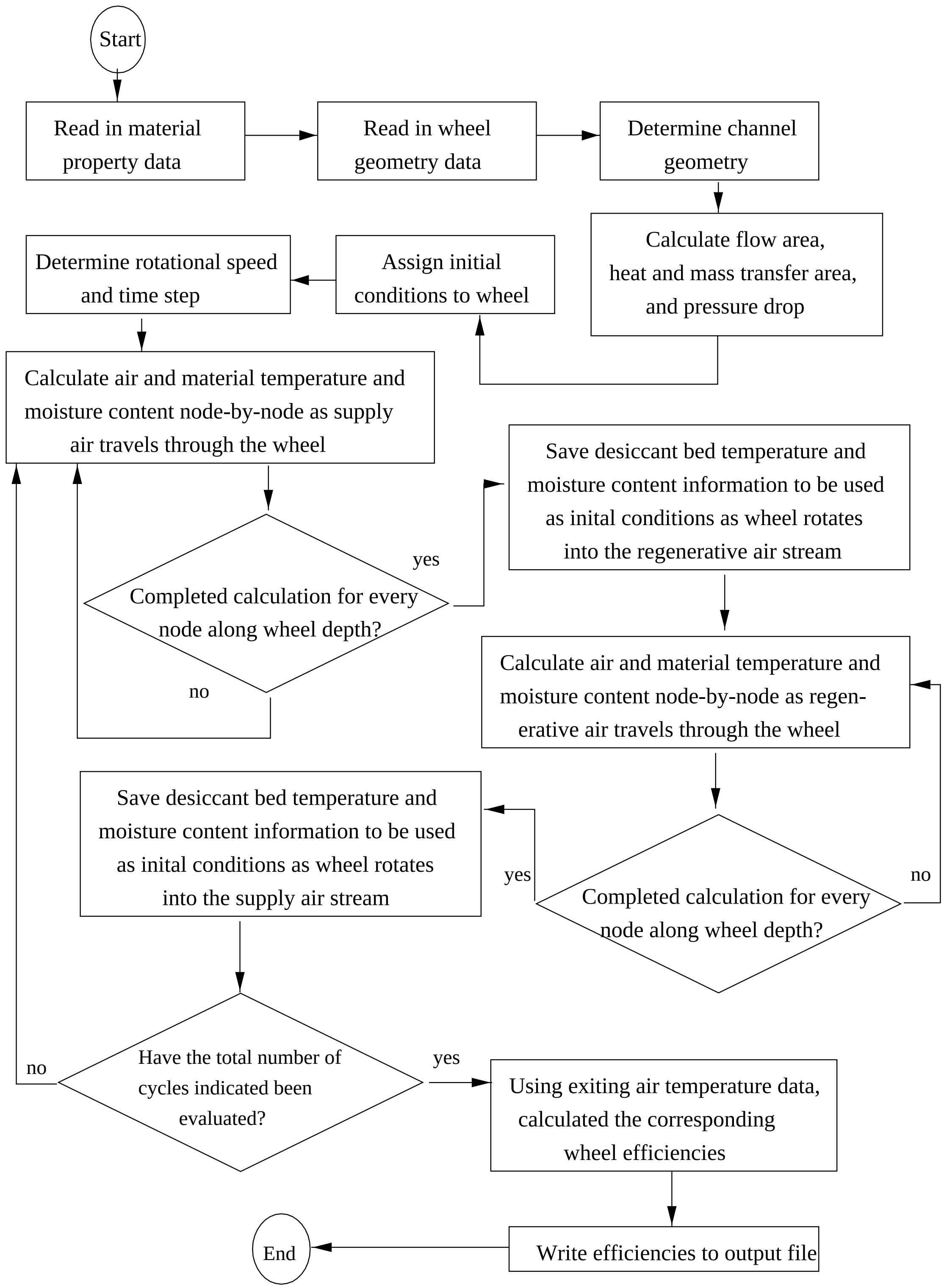

Figure 4.4 Flowchart Outlining Rotary Wheel Efficiency Estimation Program

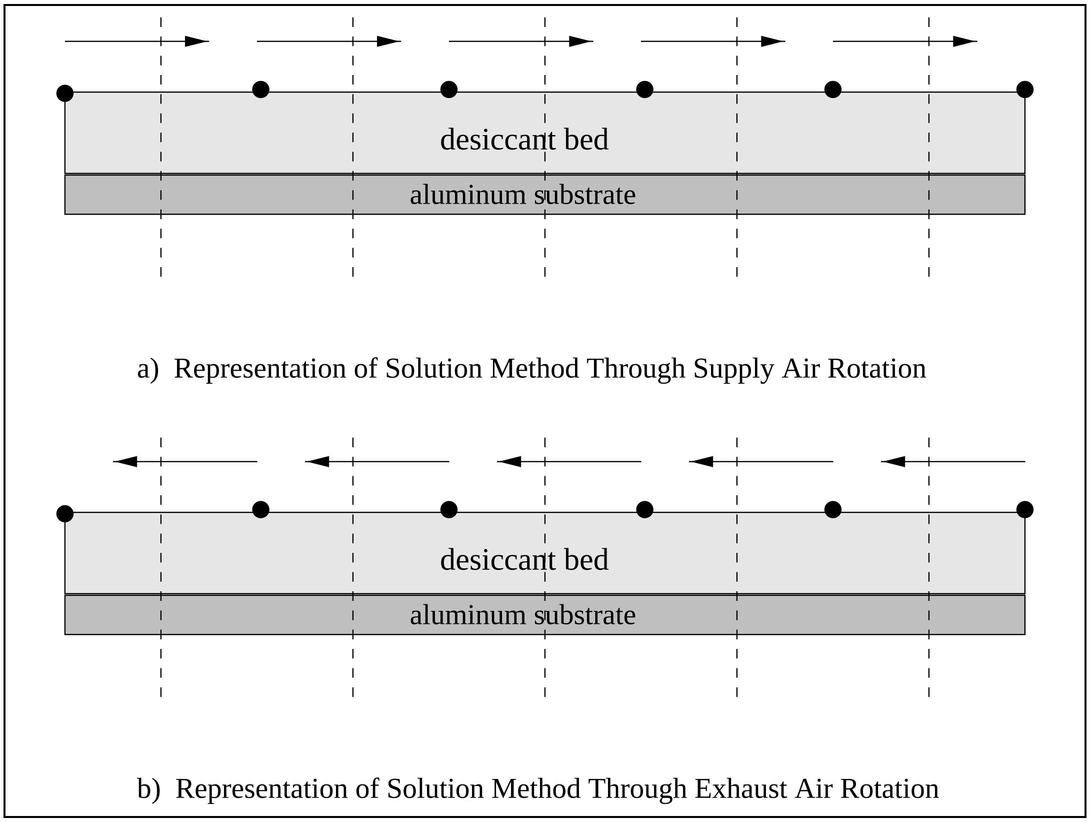

Figure 4.5 Method of Solution: Rotary Wheel Exchanger

These conditions will be used as 'previous' conditions when the exchanger rotates to the exhaust side and the regenerative process is begun.

3. Next, the regenerative process is analyzed via the same method outlined in step 1. In this step the program solves for the regenerative air temperature and moisture content as well as the bed temperature and moisture content during the exhaust rotation. Since the exchanger modeled featured counter-current air flow, this step was performed while marching in the negative x direction. This is illustrated in part (b) of Fig. 4.5.

4. Half-way through the exhaust rotation, air and bed conditions are recorded and stored. These conditions will be used as 'previous' conditions when the exchanger rotates to the exhaust side and the supply process is begun.

5. Once the entire supply and regenerative processes are evaluated once, the bed and air stream conditions are both updated and the program continues to the next time step.

6. Steps 1-5 are repeated for each time step until the analysis is complete.

7. The final exiting air conditions are recorded and used to evaluated the exchanger's efficiency.

4.2 Mathematical Model of Two Dimensional Fixed Plate

Since polyvinyl alcohol foam was the only polymer material in the study that was structurally robust enough to be used alone as a plate without a supporting structure, this analysis focused solely on the implementation of polyvinyl alcohol open-cell foam as the operating desiccant material of the fixed plate total enthalpy exchanger. The polyvinyl alcohol open-cell foam modeled in this section was the same material used in the adsorbing rotary wheel total enthalpy exchanger model. Thus, all properties and characteristics associated with polyvinyl alcohol foam revealed in Section 4.1.4 were applied to the polyvinyl alcohol foam fixed plate total enthalpy exchanger performance model.

Since the polyvinyl alcohol foam was now being used in a new application, new assumptions had to be developed for the two-dimensional fixed plate total enthalpy exchanger mathematical model; these assumptions are:

- Considering the air passage spacing typical in existing fixed plate total enthalpy exchanger products, it was assumed that no substantial gradient would exist within the height of the air stream.

- Considering the transfer taking place within the thickness of the polyvinyl alcohol desiccant foam, it was assumed that gradients would exist within the thickness of the material. These gradients were assumed to be linear.

- It was assumed that the transfer of heat and mass axially by conduction and diffusion along the length of the dehumidifier, both in the air and in the material, were negligible.

- The heat and mass transfer coefficients of the fixed plates were constant throughout the apparatus.
- All air passages were assumed to be symmetric.

These assumptions were incorporated into the code used to estimate performance efficiencies of various fixed plate exchangers. By varying parameters including material properties, plate size and spacing, different fixed plate total enthalpy exchanger configurations could be analyzed.

4.2.1 Two Dimensional Equation Development

Although the performance estimates for the fixed plate total enthalpy exchanger were ultimately solved using a numerical scheme, both analytical and numerical equations were initially developed. The governing differential equations for the two-dimensional fixed plate total enthalpy exchanger model were derived using a control volume technique. Simultaneously, using the theories of conservation of energy and conservation of mass, finite difference equations were derived.

Using the same method as described in Section 4.1, differential elements with length dx were chosen and analyzed with respect to the principles of conservation of energy and conservation of mass. For this analysis two control volumes were used: a control volume for the air stream/material surface interface and a control volume actually within the desiccant material. A schematic depicting both control volume analyses of the conservation of energy is presented in Fig. 4.6. While considering the assumptions outlined in Section 4.2.2, examining an incremental distance of polyvinyl alcohol desiccant foam in addition to examining an incremental distance of air stream/material interface within an incremental portion of time allows for the development of difference equations as well as differential equations presented later in this chapter.

The finite difference techniques used in the solution of the equations of the rotary wheel total enthalpy exchanger models were also applied to the fixed plate total enthalpy exchanger equations. Here they were used to evaluate the plate surface transfer phenomena. With the addition of diffusion through the foam, supplementary finite difference equations were also used in the fixed plate total enthalpy exchanger model. The inclusion of heat and mass transfer through the desiccant plate required supplemental diffusion equations.

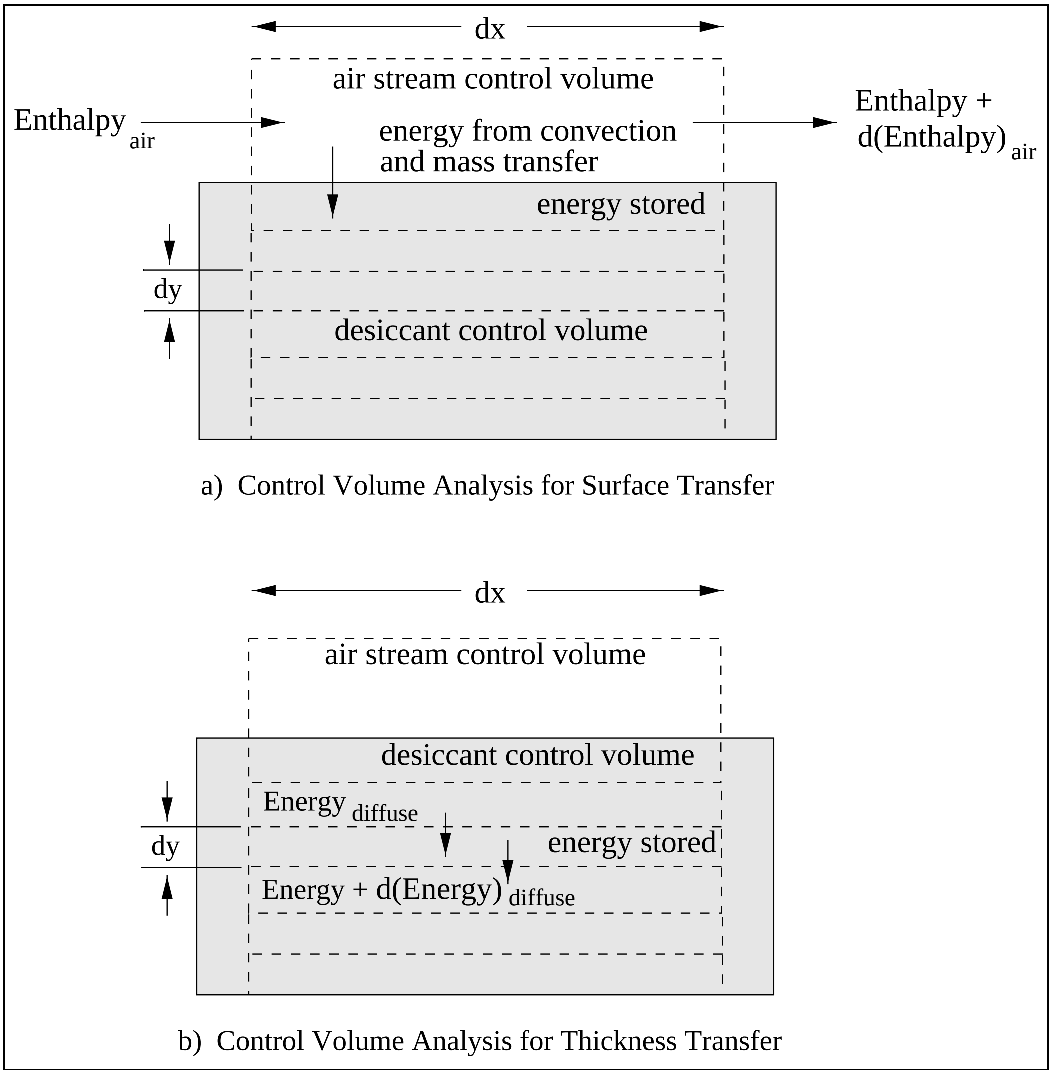

Figure 4.6 Control Volume Schematic Representing Conservation of Energy for Fixed Plate Scenario

Using the hypothetical term A, the finite differencing of the second-order terms in these relationships is related to the differential expression in space as

$$\frac{\partial^2 A}{\partial x^2} \quad \cong \quad \frac{(A_{n+1}^{\mathrm{k+1}} - 2A_n^{\mathrm{k+1}} + A_{n-1}^{\mathrm{k+1}})}{(\Delta x)^2}. \tag{4.35}$$

By incorporating this term with those discussed previously in Section 4.1.1, finite difference equations for the two-dimensional fixed plate total enthalpy exchanger model were derived. These equations, along with their corresponding differential equations are outlined in the next section.

4.2.2 Counter-Flow Plate Exchanger System of Equations

The use of polyvinyl alcohol foam in a non-regenerative situation like the fixed plate total enthalpy exchanger requires the addition of another dimension to the model. The lumped capacitance assumption used with the regenerative rotary wheel total enthalpy exchanger is no longer valid; the polyvinyl alcohol foam must now disperse the heat and moisture using material attributes like mass and thermal diffusion. The material is required to transport both heat and mass to areas of lower heat and mass concentration; i.e. cooler, drier areas. Specifically, this displacement needs to be carried out without the transfer of air-borne toxins; a popular feature of the fixed plate application is its ability to transfer moisture exclusively. This is accomplished in part with the implementation of a selectively diffusive material; in other words, a material that is capable of transferring moisture and heat through its thickness without allowing the passage of air through its thickness.

As in the study of the foam-coated rotary wheel total enthalpy exchanger model, the governing differential equations representing the physics associated with the transfer phenomena within the adsorbing fixed plate model were also derived from the control volume analyses. A parallel development of both the governing differential equations and the numerical finite difference equations is presented below. Seven equations were used for the adsorbing fixed plate total enthalpy exchanger numerical analysis of the plate surface: conservation of energy within the air stream (Eq. 4.25), conservation of energy within the desiccant bed (Eqs. 4.27 and 4.36), conservation of mass within the air stream (Eq. 4.29), conservation of mass within the desiccant bed (Eqs. 4.31 and 4.38), and finally an equilibrium expression enforcing the relationship between the moisture content of the pore air and the moisture content and temperature of the desiccant material (Eq. 4.33). These equations represented the fundamental theories employed throughout the completion of the control volume analysis used to model the performance of polyvinyl alcohol foam as a desiccant plate for a fixed plate total enthalpy exchanger.

Since the same polyvinyl alcohol foam material used in the adsorbing rotary wheel total enthalpy exchanger model was being used for the fixed plate total enthalpy exchanger, it was possible to use conservation equations Eq. 4.25, Eq. 4.27, Eq. 4.29, and Eq. 4.31, as well as the equilibrium equation, Eq. 4.33, to represent the convective heat and mass transfer occurring on both face surfaces of the plate. These equations were used as surface boundary equations for the solution of thermal and mass diffusion through the thickness of the plate. The three equations derived below were used to model the transfer of heat and moisture through the thickness of the polyvinyl alcohol foam plate. Fick's law was used to define the moisture transfer through the bed thickness. Fourier's Law was augmented to account for the thermal energy associated with the moisture transfer as well as the thermal behavior introduced by conduction. The equilibrium equation Eq. 4.33 was also used in this portion of the analysis to determine the amount of moisture present within the air of the desiccant pores, Y_w.

The relationship shown below indicates that the heat transfer from one air stream to another through the polyvinyl alcohol foam thickness is driven by both conduction effects and mass transfer effects. The first term on the right-hand-side represents the thermal contributions of conduction while the second term on the right-hand-side depicts the thermal effects corresponding to the mass flow of water through the plate thickness; thus

$$\frac{\partial T_w}{\partial t} = \alpha \frac{\partial^2 T_w}{\partial y^2} + V_M \frac{\partial T_w}{\partial y}. \tag{4.36}$$

The finite difference version of thermal diffusion within the desiccant bed is

$$\frac{(T^{\mathrm{k+1}}_{w_{n+1,m}} - T^{\mathrm{k}}_{w_{n+1,m}})}{\Delta t} = \frac{\alpha}{\Delta y^2}(T^{\mathrm{k+1}}_{w_{n+1,m+1}} - 2T^{\mathrm{k+1}}_{w_{n+1,m}} + T^{\mathrm{k+1}}_{w_{n+1,m-1}}) + \frac{V_M}{2\Delta y}(T^{\mathrm{k+1}}_{w_{n+1,m-1}} + T^{\mathrm{k+1}}_{w_{n+1,m+1}}). \tag{4.37}$$

The mass transfer is expressed as a second order Fickian phenomenon. Fundamentally, diffusion is based on Fick's first law which describes the macroscopic transport of molecules by a concentration gradient. The mass transfer, controlled by the mass diffusion coefficient, D_{ab}, is the only contributing factor to the determination of the amount of moisture present throughout the thickness of the polyvinyl alcohol desiccant plate as:

$$\frac{\partial W}{\partial t} = D_{ab} \frac{\partial^2 W}{\partial y^2}. \tag{4.38}$$

The corresponding finite difference form of the above diffusive equation is represented as

$$\frac{(W^{\mathrm{k+1}}_{n+1,m} - W^{\mathrm{k}}_{n+1,m})}{\Delta t} = \frac{D_{ab}}{\Delta y^2}(W^{\mathrm{k+1}}_{n+1,m+1} - 2W^{\mathrm{k+1}}_{n+1,m} + W^{\mathrm{k+1}}_{n+1,m-1}). \tag{4.39}$$

The governing equations discussed above were subjected to boundary and initial conditions during the solution process. As outlined in the Section 4.2.3, the counter-flow fixed plate total enthalpy exchanger solution method consisted of two main components:

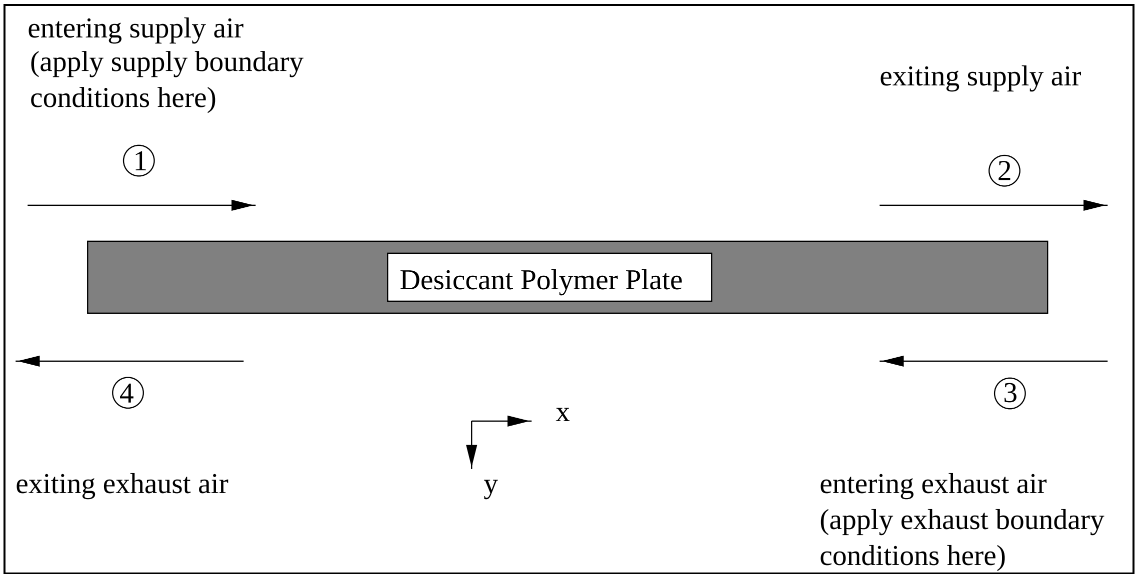

Figure 4.7 Representative Flat Plate Total Enthalpy Exchanger Environment

analysis of the supply air flow and analysis of the exhaust air flow. Consequently, two sets of boundary conditions were required. This is illustrated in Fig. 4.7. To allow for different, independent air conditions of the supply and exhaust air streams, different supply and exhaust boundary conditions must be observed. For most realistic applications the exiting supply and inlet exhaust conditions would be dependent in some manner. However, since such dependence is highly application specific, independent conditions were used for the exhaust boundary.

When the passage of supply air over the surface of the desiccant plate is modeled, the solution imposes the supply air boundary conditions as follows:

$$T_a(0, 1, t) = T_{abc_{pro}}, \tag{4.40}$$

$$Y_a(0, 1, t) = Y_{abc_{pro}}. \tag{4.41}$$

When the passage of exhaust air over the surface of the desiccant plate is modeled, the solution imposes the exhaust air boundary conditions:

$$T_a(X, Y, t) = T_{abc_{reg}}, \tag{4.42}$$

$$Y_a(X, Y, t) = Y_{abc_{reg}}. \tag{4.43}$$

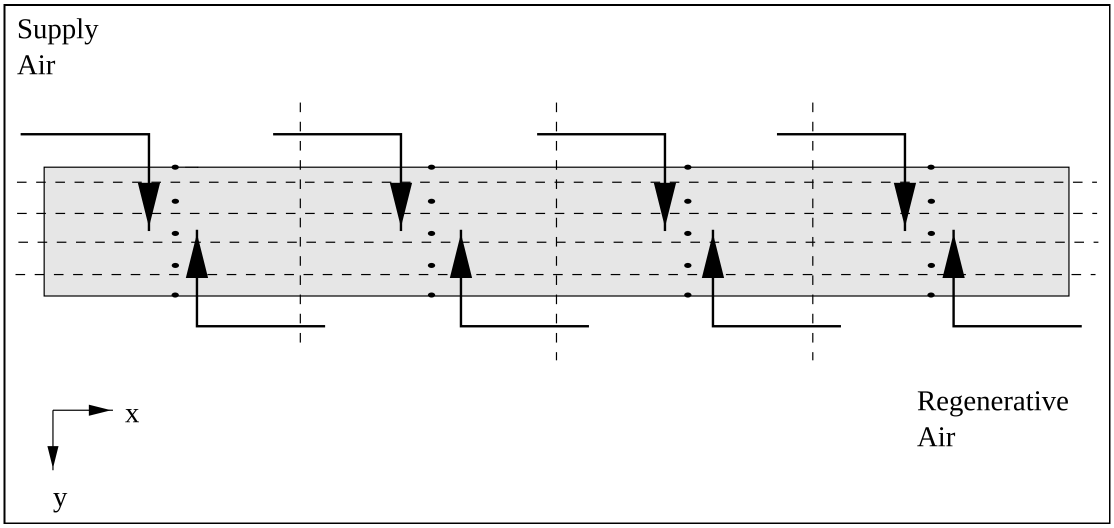

Figure 4.8 Method of Solution: Counter Flow Plate Exchanger

The initial conditions, regardless of which air passage is being analyzed first, at the onset of the solution process are

$$W(x, y, 0) = W_{ic} \tag{4.44}$$

and

$$T_w(x, y, 0) = T_{w_{ic}}. \tag{4.45}$$

4.2.3 Solution Method

The addition of another dimension to the fixed plate total enthalpy exchanger model resulted in the need for a more complicated solution method than the method used to solve both one-dimensional performance estimate models. In order to illustrate this solution method for the fixed plate total enthalpy exchanger model equations, Fig. 4.8 is included. Also, an itemized description of solution steps outlining the step sequence of the counter-flow fixed plate total enthalpy exchanger flowchart, Fig. 4.9, is presented.

1. Stepping along the length of the plate (x direction), the air temperature and moisture content are calculated incrementally for the supply process, along with the temperature and moisture content of the supply bed surface.

2. Before continuing to another node along the plate (x direction), the temperature and

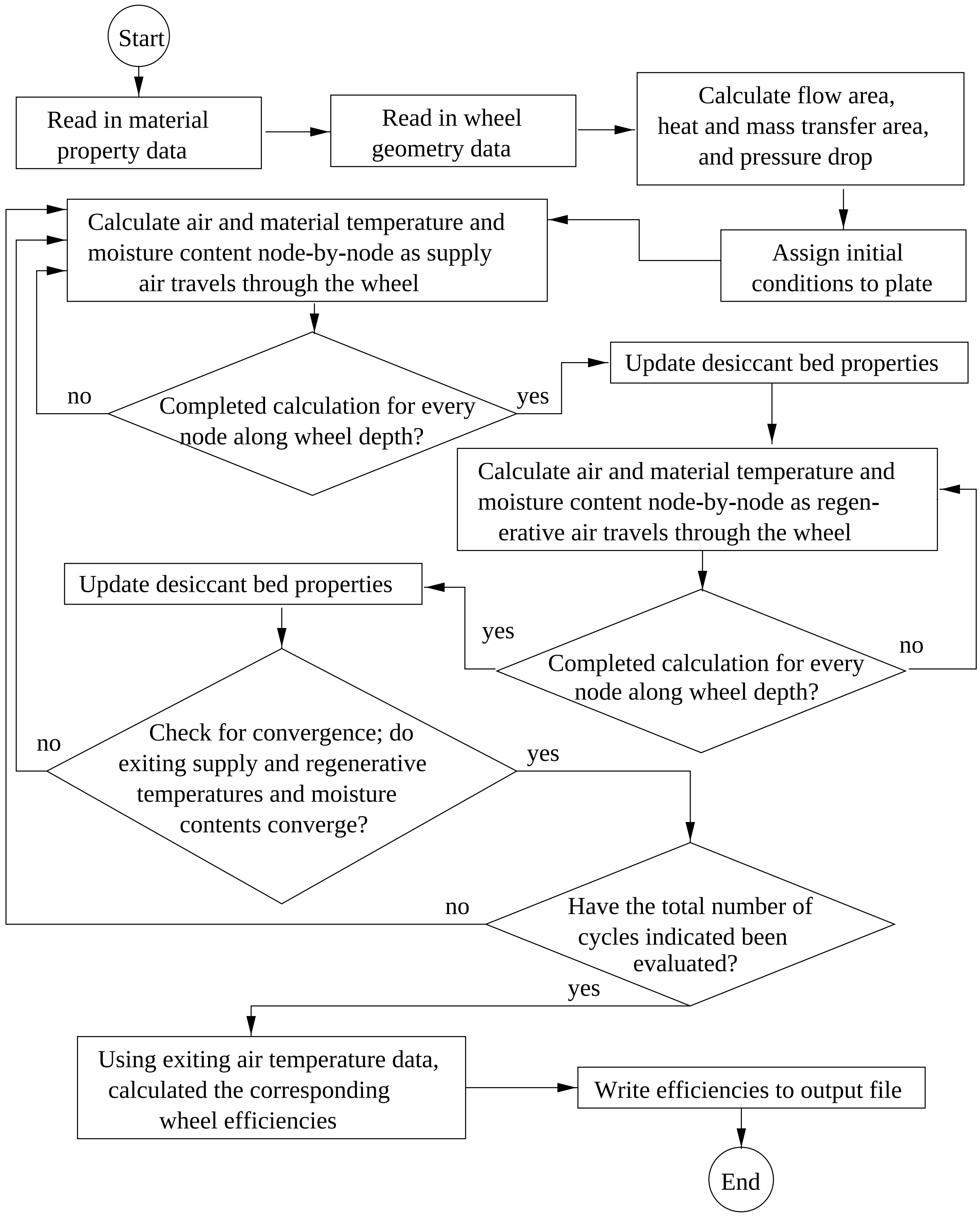

Figure 4.9 Flowchart outlining Counterflow Plate efficiency estimation program

moisture content of the bed are determined incrementally by stepping through the thickness of the desiccant plate (y direction).

3. Next, the regenerative process is analyzed via the same method outlined in step 1. This step solves for the regenerative air temperature and moisture content as well as the bed temperature and moisture content on the regenerative surface. Since the exchanger modeled was a counter-flow exchanger, this step was performed while marching in the negative x direction.

4. As in step 2, before continuing along the plate, the temperature and moisture content of the bed are determined incrementally by stepping through the thickness of the desiccant plate (negative y direction).

5. Once the entire supply and regenerative processes are evaluated once, an iterative convergence check is performed on the exiting supply air conditions. Convergence is determined by comparing the change in exiting supply air conditions. If convergence is not met, the desiccant bed conditions are reset, the supply and exhaust air conditions are updated, and items 1-4 are repeated. Once convergence is met, bed and air stream conditions are both updated and the program continues to the next time step.

6. Steps 1-5 are repeated for each time step until the analysis is complete.

7. The final exiting air conditions are recorded and used to evaluated the exchanger's efficiency.

4.3 Evaluation of Mathematical Models

The analysis of each mathematical model consisted of three main components: the determination of steady-state operation, the computation of steady-state system efficiencies, and the determination of model uncertainty.

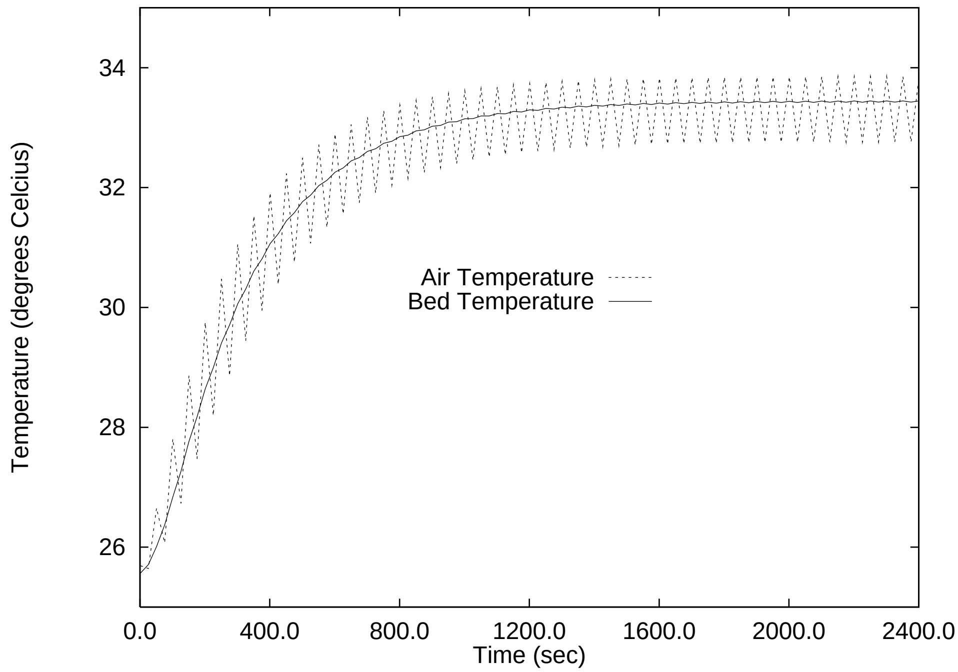

Figure 4.10 Transient Temperature Profiles: Rotary Wheel Total Enthalpy Exchanger

4.3.1 Steady-State Operation

The main function of the mathematical models developed from the finite difference equations discussed throughout this chapter involved evaluating the exit supply air quality using the exiting air temperature and humidity ratio. Both transient and steady-state air condition information was desired. Transient models were used to determine the amount of time required for the system to reach steady conditions. The performance of each system was based on the corresponding steady-state system efficiency. Consequently, the determination of whether a model was being evaluated long enough to achieve steady-state was of great importance.

For each model, steady-state conditions were determined by plotting the temperature and moisture content variables with respect to time. Examples of these plots are given in Figs. 4.10 and 4.11. Figure 4.10 represents the time response of both air and desiccant bed temperature associated with an interior node of a polyvinyl alcohol foam-coated rotary wheel. Similarly, Fig. 4.11 represents the time response of both air and desiccant bed temperature associated with the exiting supply node of a polyvinyl alcohol foam fixed plate exchanger. By visually assessing the approximate time it took for the model to achieve

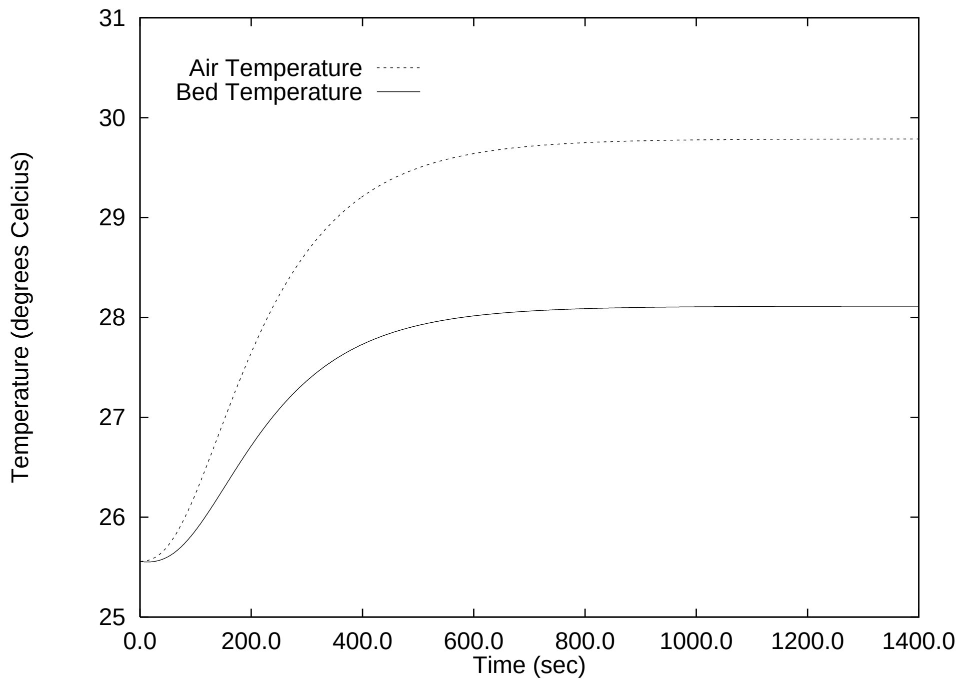

Figure 4.11 Transient Temperature Profiles: Fixed Plate Total Enthalpy Exchanger

steady state, it could be assured that the model was being evaluated for the appropriate time period.

4.3.2 Determination of Energy Efficiencies

In both the rotary wheel total enthalpy exchanger and fixed plate total enthalpy exchanger models, the exiting supply air temperatures and humidity ratios were used along with supply and exhaust air stream boundary conditions to determine exchanger efficiencies. Three efficiencies were used to evaluate an exchanger's overall steady-state performance: sensible efficiency, latent efficiency, and total efficiency.

Sensible Efficiency

The sensible efficiency of the system describes the exchanger's ability to remove heat from one air stream and transfer it to another. The sensible efficiencies for all numerical models evaluated in this project were determined by

$$\eta_s = \frac{(T_{a_1} - T_{a_2})}{(T_{a_1} - T_{a_3})}. \tag{4.46}$$

The sensible efficiency of a total enthalpy exchange system is represented as the difference in the inlet and exit temperatures of the supply air, temperatures at locations 1 and 2 in Fig. 4.3 and Fig. 4.7, as compared to the difference in inlet supply air and inlet exhaust air temperatures of locations 1 and 3 in Fig. 4.3 and Fig. 4.7.

Latent Efficiency

The latent efficiency of the system describes the exchanger's ability to remove moisture from one air stream and transfer it to another. The energy associated with the phase change between liquid and vapor states is called latent energy. The latent efficiencies for all numerical models evaluated in this project were determined from

$$\eta_l = \frac{(Y_{a_1} - Y_{a_2})}{(Y_{a_1} - Y_{a_3})}. \tag{4.47}$$

The latent efficiency of a total enthalpy exchange system is represented as the difference in inlet and exit humidity ratios of the supply air, temperatures at locations 1 and 2 in Fig. 4.3 and Fig. 4.7, as compared to the difference in inlet supply air and inlet exhaust air humidity ratios of locations 1 and 3 in Fig. 4.3 and Fig. 4.7.

Total Efficiency

The total efficiency of the system describes the exchanger's ability to remove total sensible and latent energy from one air stream and transfer it to another. The total efficiencies for all numerical models evaluated in this project were determined from

$$\eta_{tot} = \frac{Cda(T_{a_1} - T_{a_2}) + (Y_{a_1}h_{g_1} - Y_{a_2}h_{g_2})}{Cda(T_{a_1} - T_{a_3}) + (Y_{a_1}h_{g_1} - Y_{a_3}h_{g_3})}. \tag{4.48}$$

The total efficiency is represented as the difference in total enthalpy associated with the inlet and exit temperatures and humidity ratios of the supply air as compared to the total enthalpy associated with the inlet supply air and inlet exhaust air temperatures and humidity ratios. The thermal differences are converted into enthalpy changes by the multiplication of the temperature difference by the specific heat of air. The humidity ratios are multiplied by the enthalpy of vapor at the specified temperature to yield the enthalpic differences contributed by the humidity ratio differences.

4.3.3 Uncertainty Analysis

During the development of the exchanger models it was realized that added meaning could be attributed to the performance estimates obtained in Chapter 5 if the model outputs included estimates of error. Such estimates of error can be determined from uncertainties in assumed 'known' parameters within the model. In other words, final estimates of the efficiencies of the various exchangers should include error resulting from error in the material properties used to model the total enthalpy exchanger systems.

In both the rotary wheel total enthalpy exchanger and fixed plate total enthalpy exchanger models, thermal and structural properties of the different polyvinyl alcohol materials were set as constant, 'known' parameters. Variations in model output can be observed if changes are made to these 'known' parameters. Therefore, any possible uncertainty arising from the estimation of these properties must be included in the model in order to obtain the overall uncertainty associated with the performance estimate. Using the data gathered during the property estimation tests described in Chapter 3, rough uncertainties were determined for each material property.

The uncertainties of each known parameter, in conjunction with sensitivity coefficients, were used to determine an overall expected standard deviation of the model output. This is demonstrated by (Moffat, 1988)

$$\sigma_{i,\eta} = X_{i,\eta}\sigma_i. \tag{4.49}$$

The sensitivity coefficients, $X_{i,\eta}$, are an indication of how strongly the model output reacts to small changes to the 'known' parameters. By multiplying each sensitivity coefficient by the standard deviation, σ_i, of the corresponding parameter, the performance estimation errors resulting from each parameter were determined (Moffat, 1988). As shown in Eq. 4.49, the expected deviation of efficiency due to the uncertainty in the i^{th} parameter is represented as $\sigma_{i,\eta}$. After each error is calculated, a summation of the errors yields the total uncertainty within the model performance estimations:

$$\sigma_{total} = \sum_{i=1}^{n} X_{i,\eta}\sigma_i = \sum_{i=1}^{n} \sigma_{i,\eta}. \tag{4.50}$$

4.4 Implementation of Mathematical Models

Upon completion of the three mathematical models discussed in this chapter, the one-dimensional absorbing rotary wheel total enthalpy exchanger, the one-dimensional adsorbing rotary wheel total enthalpy exchanger, and the two-dimensional counter-flow fixed plate total enthalpy exchanger models, the finite difference equations were implemented in Fortran 77 codes. Copies of the absorbing and adsorbing one-dimensional codes are included in Appendices B and C, respectively, while a copy of the two-dimensional counter-flow code is found in Appendix D. The programs were compiled and executed on Dell® Dimension XPS Pro200n personal computer. Several test cases were studied in order to gain insight into each model's behavior. Results of these test cases are presented in Chapter 5.

Chapter 5

Results and Discussion

Upon theoretical development of the finite difference equations and the solution methods for both the rotary wheel total enthalpy exchanger and the fixed plate total enthalpy exchanger systems, Fortran codes were written to accomplish the numerical solution of the finite differencing schemes. These models were used to determine each systems' response to a variety of parameters. During each desiccant system study, parameters relating to the material properties and their effect on the sensible, latent, and total efficiencies of the system were observed. Within each material study, the response of these efficiencies to changes in system parameters such as the desiccant thickness, desiccant surface area, and operating air stream velocity were evaluated. When applicable, the efficiency response to substrate thickness and air channel height was also investigated.

5.1 One Dimensional Analysis of Rotary Wheel

As discussed in Chapter 4, the absorbing rotary wheel total enthalpy exchanger model was used to evaluate the thermal and mass transfer efficiency performance of cross-linked polyvinyl alcohol film. Polyvinyl alcohol film was the only absorbing material studied during the course of this research, and was the first desiccant to be modeled. Due to its performance in the initial uptake test examined in Chapter 3, polyvinyl alcohol was believed to be the most promising polymer desiccant coating.

After reviewing model runs predicting the performance of polyvinyl alcohol film-coated rotary wheel total enthalpy exchangers, however, it became evident that a material with more inherent surface area would provide better energy transfer. At this point,

Table 5.1 Default Parameters Used in One-Dimensional Absorbing Analysis of Polyvinyl Alcohol Film

Parameter	Default Value
Channel Depth	0.152 m
Desiccant Thickness	0.075 mm
Air Stream Velocity	2.0 m/s
Substrate Thickness	0.254 mm
Channel Height	1.52 mm

polyvinyl alcohol foam, chosen because of its increased surface area provided by the pores, was modeled as the next material coating for a rotary wheel. Due to application difficulties concerning the rotary wheel with the polyvinyl alcohol foam, a third polyvinyl alcohol material, polyvinyl alcohol/ceramic composite was studied. Finally, a material currently used in the rotary wheel total enthalpy exchanger market, LaRoche paper, was then brought in and evaluated for use in comparing the polyvinyl alcohol materials and their performance. Results of these desiccant material coatings and their performance with rotary wheel total enthalpy exchanger systems are discussed throughout this section.

5.1.1 Absorbing Rotary Wheel

Incorporating the polyvinyl alcohol film material properties presented in Chapter 3 into the absorbing one-dimensional model, sensible, latent, and total efficiencies of the polyvinyl alcohol film-coated rotary wheel total enthalpy exchanger were determined. Test cases were run to demonstrate the wheel's performance with respect to surface area, desiccant thickness, aluminum substrate thickness, and operating air stream velocity. The results of these runs are shown in Figs. 5.1 - 5.4.

As one parameter was varied, the other parameters were held constant. Default values for each parameter are given in Table 5.1. The first variable studied was the available surface area contributed by the desiccant coating and its on rotary wheel efficiency. This was accomplished by varying the desiccant channel depth, or the depth of the desiccant wheel. These results are illustrated in Fig. 5.1. Trends from this graph reinforce the idea that an increase in surface area produces an increase in sensible, latent, and consequently total efficiency.

Next, the effect of desiccant thickness on efficiency was evaluated. It is evident from

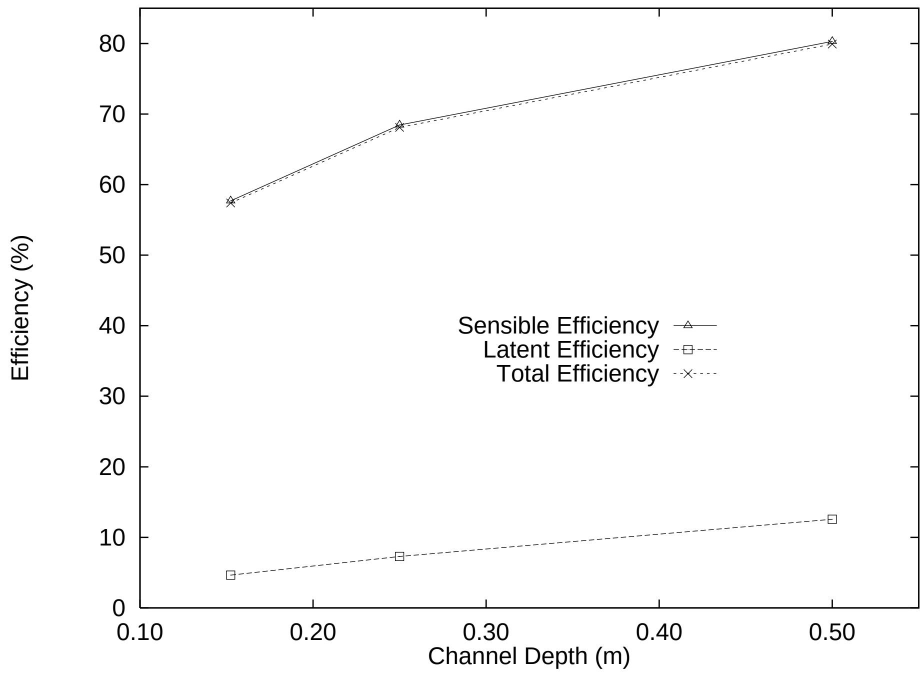

Figure 5.1 Polyvinyl Alcohol Film-Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Channel Depth

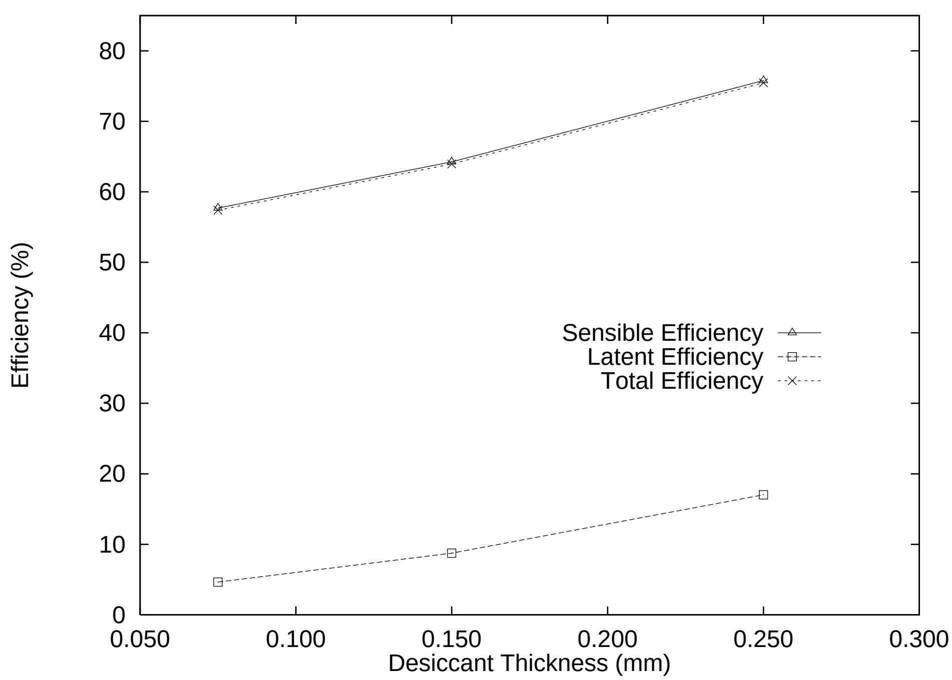

Figure 5.2 Polyvinyl Alcohol Film-Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Desiccant Thickness

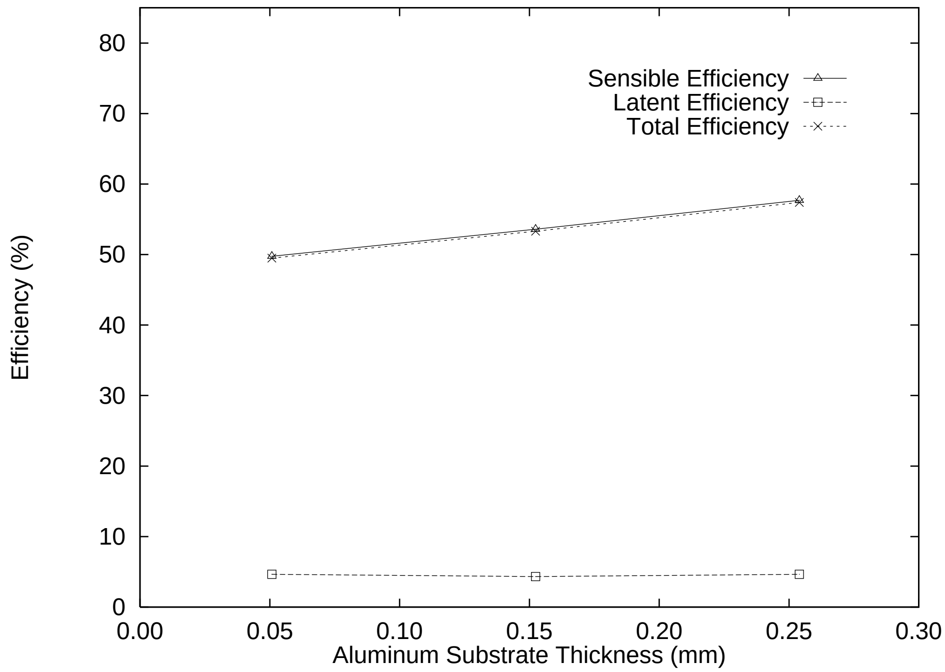

Figure 5.3 Polyvinyl Alcohol Film-Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Aluminum Substrate Thickness

Fig. 5.2, that as the amount of coating on the constant-thickness aluminum substrate is increased, an increase in all efficiencies is realized. The addition of desiccant resulted in a change in the effective properties of the dehumidifier. These effective properties were obtained via a volume-averaging technique. The test results indicate that the more contribution the desiccant properties make with respect to the effective dehumidifier properties, the greater the overall system efficiency.

This knowledge resulted in an effort to study the wheel performance changes as the substrate thickness was varied. The results of these runs are included in Fig. 5.3. It is shown that an increase in the aluminum content of the wheel helped boost the sensible (and thus the total) efficiency of the system. Since the desiccant-to-air flow ratio remained constant in this test, there was no noted change in the response of the latent efficiency.

Finally, the rotary wheel total enthalpy exchanger response to changing air stream velocities was determined. By increasing and decreasing the speed at which both the supply and exhaust air passed through the channels of the dehumidifier, changes were made to the performance of the wheel. The operating air speed reflects the amount of time the air is allowed to be conditioned within the rotary wheel. According to Fig. 5.4, the more resident time the air is given, the more heat and mass transfer is allowed to take place. Therefore,

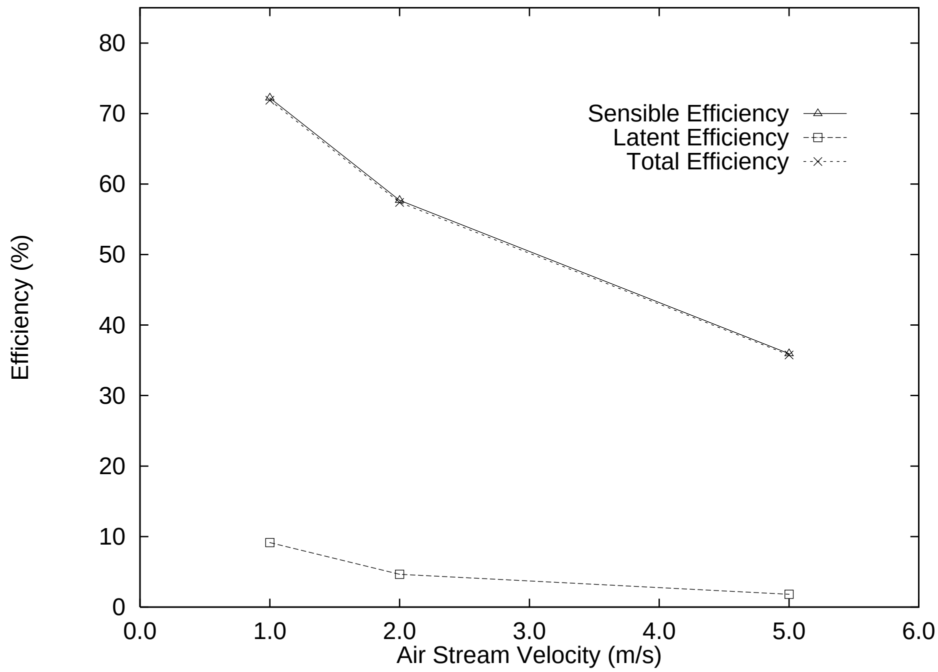

Figure 5.4 Polyvinyl Alcohol Film-Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Air Stream Velocity

it was observed that as the velocity of the air flows increases, sensible, latent, and total efficiencies declines.

The results of the polyvinyl alcohol film-coated model runs are summarized in Table 5.2. Overall, the program results produced expected trends. Although certain configurations produced acceptable sensible efficiency outputs, every test indicated a significantly low latent efficiency with the use of the polyvinyl alcohol film coating. In fact, the latent efficiency was consistently an order of magnitude lower than the sensible efficiency. This issue, along with the indication that increased surface area increased the wheel performance, led to the selection of a porous polyvinyl alcohol foam for the rotary wheel application. The results of this study are presented in the next section.

5.1.2 Adsorbing Rotary Wheel

The open-cell structure of the polyvinyl alcohol foam required a modeling scheme separate from that of the absorbing system; a new estimation program was developed. The primary differences between the film coated wheel model and the foam coated wheel model lay in the differences in the mass transfer coefficients and the absence/presence of air within the

Table 5.2 Polyvinyl Alcohol Film-Coated Rotary Wheel Total Enthalpy Exchanger Model Summary

Parameter	Change in Parameter	Change in Sensible Eff.	Change in Latent Eff.	Change in Total Eff.
Channel Depth	.348 m	39.1 %	170.7 %	39.3 %
Film Thickness	0.175 mm	31.3 %	266.8 %	31.5 %
Substrate Thickness	.254 mm	13.8 %	0.2 %	13.7 %
Air Stream Velocity	4.0 m/s	100.8%	402.2 %	101.1%

absorbing/adsorbing material. Desiccant materials other than the polyvinyl alcohol foam were also modeled with this program. After the polyvinyl alcohol foam desiccant coating was tested, a polyvinyl alcohol/ceramic composite coating, as well as the LaRoche paper coating were modeled. The results of these tests are presented later in this section.

Polyvinyl Alcohol Foam

Sensible, latent, and total efficiencies of the polyvinyl alcohol foam-coated rotary wheel total enthalpy exchanger were determined by incorporating the polyvinyl alcohol foam material properties presented in Chapter 3 into the adsorbing one-dimensional model. As in the study of the polyvinyl alcohol film-coated wheel, test cases were run to demonstrate the wheel's performance with respect surface area, desiccant thickness, operating air stream velocity, and aluminum substrate thickness. The results of these runs are shown in Figs. 5.5 - 5.8.

The same method outlined in Section 5.1.1 was used for the each adsorbing model; as one parameter was varied, the other parameters were held constant. Default values for each parameter are given in Table 5.3. Again, the first variable studied was the available surface area contributed by the desiccant coating and its effect on rotary wheel efficiency. This was accomplished by varying the depth of the desiccant wheel. These results are illustrated in Fig. 5.5. Trends from the test results re-affirm the concept introduced earlier regarding a coupling between an increase in surface area and an increase in sensible, latent, and total efficiency. As suspected, Fig. 5.5 also shows that the use of an open-cell medium increases the latent efficiency.

The efficiency response to changes in desiccant thickness was evaluated next. As illustrated with the polyvinyl alcohol film, Fig. 5.6 indicates that as the coating on the

Table 5.3 Default Parameters Used in One-Dimensional Adsorbing Analyses

Parameter	Default Value
Channel Depth	0.152 m
Desiccant Thickness	0.075 mm
Air Stream Velocity	2.0 m/s
Substrate Thickness	0.254 mm
Channel Height	1.52 mm

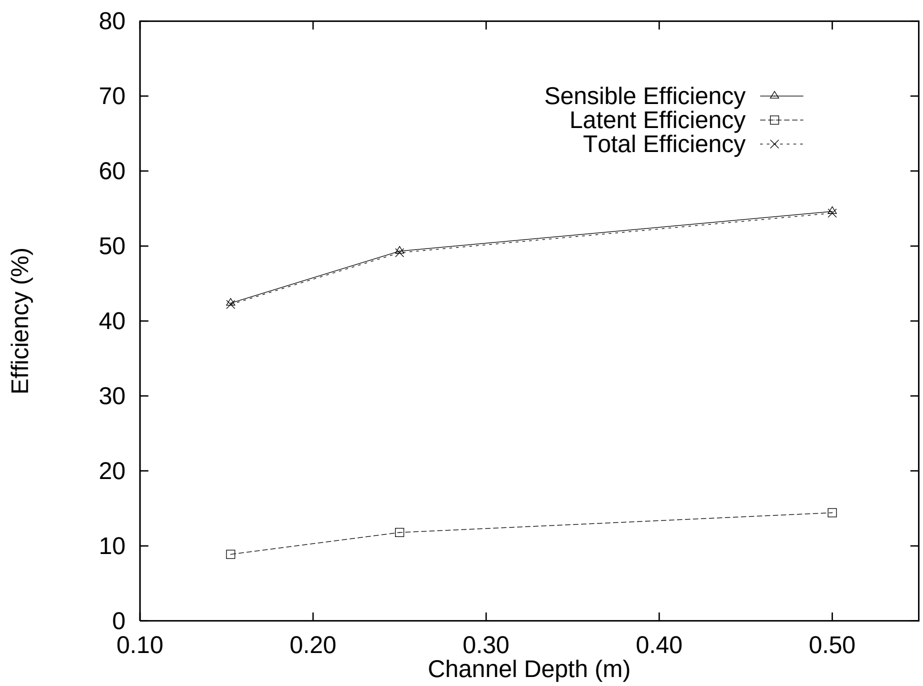

Figure 5.5 Polyvinyl Alcohol Foam Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Channel Depth

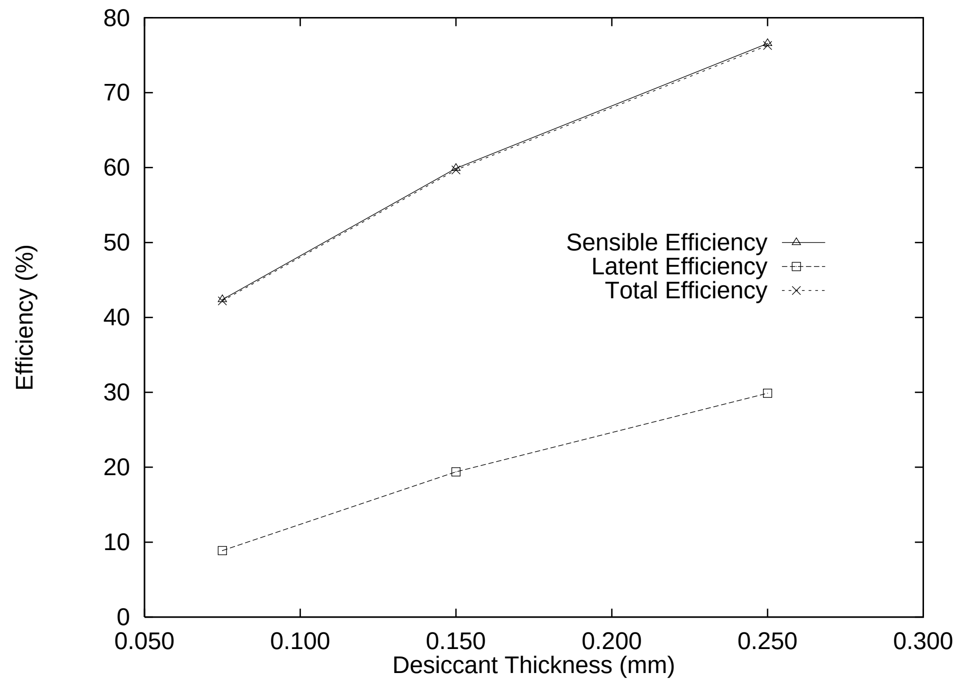

Figure 5.6 Polyvinyl Alcohol Foam Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Desiccant Thickness

constant-thickness aluminum substrate is increased, an increase in all efficiencies is achieved. Despite some differences in sorption kinetics between the absorbing and adsorbing models, the addition of desiccant still results in a change in the volume-averaged effective properties of the dehumidifier. Using the same increase in desiccant thickness as in Section 5.1.1, more significant increases in efficiencies were discovered in the polyvinyl alcohol foam study than the polyvinyl alcohol film study. The polyvinyl alcohol foam exhibited more sensitivity to the desiccant thickness increase. An increase in desiccant thickness from .075 mm to .250 mm resulted in a 80.8% increase in total efficiency for the polyvinyl alcohol foam while the polyvinyl alcohol film experienced a 31.6% increase in total efficiency.

Wheel performance changes with variation of the substrate thickness were also studied. The results of these runs are included in Fig. 5.7. Since it was assumed that the type of polyvinyl alcohol foam used in the modeling was structurally capable of being used in the rotary wheel application without a supporting substrate, a substrate thickness of $0.0mm$ was added to the study. It is shown that the introduction of aluminum to the wheel yielded a growth in sensible and total efficiency, but subsequent increases in the aluminum content of the wheel produced a negative effect on the sensible (and thus the total) efficiency of the system. Figure 5.7 indicates that the mass transfer capability of the wheel, reflected by the

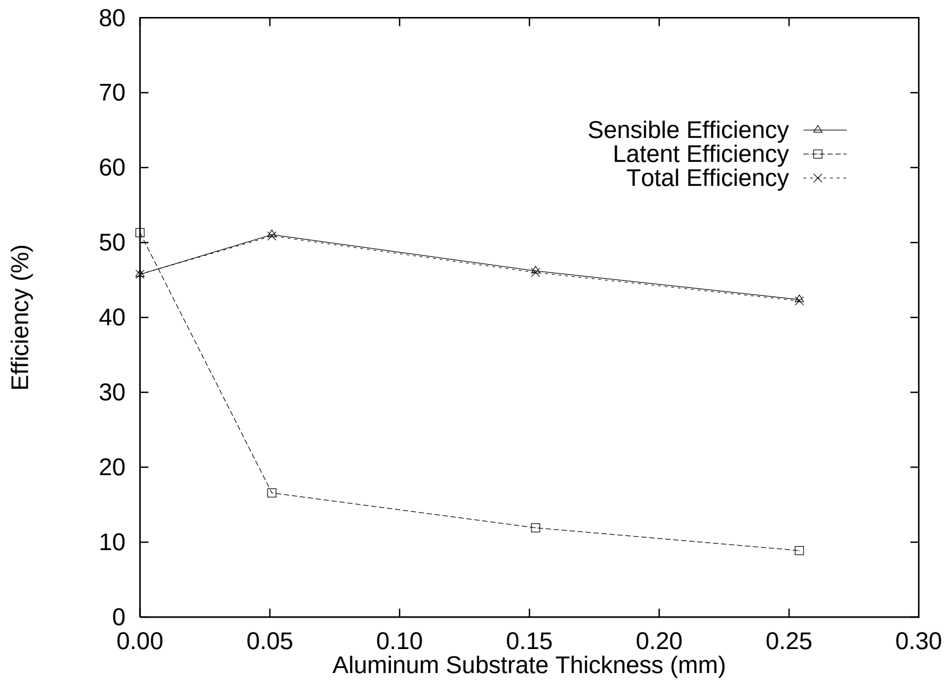

Figure 5.7 Polyvinyl Alcohol Foam Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Aluminum Substrate Thickness

latent efficiency, was significantly higher without the presence of aluminum in the wheel. Since aluminum has no affinity for water, any amount of aluminum in the wheel will affect the sorption characteristics of the system in an adverse manner.

As in Section 5.1.1, the rotary wheel total enthalpy exchanger response to changing air stream velocities was determined. By increasing and decreasing the speed at which both the supply and exhaust air passed through the channels of the dehumidifier, changes were made to the performance of the wheel. The operating air speed reflects the amount of time the air is allowed to be conditioned within the rotary wheel. According to Fig. 5.8, the more resident time the air is given, the more heat and mass transfer is allowed to take place. Therefore, it was observed that as the velocity of the air flows increased, sensible, latent, and total efficiencies declined.

Due to the significance of the increase in latent efficiency resulting from the lack of aluminum substrate shown in Fig. 5.7, two additional tests were run to demonstrate the effect of surface area and desiccant thickness on pure-polyvinyl alcohol foam rotary wheels. The rotary wheel total enthalpy exchanger systems represented in the following figures are assumed to be completely constructed of polyvinyl alcohol foam and contain no supporting substrate. First, the test used to find the wheel's response to changes in surface area was

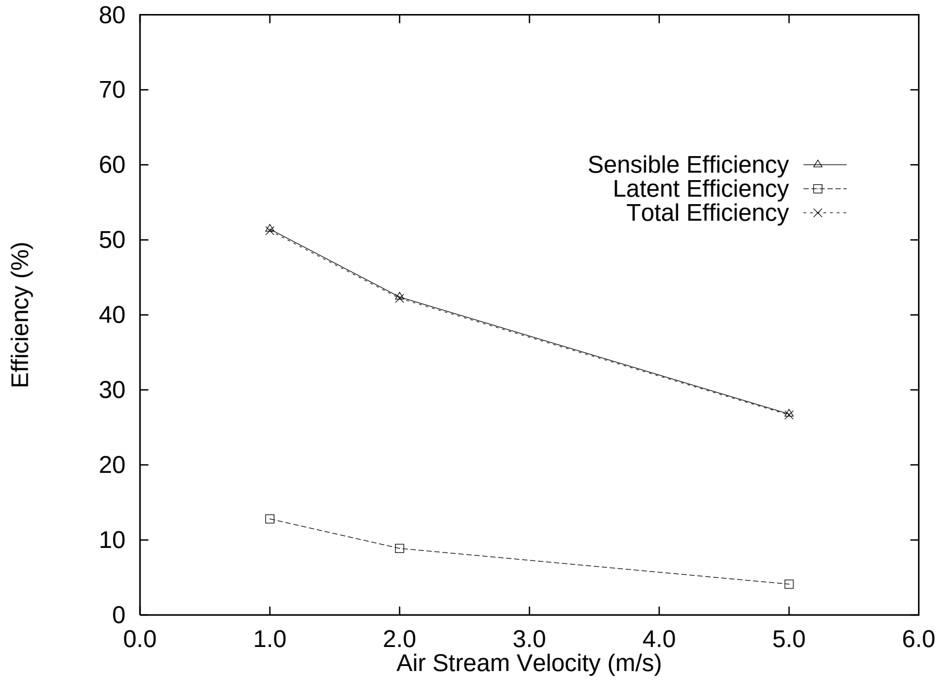

Figure 5.8 Polyvinyl Alcohol Foam Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Air Stream Velocity

re-run. The results of this test are shown in Fig. 5.9 and depict an average latent efficiency increase of over 440% from the previous results of Fig. 5.5. Likewise, the test used to find the wheel's response to changes in desiccant thickness was used again to determine the significance of latent efficiency increase in a pure-foam wheel. The results of this test are shown in Fig. 5.10 and depict an average latent efficiency increase of over 250% from the previous results of Fig. 5.6.

The results of the polyvinyl alcohol foam-coated model runs are summarized in Table 5.4. Overall, the program results produced trends similar to those discovered in Section 5.1.1. The use of polyvinyl alcohol foam noticeably increased the latent performance in every test. Although all configurations produced latent efficiency outputs superior to those of the polyvinyl alcohol film-coated rotary wheel total enthalpy exchanger, those configurations consisting entirely of polyvinyl alcohol without an aluminum substrate exhibited the best mass transfer behavior. In fact, the latent efficiency was at least twice-improved on every situation studied in the no-substrate systems. Once these results were gathered, an investigation into the possibility of obtaining polyvinyl alcohol foam material was conducted. Unfortunately, a polyvinyl alcohol foam capable of being molded into a rotary wheel total enthalpy exchanger was never discovered. Consequently, another adsorbing material was

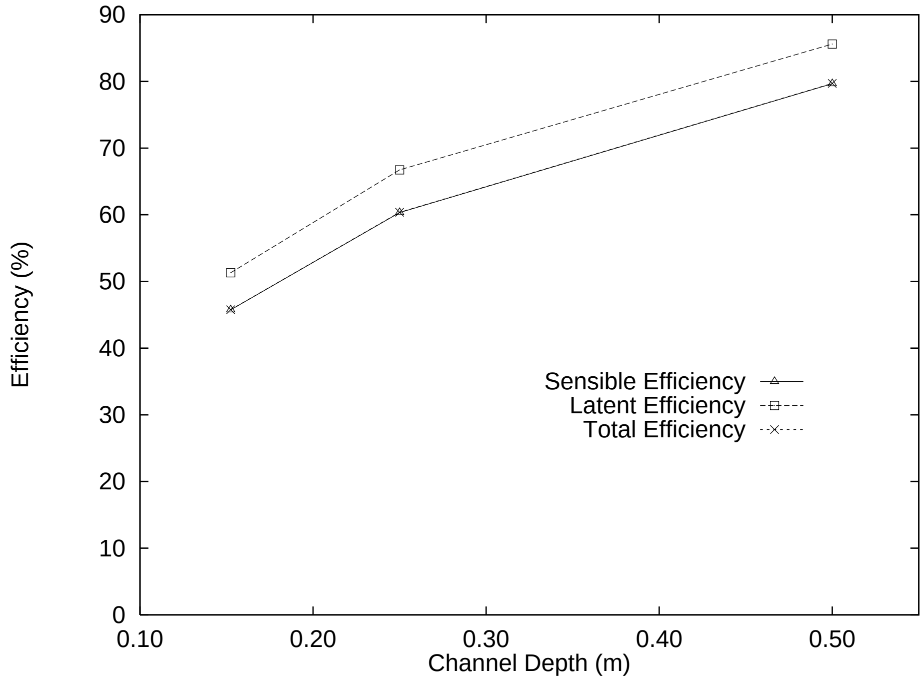

Figure 5.9 Pure Polyvinyl Alcohol Foam Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Channel Depth

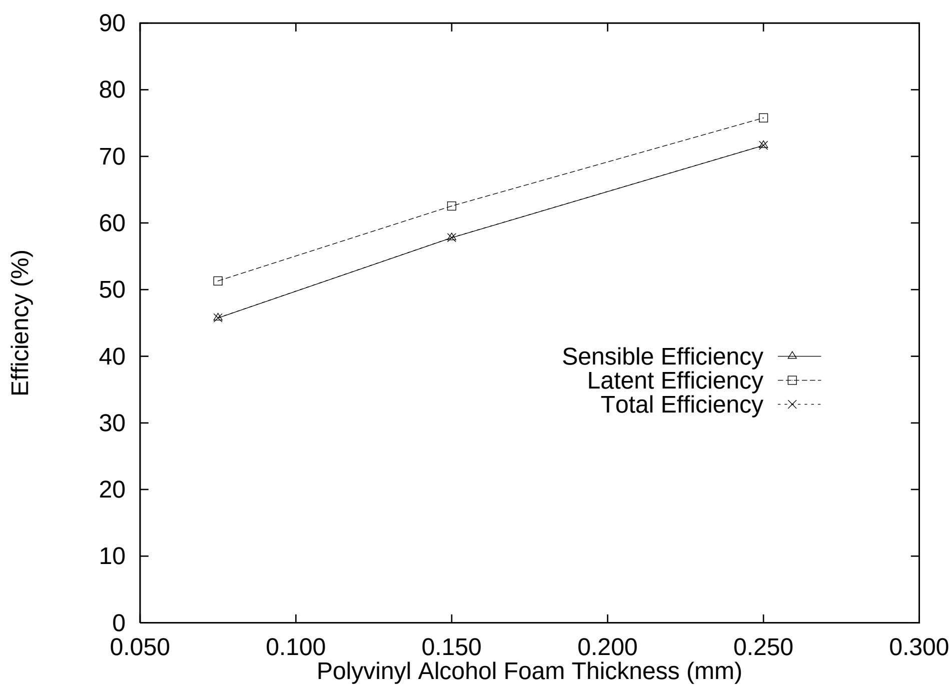

Figure 5.10 Pure Polyvinyl Alcohol Foam Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Desiccant Thickness

Table 5.4 Polyvinyl Alcohol Foam-Coated Rotary Wheel Total Enthalpy Exchanger Model Summary

Parameter	Change in Parameter	Change in Sensible Eff.	Change in Latent Eff.	Change in Total Eff.
Channel Depth	.348 m	28.9 %	62.7 %	28.9 %
Foam Thickness	0.175 mm	80.6%	236.9 %	80.8 %
Substrate Thickness	.254 mm	20.4%	478.5 %	20.5 %
Air Stream Velocity	4.0 m/s	92.0%	212.2 %	92.1%

brought into the research. Using the advantages offered by ceramic desiccants in conjunction with a polyvinyl alcohol film binder a polyvinyl alcohol/silica gel/molecular composite was developed and evaluated. The results of this study are presented in the next section.

Polyvinyl Alcohol/Ceramic Composite

Material properties for the polyvinyl alcohol/ceramic composite coating were obtained by volume averaging polyvinyl alcohol film, silica gel, and molecular sieve properties. Incorporating these material properties presented in Chapter 3 into the adsorbing one-dimensional model, sensible, latent, and total efficiencies of the polyvinyl alcohol/ceramic-coated rotary wheel total enthalpy exchanger were determined. The polyvinyl alcohol/ceramic composite was ideal because it could be easily mixed and coated onto aluminum sheets for the manufacture of rotary wheel total enthalpy exchanger systems. Test cases were run to demonstrate the wheel's performance with respect to surface area, desiccant thickness, aluminum substrate thickness, and operating air stream velocity. The results of these runs are shown in Figs. 5.11 - 5.14.

As was the case with all desiccant material tests, other parameters were held constant as one parameter was varied. Default values for each parameter are given in Table 5.3. The first variable studied was the effect of available surface area contributed by the desiccant coating on rotary wheel efficiency. This was accomplished by varying the desiccant channel depth or, the depth of the desiccant wheel. These results are illustrated in Fig. 5.11. Trends from this graph, similar to those obtained in previous tests, reinforce the idea that an increase in surface area produces an increase in all efficiencies. Figure 5.11 illustrates that the polyvinyl alcohol/composite yields the highest latent efficiencies of all the desiccant coated aluminum substrate configurations. These latent efficiencies, on the average of 22%,

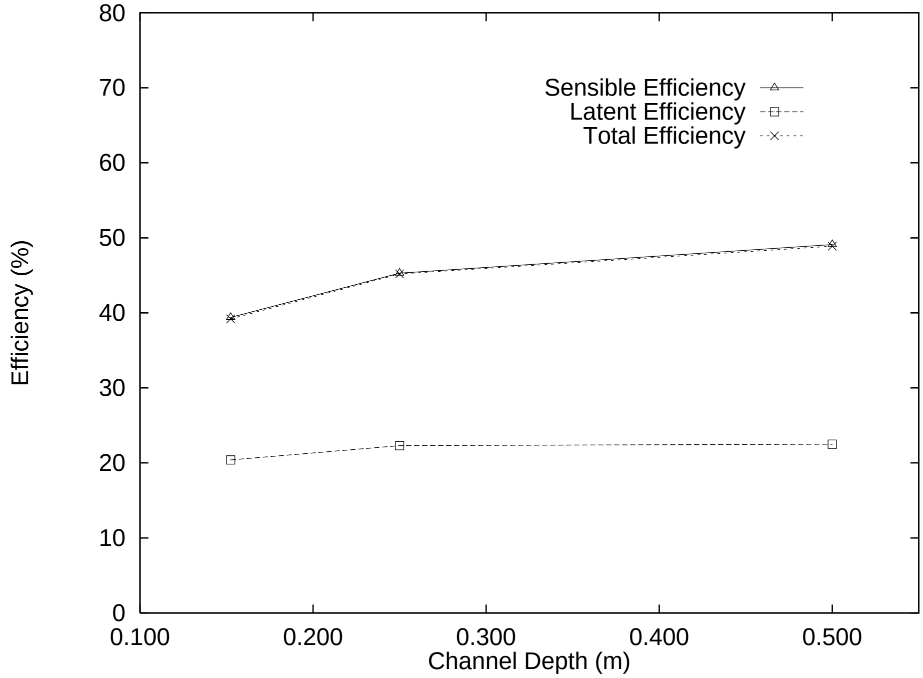

Figure 5.11 Polyvinyl Alcohol/Ceramic Composite Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Channel Depth

are still quite lower than the latent efficiencies of the pure-polyvinyl alcohol foam rotary wheel total enthalpy exchanger, however.

Next, the effect of desiccant thickness on efficiency was evaluated. Illustrated in Fig. 5.12, as the coating on the constant-thickness aluminum substrate was increased, a substantial increase in all efficiencies was realized. Again, the addition of desiccant resulted in a change in the effective properties of the dehumidifier. The test results indicate that as the desiccant thickness is increased from .075 mm to .25 mm, a total efficiency increase of 100% results. Interestingly, the polyvinyl alcohol/ceramic system performance proves extremely sensitive to desiccant coating thickness.

Subsequently, the wheel performance changes with respect to varied substrate thicknesses were observed. The results of these runs are included in Fig. 5.13. Using a desiccant thickness of .075 mm, the changing aluminum content in the wheel results in a desiccant to substrate ratio range of 1.7 to 0.3. As exhibited with both the polyvinyl alcohol film and the polyvinyl alcohol foam, a decrease in aluminum substrate present in the wheel produces an increase in system performance. The mass transfer capabilities are cut in half as the substrate thickness is adjusted from .051 mm to .254 mm.

Finally, the rotary wheel total enthalpy exchanger response to changing air stream

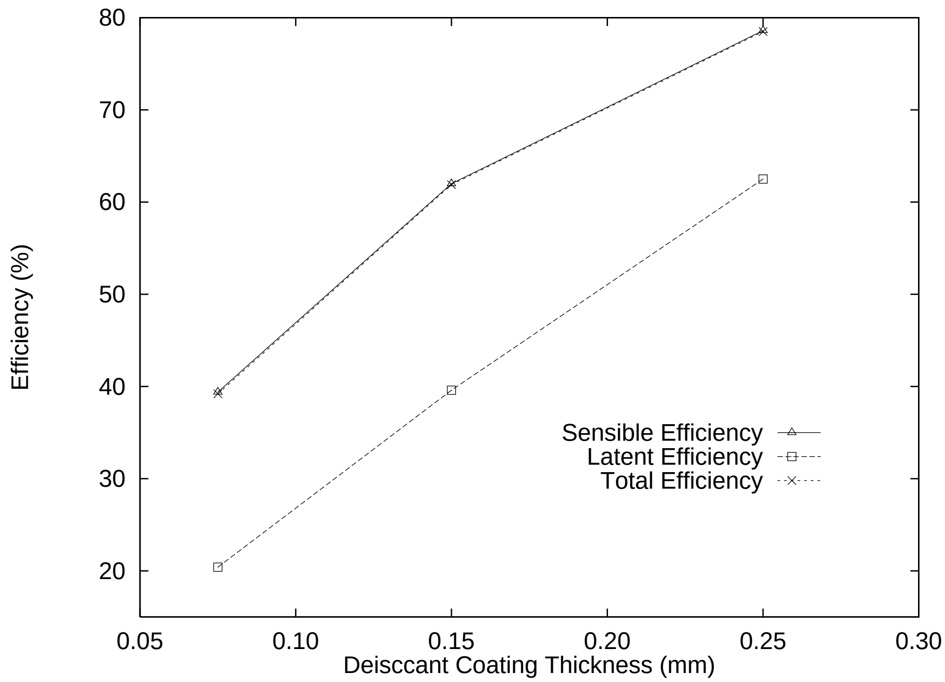

Figure 5.12 Polyvinyl Alcohol/Ceramic Composite Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Desiccant Thickness

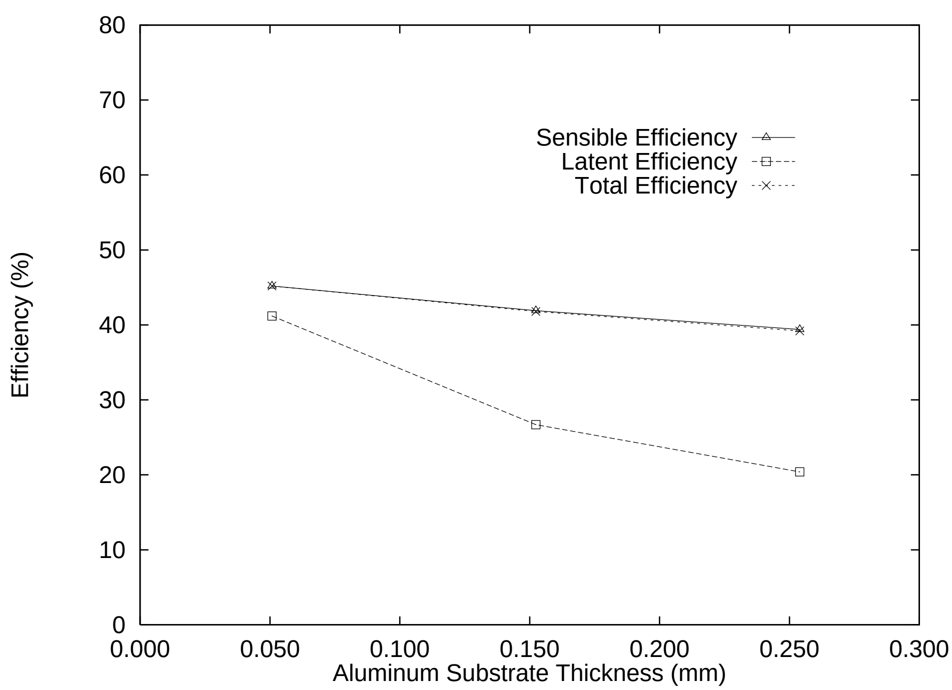

Figure 5.13 Polyvinyl Alcohol/Ceramic Composite Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Aluminum Substrate Thickness

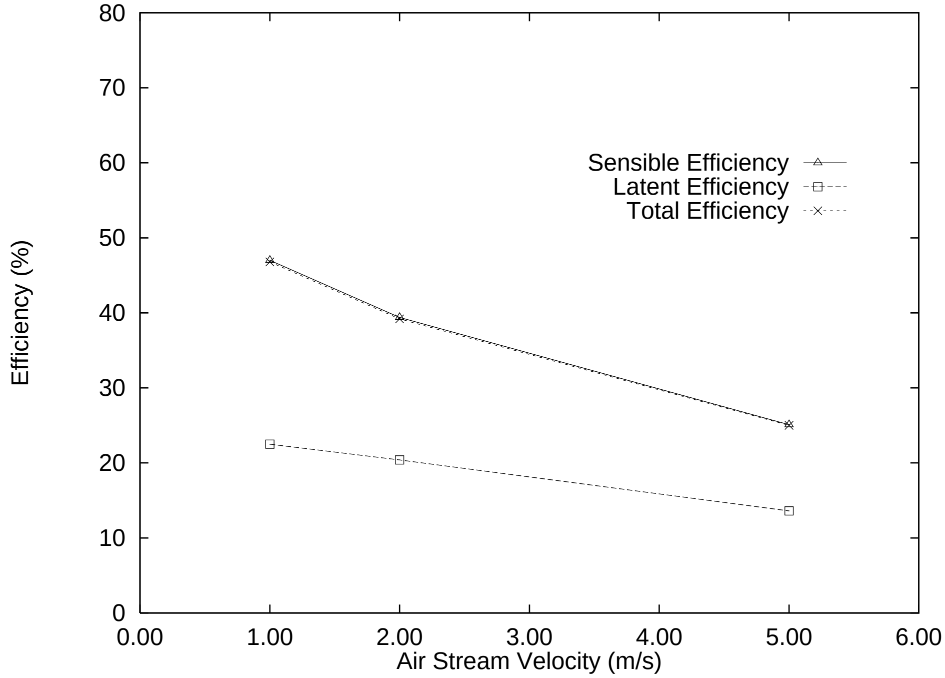

Figure 5.14 Polyvinyl Alcohol/Ceramic Composite Coated Rotary Wheel Total Enthalpy Exchanger Efficiency vs. Air Stream Velocity

velocities was determined. As in the previous tests, changes were made to the performance of the wheel by increasing and decreasing the speed at which both the supply and exhaust air passed through the channels of the dehumidifier. The operating air speed reflects the amount of time the air is allowed to be conditioned within the rotary wheel. According to Fig. 5.14, the more resident time the air is given, the more heat and mass transfer is allowed to take place. Therefore, it was observed that as the velocity of the air flows increased, sensible, latent, and total efficiencies declined.

The results of the polyvinyl alcohol/ceramic composite-coated model runs are summarized in Table 5.5. Program results for the polyvinyl alcohol/ceramic composite study produced likely trends similar to those of the polyvinyl alcohol film and foam. Of all the substrate-coated systems, the best mass transfer was attained with this configuration. Figures 5.12 and 5.13 suggest that this composite coating could potentially produce efficiencies in the range of the pure polyvinyl alcohol foam wheel if the right configuration was discovered.

Table 5.5 Polyvinyl Alcohol/Ceramic Composite-Coated Rotary Wheel Total Enthalpy Exchanger Model Summary

Parameter	Change in Parameter	Change in Sensible Eff.	Change in Latent Eff.	Change in Total Eff.
Channel Depth	.348 m	24.6 %	10.3 %	24.7 %
Foam Thickness	0.175 mm	99.5 %	206.4 %	100.3 %
Substrate Thickness	.254 mm	14.7%	102.0 %	15.3%
Air Stream Velocity	4.0 m/s	87.3 %	65.4 %	87.2 %

5.1.3 Material Comparisons

The three polyvinyl alcohol material coatings evaluated above were tested in comparison to a currently commercially desiccant dehumidifier manufactured by LaRoche Air Systems. The desiccant material used in the LaRoche application appears to be a ceramic coating on a fibrous paper substrate. In a manner similar to the pure polyvinyl alcohol foam wheel model, the LaRoche paper wheel needs no supporting substrate; the wheel is made entirely of the ceramic-coated paper. To effectively compare each desiccant material, the LaRoche paper material properties were determined by the same techniques used to obtain the different polyvinyl alcohol coating properties. These properties are included in Table 5.6.

Just as the LaRoche paper material properties were evaluated in the same manner as the polyvinyl alcohol properties, each comparative test was run under the same design parameters. These parameters, listed in Table 5.3, were used throughout the rotary wheel total enthalpy exchanger material evaluations. They were the same default parameters used in the polyvinyl alcohol film, polyvinyl alcohol foam, and polyvinyl alcohol/ceramic performance estimates. In addition, such parameters as channel depth, channel height, and

Table 5.6 Estimated Material Properties for LaRoche Paper

Material Property	Units	Estimated Value
Density, ρ	kg/m^3	1060.
Specific Heat, C_d	$kJ/kg \cdot K$	2500.
Thermal Diffusivity, α	m/s	2.0 x 10^{-7}
Thermal Conductivity, k	$W/m^2 \cdot K$	0.53
Mass Diffusivity, D_{ab}	m^2/s	4.0 x 10^{-10}
Mass Transfer Coefficient, K_y	$kg/m^2 \cdot s$	order of 10^{-2}
Maximum Moisture Capacity, W_{max}	$kg_{water}/kg_{desiccant}$	0.025

material thickness have been used in a marketed LaRoche Air Products rotary wheel total enthalpy exchanger.

As discussed in Chapter 4, an estimation of the uncertainty associated with the performance estimates adds more significance to the model results. Therefore, to gain more understanding of the comparative results, an uncertainty analysis was performed. Intermediate results from this analysis are represented in Table 5.7.

For each material, properties used in the desiccant evaluation were assigned a standard deviation, σ_i. The values associated with these standard deviations were based somewhat loosely on the experimental data used in the determination of the properties. These property uncertainties were used to determine the final performance uncertainty. Explained in Chapter 4, this is achieved by multiplying the output uncertainty obtained from a small change in property i, $X_{i,\eta}$, with the standard deviation of property i, σ_i. The product emerges as $\sigma_{i,\eta}$, the contribution of the uncertainty in property i to the overall performance estimate uncertainty. Table 5.7 presents uncertainty components, contributed from various properties, for the sensible, latent, and total model efficiencies. If all $\sigma_{i,\eta}$ components are summed, the total uncertainty will be found. These summations accompany the results of the comparative tests in Table 5.8. Total efficiencies resulting from this comparative test are also represented graphically in Fig. 5.15.

To make each dehumidifier equal in volume, the wheels that were modeled without substrate, the pure polyvinyl alcohol foam and the LaRoche wheels, used the respective desiccant material properties in place of aluminum properties when accounting for the substrate. Using the default parameters, the results of the LaRoche paper test surpassed all of the polyvinyl alcohol desiccant performances. However, it must be noted that both the pure polyvinyl alcohol foam and the polyvinyl alcohol/ceramic-coated rotary wheel total enthalpy exchanger systems demonstrated potential for performance comparable to that of the LaRoche paper rotary wheel total enthalpy exchanger. For example, Fig. 5.10 illustrates that a 0.25 mm thick pure polyvinyl alcohol foam wheel can produce 71.6%, 75.8%, and 71.7% sensible, latent, and total efficiencies, respectively. Additionally, Fig. 5.12 indicates that a 0.25 mm thick polyvinyl alcohol/ceramic composite-coated wheel is capable of giving 78.6%, 62.5%, and 78.5% in sensible, latent, and total efficiencies, respectively. These cases suggest that, with an optimized configuration, performance equal to that of the LaRoche wheel can be attained.

Table 5.7 Representation of Material-Specific Uncertainties

Property	σ_i	σ_{i,η_s}	σ_{i,η_l}	$\sigma_{i,\eta_{tot}}$
Polyvinyl Alcohol Film				
W_{max}	±80%	± 0.0	± 1.1	± 0.08
K_y	±20%	± 0.06	± 1.98	± 0.08
C_d	±5%	± 0.020	± 0.01	± 0.025
ρ	±5%	± 0.025	± 0.180	± 0.03
$thick$	±15%	± 0.825	± 0.945	± 0.84
$substr.thick$	±5%	± 0.345	± 0.120	± 0.385
Polyvinyl Alcohol Foam (w/ substrate)				
W_{max}	±50%	± 0.20	± 0.48	± 0.22
K_y	±20%	± 7.956	± 4.212	± 7.92
C_d	±5%	± 2.002	± 1.35	± 1.995
ρ	±5%	± 0.01	± 0.155	± 0.01
$thick$	±15%	± 4.92	± 3.315	± 4.905
$substr.thick$	±5%	± 1.261	± 0.82	± 1.26
Polyvinyl Alcohol Foam (w/o substrate)				
W_{max}	±50%	± 2.50	± 0.40	± 2.50
K_y	±20%	± 2.30	± 5.52	± 2.08
C_d	±5%	± 0.43	± 0.060	± 0.425
ρ	±5%	± 0.18	± 0.105	± 0.175
$thick$	±15%	± 1.575	± 1.215	± 1.56
Polyvinyl Alcohol/Ceramic Composite				
W_{max}	±50%	± 1.90	± 0.20	± 1.90
K_y	±20%	± 1.12	± 1.26	± 1.12
C_d	±5%	± 0.036	± 0.005	± 0.035
ρ	±5%	± 0.100	± 0.793	± 0.095
$thick$	±15%	± 2.97	± 2.67	± 2.97
$substr.thick$	±5%	± 0.280	± 0.555	± 0.28
LaRoche Paper				
W_{max}	±100%	± 0.0	± 0.0	± 0.10
K_y	±20%	± 4.64	± 0.20	± 4.58
C_d	±5%	± 0.01	± 0.0	± 0.01
ρ	±5%	± 0.005	± 0.0	± 0.005
$thick$	±15%	± 3.375	± 0.0	± 3.435

Table 5.8 Comparison Between Different Rotary Wheel Total Enthalpy Exchanger Coatings

Material	Sensible Eff.	Latent Eff.	Total Eff.
Polyvinyl Alcohol Film	57.7 ± 2%	4.6 ± 35%	57.4 ± 2%
Polyvinyl Alcohol Foam (w/ substrate)	42.4 ± 30%	8.9 ± 70%	42.2 ± 30%
Polyvinyl Alcohol Foam (w/o substrate)	45.7 ± 9%	51.3 ± 9%	45.77 ± 9%
Polyvinyl Alcohol/Ceramic Composite	39.4 ± 15%	20.4 ± 27%	39.2 ± 15%
LaRoche Paper	72.0 ± 11%	99.9 ± .2%	72.2 ± 11%

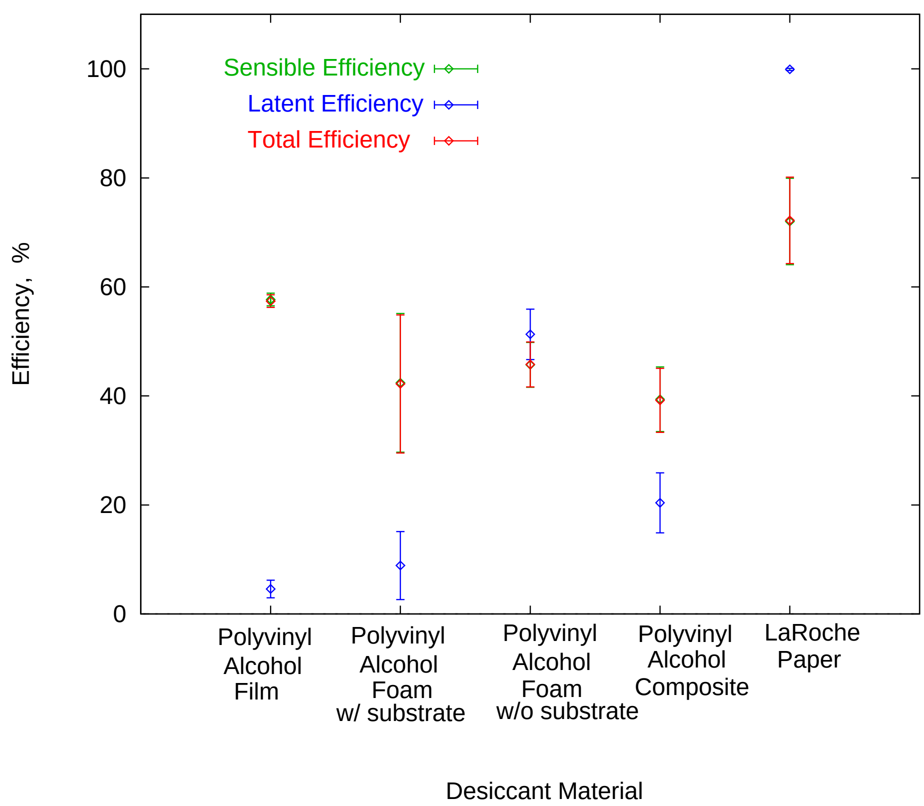

Figure 5.15 Comparison Between Different Rotary Wheel Total Enthalpy Exchanger Coatings

5.2 Two Dimensional Analysis of Fixed Plate Exchanger

As discussed in Chapter 4, the fixed plate total enthalpy exchanger model was used to evaluate the thermal and mass transfer efficiency performance of polyvinyl alcohol foam. Polyvinyl alcohol foam was the only polymer material studied for the fixed plate total enthalpy exchanger applications. This is because the polyvinyl alcohol foam was the only material structurally capable of being used independently as a plate. The other polymer materials, polyvinyl alcohol film and the polyvinyl alcohol/ceramic composite, were only suitable for use in coating applications. Since the fixed plate configuration requires the transport of moisture *through* the plate, coated aluminum could not be used; aluminum can not diffuse moisture. Due to its performance in the rotary wheel total enthalpy exchanger tests examined in Section 5.1.2, polyvinyl alcohol foam was believed to be a promising polymer desiccant material for this type of application.

Model runs predicting the performance of polyvinyl alcohol foam in fixed plate total enthalpy exchangers were conducted and evaluated. As in Section 5.1, LaRoche paper, a material currently used in the rotary wheel total enthalpy exchanger market, was brought in and evaluated as the operating material in a counter-flow fixed plate total enthalpy exchanger. This study was used as a reference with which to compare the polyvinyl alcohol foam materials and their performance. Results of these desiccant material applications and their performance with fixed plate total enthalpy exchanger systems is discussed throughout this section.

5.2.1 Counterflow Plate Exchanger

Incorporating the polyvinyl alcohol foam material properties presented in Chapter 3 into the adsorbing two-dimensional model, sensible, latent, and total efficiencies of the polyvinyl alcohol foam fixed plate total enthalpy exchanger were determined. Test cases were run to demonstrate the counter-flow plate exchanger's performance with respect surface area, desiccant thickness, operating air stream velocity, and plate spacing. The results of these runs are shown in Figs. 5.16 - 5.19.

The test strategy for the fixed plate total enthalpy exchanger models was similar to that of the rotary wheel total enthalpy exchanger models; as one parameter was varied, the other parameters were held constant. Default values for each parameter are given

Table 5.9 Default Parameters Used in Two-Dimensional Polyvinyl Alcohol Foam Plate Model

Parameter	Default Value
Plate Depth	0.762 m
Plate Width	0.0762 m
Desiccant Plate Thickness	2.0 mm
Air Stream Velocity	2.0 m/s
Channel Height	4.68 mm

in Table 5.9. These particular default parameters were chosen because they are typical parameters for existing plate exchangers and polyvinyl alcohol foam. The first variable studied was the effect of available surface area contributed by the desiccant counter-flow plate efficiency. This was accomplished by varying the desiccant plate depth. These results are illustrated in Fig. 5.16. Trends from this graph introduce the idea that the sensible (and thus, total) efficiencies of the foam plate show little sensitivity to an increase in surface area. However, contrary to tests conducted in he previous section, the latent efficiency of the foam plate decreases with plate depth. Perhaps, with the parameters given, supply air dries sufficiently with a short plate depth; as the plate depth increases, the air regains some of its moisture before exiting.

Next, the effect of desiccant plate thickness on exchanger efficiency was evaluated. Illustrated in Fig. 5.17, as the thickness of the polyvinyl alcohol foam plate increased, an increase in all efficiencies was realized. By holding the plate spacing constant, the addition of desiccant resulted an increased desiccant volume throughout the dehumidifier. As in the previous evaluation, the sensible and total efficiencies showed great insensitivity to changes in the desiccant plate thickness. However, the latent efficiency exhibited a decreasing trend as thickness was added. As the desiccant plate became thicker, the path of mass diffusion grew longer and as a result, the plate was not able to diffuse the moisture as easily.

The next task of the study included determining the plate efficiency response to changes in the plate spacing. Adjustments in plate spacing in a fixed plate total enthalpy exchanger directly affect the height of the air channel through which supply and exhaust air travel. The results of these runs are included in Fig. 5.18. As the plates were spaced farther apart, a reduction in both sensible and total efficiencies was realized. Plate spacing contributes directly to the determination of the heat transfer coefficient, h. As the plates

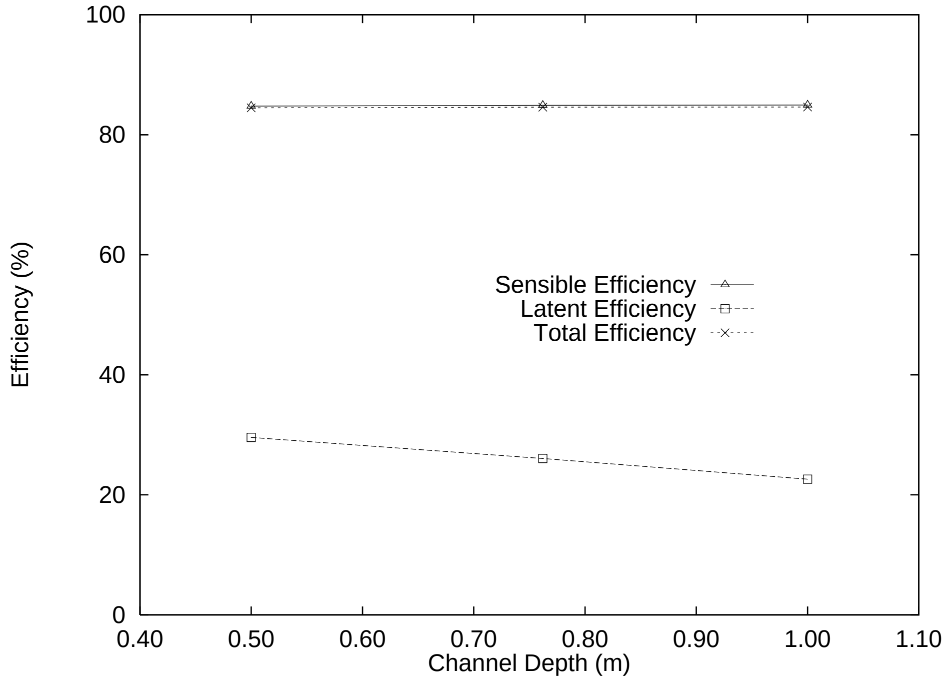

Figure 5.16 Polyvinyl Alcohol Foam Fixed Plate Total Enthalpy Exchanger Efficiency vs. Plate Depth

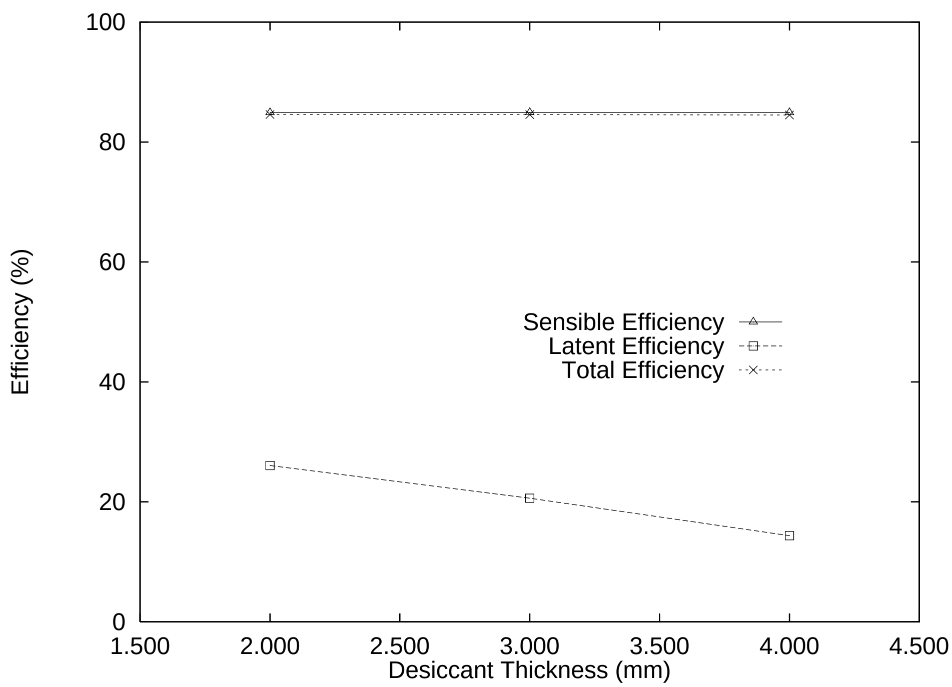

Figure 5.17 Polyvinyl Alcohol Foam Fixed Plate Total Enthalpy Exchanger Efficiency vs. Plate Thickness

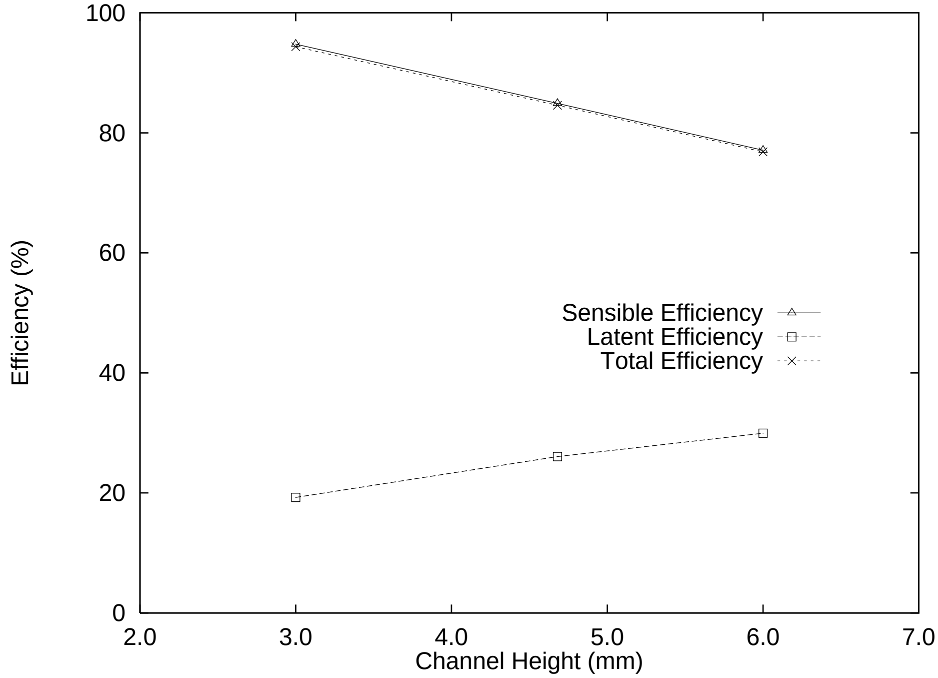

Figure 5.18 Polyvinyl Alcohol Foam Fixed Plate Total Enthalpy Exchanger Efficiency vs. Air Channel Height

grew apart, h was reduced. Thus, the plate's ability to transfer heat, indicated by the sensible efficiency of the system was negatively affected by an increase in the distance between the plates. Conversely, the latent efficiency responded positively to the increased plate spacing.

Finally, the rotary wheel total enthalpy exchanger response to changing air stream velocities was determined. By increasing and decreasing the speed at which both the supply and exhaust air passed through the channels of the dehumidifier, changes were made to the performance of the wheel. The operating air speed reflects the amount of time the air is allowed to be conditioned within the rotary wheel. According to Fig. 5.19, the more resident time the air is given, the more heat and mass transfer is allowed to take place. Therefore, it was observed that as the velocity of the air flows increased, sensible, latent, and total efficiencies declined.

The results of the polyvinyl alcohol foam plate model runs are summarized in Table 5.10. Program results for the polyvinyl alcohol foam plate exchanger study produced justifiable trends. Although certain scenarios produced significantly high sensible and total efficiencies, the latent efficiency was consistently lower, never exceeding 40%. It was decided that a comparative study would be performed to determine is the low mass transfer

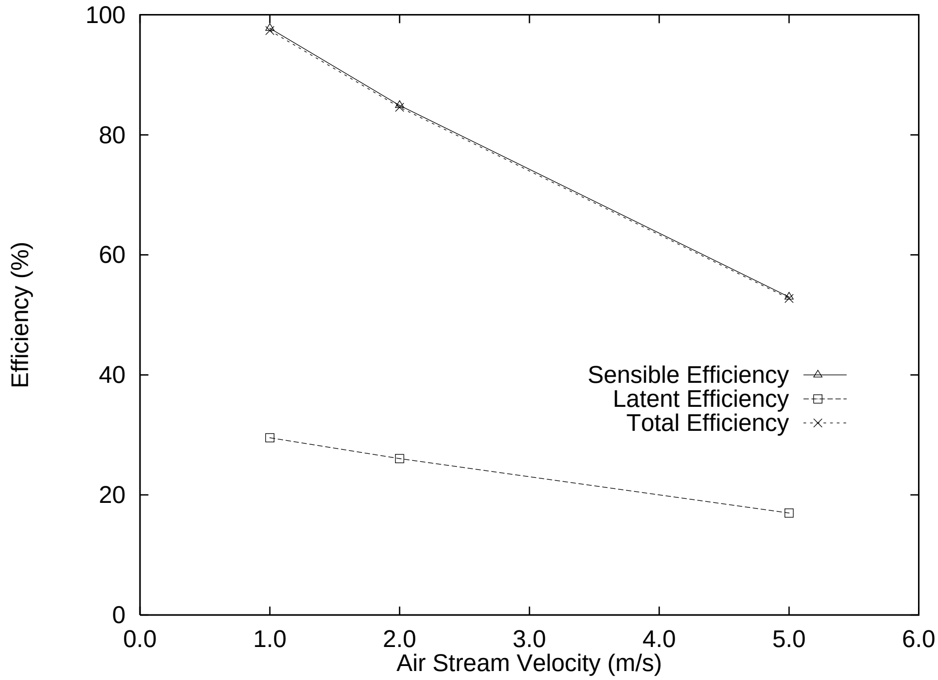

Figure 5.19 Polyvinyl Alcohol Foam Fixed Plate Total Enthalpy Exchanger Efficiency vs. Air Stream Velocity

results were a good indication of the actual plate exchanger performance. A discussion of the comparative study is presented in the next section.

5.2.2 Material Comparisons

The polyvinyl alcohol desiccant plating evaluated above was tested in comparison to a currently commercially desiccant material manufactured by LaRoche Air Systems, a ceramic coating on a fibrous paper substrate. In a manner similar to the pure polyvinyl alcohol foam fixed plate model, the LaRoche paper plate model needs no supporting substrate; the fixed

Table 5.10 Polyvinyl Alcohol Foam Fixed Plate Total Enthalpy Exchanger Model Summary

Parameter	Change in Parameter	Change in Sensible Eff.	Change in Latent Eff.	Change in Total Eff.
Plate Depth	.5 m	0.2 %	30.7 %	0.2 %
Foam Thickness	2.0 mm	0.0 %	73.8 %	0.0 %
Channel Height	3.0 mm	22.9 %	55.5 %	22.8%
Air Stream Velocity	4.0 m/s	84.5 %	73.7 %	84.4 %

Table 5.11 Parameters Used in Two-Dimensional Model Comparisons

Parameter	Default Value
Plate Depth	0.762 m
Plate Width	0.0762 m
Desiccant Plate Thickness	.4 mm
Air Stream Velocity	2.0 m/s
Channel Height	4.68 mm

plate total enthalpy exchanger is assumed to be made entirely of the ceramic-coated paper. To effectively compare these desiccant material, the LaRoche paper material properties were determined by the same techniques used to obtain the different polyvinyl alcohol coating properties. These properties are included in Table 5.6.

Just as the LaRoche paper material properties were evaluated in the same manner as the polyvinyl alcohol properties, the fixed plate total enthalpy exchanger comparative test was run under the same design parameters. These parameters are listed in Table 5.11. These inputs were chosen because such parameters as plate depth, material thickness, and plate spacing have been used in a marketed cross-flow fixed plate total enthalpy exchanger.

Just as an estimation of the uncertainty associated with the performance estimates added more significance to the model results of the one-dimensional rotary wheel total enthalpy exchanger comparisons, an uncertainty analysis of the comparative fixed plate total enthalpy exchanger models also proved insightful. Intermediate results from this analysis are represented in Table 5.12.

As in the rotary wheel analyses, material properties used in the desiccant evaluation were assigned a standard deviation, σ_i; the values associated with these standard deviations were based somewhat loosely on the experimental data used in the determination of the properties. These property uncertainties were used to determine the final performance uncertainty. Explained in Chapter 4, this is achieved by multiplying the output uncertainty obtained from a small change in property i, $X_{i,\eta}$, with the standard deviation of property i, σ_i. The product emerges as $\sigma_{i,\eta}$, the contribution of the uncertainty in property i to the overall performance estimate uncertainty. Table 5.12 presents uncertainty components, contributed from various properties, for the sensible, latent, and total model efficiencies. If all $\sigma_{i,\eta}$ components are summed, the total uncertainty will be found. These summations accompany the results of the comparative tests in Table 5.13. Total efficiencies resulting

Table 5.12 Representation of Material-Specific Uncertainties

Property	σ_i	σ_{i,η_s}	σ_{i,η_l}	$\sigma_{i,\eta_{tot}}$
Polyvinyl Alcohol Foam				
W_{max}	±50%	± 2.0	± 0.005	± 2.0
K_y	±20%	± 2.8	± 0.45	± 2.85
C_d	±5%	± 0.01	± 0.001	± 0.01
ρ	±5%	± 0.23	± 0.002	± 0.22
$thick$	±15%	± 0.0	± 0.04	± 0.01
D_{ab}	±55%	± 1.63	± 0.20	± 1.64
α	±500%	± 3.10	± 0.002	± 3.10
LaRoche Paper				
W_{max}	±100%	± 1.0	± 0.01	± 1.0
K_y	±20%	± 5.63	± 4.24	± 5.73
C_d	±5%	± 0.02	± 0.008	± 0.02
ρ	±5%	± 0.32	± 0.13	± 0.32
$thick$	±15%	± 0.08	± 0.08	± 0.08
D_{ab}	±65%	± 2.01	± 1.25	± 2.02
α	±500%	± 3.54	± 0.01	± 3.54

from this comparative test are also represented graphically in Fig. 5.20.

Using the default parameters, the total efficiency of the polyvinyl alcohol desiccant plate test surpassed that of the LaRoche paper test. However, the LaRoche paper exhibited a higher latent efficiency than the polyvinyl alcohol foam. Using other geometric parameters, the pure polyvinyl alcohol foam fixed plate total enthalpy exchanger system demonstrated potential for latent performance exceeding to that of the LaRoche paper fixed plate total enthalpy exchanger. For example, a 1.0 mm thick plate exchanger at an air speed of 1.5 m/s produces 99.9%, 52.4%, and 99.7% sensible, latent, and total efficiencies, respectively.

Table 5.13 Comparison Between Different Fixed Plate Total Enthalpy Exchanger Materials

Material	Sensible Eff.	Latent Eff.	Total Eff.
Polyvinyl Alcohol Foam	97.7 ± 10%	1.9 ± 35%	96.8 ± 10%
LaRoche Paper	53.2 ± 33%	21.6 ± 30%	53.0 ± 33%

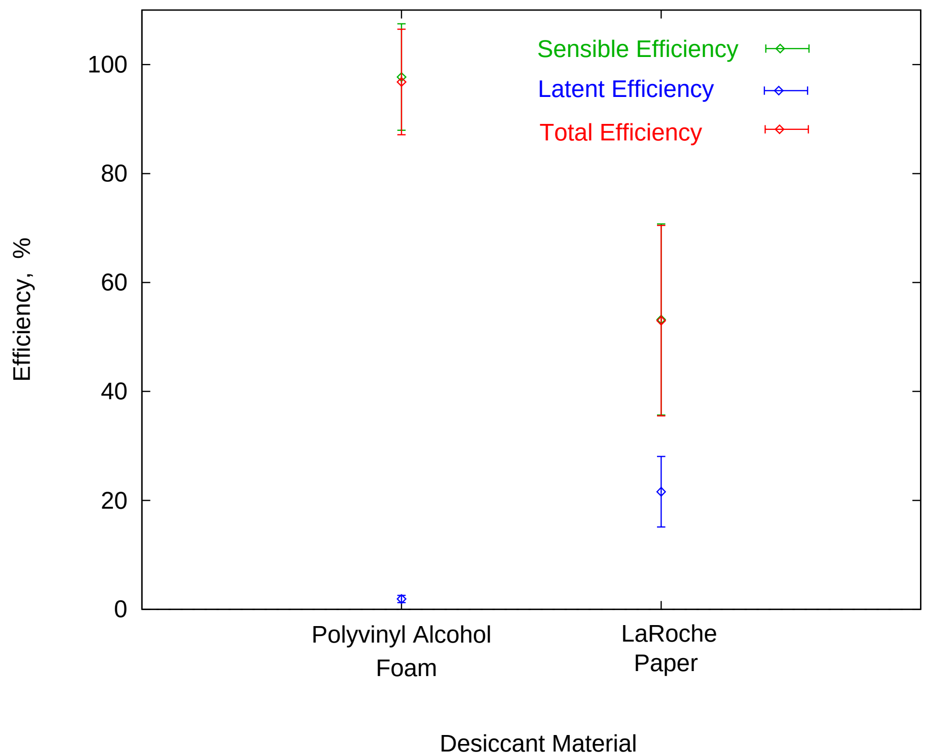

Figure 5.20 Comparison Between Different Fixed Plate Total Enthalpy Exchanger Materials

Chapter 6

Summary and Conclusions

The Mechanical Engineering and the Materials Science Engineering Departments at Virginia Tech have been working cooperatively for the past two years under the direction of Des Champs Laboratories, Inc. to determine appropriate polymeric materials for use in both rotary wheel total enthalpy exchanger and fixed plate total enthalpy exchanger systems. A summary of work completed during the course of this project, as well as concluding remarks and observations are presented in this chapter.

6.1 Summary

This study focused on the development and determination of appropriate polymeric desiccant materials for use in two different coupled heat and mass transfer applications. After an initial study of several candidate materials, it was decided that polyvinyl alcohol best met the pre-determined selection criteria. Various polyvinyl alcohol materials were modeled as operating desiccants in two different applications: a rotary wheel total enthalpy exchanger and a fixed plate total enthalpy exchanger. The rotary wheel total enthalpy exchanger study was accomplished first, followed by the fixed plate total enthalpy exchanger analysis.

Finite difference equations and differential equations were developed for the rotary wheel total enthalpy exchanger model through the use of the control volume technique. Using the principles of conservation of energy and conservation of mass, governing equations were derived. During the modeling of the rotary wheel total enthalpy exchanger system, four desiccant materials were evaluated: polyvinyl alcohol film, polyvinyl alcohol foam, polyvinyl alcohol/silica gel/molecular sieve composite, and LaRoche paper (a desiccant paper manufactured by LaRoche Air Products). Each material's sensible, latent, and total performance was evaluated with respect to available desiccant surface area, desiccant thickness, supporting substrate thickness, and operating supply and exhaust air speed. The LaRoche material evaluation was used in an effort to compare the three polyvinyl alcohol materials' performances as desiccant coatings for rotary wheel total enthalpy exchanger systems.

Finite difference equations and differential equations were also developed for the fixed plate total enthalpy exchanger model through the use of the control volume technique. As in the rotary wheel total enthalpy exchanger model development, principles of conservation of energy and conservation of mass were used to derive governing equations. During the modeling of the fixed plate total enthalpy exchanger system, two desiccant materials were evaluated: polyvinyl alcohol foam, polyvinyl alcohol/silica gel/molecular sieve and LaRoche paper. Polyvinyl alcohol foam was the only polymer material studied capable of being used as a plate independent of a supporting structure. Each material's sensible, latent, and total performance was evaluated with respect to available desiccant surface area, desiccant thickness, plate spacing, and operating supply and exhaust air speed. The LaRoche material evaluation was used in an effort to compare the polyvinyl alcohol foam's performance as desiccant plating for a fixed plate total enthalpy exchanger system.

6.2 Conclusions

The objectives of this project, laid out in Chapter 1 and reviewed again in the previous section, were completed and are described throughout this work. As each polyvinyl alcohol polymer material was selected, it was modeled for its appropriate application. Two applications were studied: the rotary wheel total enthalpy exchanger and the fixed plate total enthalpy exchanger. The following lists include conclusions for each system analysis completed though the course of this work.

6.2.1 One-Dimensional Rotary Wheel Total Enthalpy Exchanger Analysis

1. Due to its non-porous nature, polyvinyl alcohol film is not suitable for use as a desiccant coating for rotary wheel total enthalpy exchangers. However, when used as

a binder for an adsorbing silica gel/molecular sieve composite coating, the resulting coated rotary wheel total enthalpy exchanger performance is greatly improved.

2. Every coated rotary wheel total enthalpy exchanger system, regardless of operating desiccant material, showed increased performance after an increase in the amount of available desiccant surface area.

3. Every coated rotary wheel total enthalpy exchanger system, regardless of operating desiccant material, showed increased performance after a decrease in the amount of aluminum substrate.

4. Every coated rotary wheel total enthalpy exchanger system, regardless of operating desiccant material, showed increased performance after an increase in desiccant coating thickness.

5. Every coated rotary wheel total enthalpy exchanger system, regardless of operating desiccant material, showed an increased performance with a decrease in operating supply and exhaust air speed.

6. Under the parameters tested, rotary wheel total enthalpy exchanger systems capable of operating without a supporting substrate appear to perform better than those with supporting material.

7. With the right wheel design, pure polyvinyl alcohol foam and polyvinyl alcohol/ceramic coated wheels have the potential to match, or surpass, the performance of the LaRoche Air Products rotary wheel total enthalpy exchanger.

6.2.2 Two-Dimensional Fixed Plate Total Enthalpy Exchanger Analysis

1. Polyvinyl alcohol open-cell foam was the only polyvinyl alcohol polymer material studied in the one-dimensional rotary wheel total enthalpy exchanger analysis that was capable of being used for the fixed plate total enthalpy exchanger analysis.

2. Under the parameters tested, the sensible efficiency of the polyvinyl alcohol foam fixed plate total enthalpy exchanger model was insensitive to both plate depth and plate thickness.

3. Under the parameters tested, the latent efficiency of the polyvinyl alcohol foam fixed plate total enthalpy exchanger model decreased with increases in plate depth and plate thickness.

4. The polyvinyl alcohol foam fixed plate total enthalpy exchanger system showed an increased performance with a decrease in operating supply and exhaust air speed.

5. With the appropriate exchanger design, polyvinyl alcohol foam counter-flow plate exchangers have the potential to perform well as fixed plate total enthalpy exchangers.

6.2.3 Overall Conclusion

The ultimate goal of this research, as stated in Chapter 1, was to develop and model a material that, when applied to an existing heat exchanger, would achieve the heat *and* mass transfer in a single process necessary to transform the heat exchanger into a *total enthalpy exchanger*. The model tests conducted in Chapter 5 suggest that two polyvinyl alcohol materials, polyvinyl alcohol open-cell foam and polyvinyl alcohol/silica gel/molecular sieve composite, can be successfully used in total energy exchange applications. With cases resulting in total efficiencies exceeding 65% and latent efficiencies of over 50%, both polyvinyl alcohol foam and polyvinyl alcohol/ceramic composite were shown to perform well as desiccants in rotary wheel total enthalpy exchanger systems. In addition, with a total efficiency of greater than 80%, polyvinyl alcohol foam was shown to be efficient in fixed plate total enthalpy exchanger systems. These cases are presented below in Fig. 6.1. As indicated in Chapter 5, this research concludes that with appropriate optimization of exchanger design, these two polyvinyl alcohol materials have the potential to perform as well as desiccant materials used in current rotary wheel and fixed plate total energy exchanger applications.

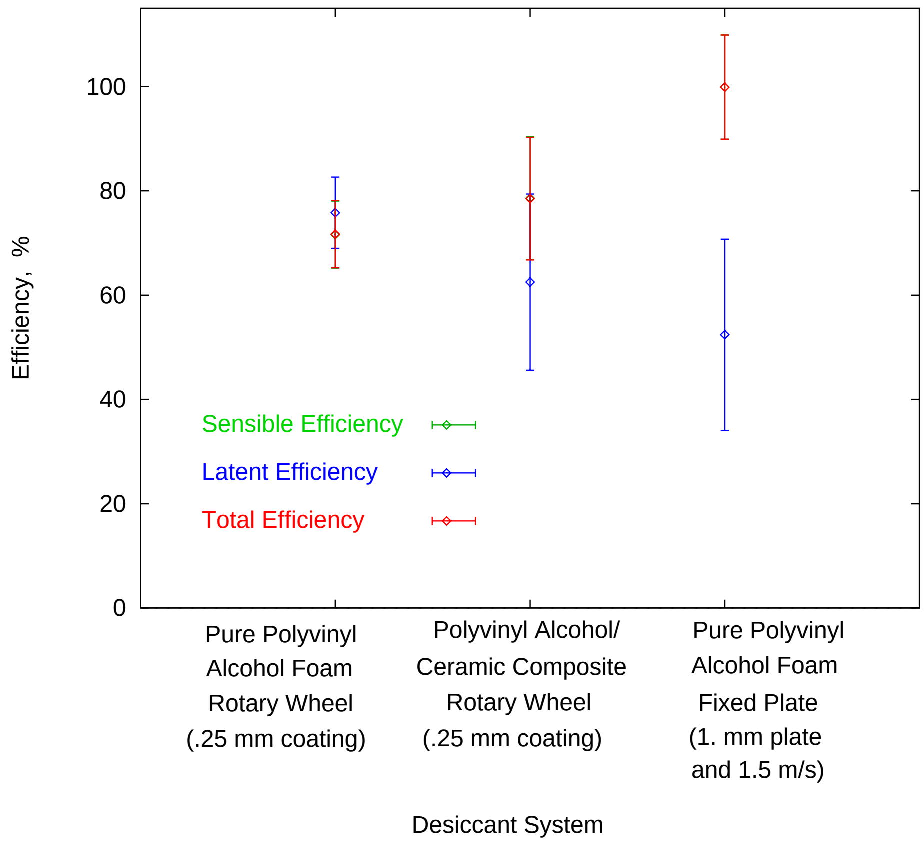

Figure 6.1 Examples of Increased Efficiency Scenarios

Chapter 7

Recommendations

The performance efficiency codes developed during this study produced acceptable results. Although not yet validated by experimental data, trends discussed in Chapter 5 seem intuitively and logically correct. Several assumptions made during the theoretical development of each model could contribute to any error that may occur between the numerical estimations and potential experimental outcomes. Refining these assumptions will produce more accurate estimations. The items listed below are recommended for change in any further modeling work.

- Allow for maximum moisture content (W_{max}), material diffusivity (D_{ab}), and mass transfer coefficient (K_y) of study material to be a function of surrounding temperature and relative humidity.
- Enhance model with the consideration of axial conduction in both the air stream and the desiccant bed.
- Enhance model with the consideration of axial diffusion in both the air stream and the desiccant bed.
- Improve model by allowing inlet exhaust temperature to be a function of the exit supply temperature.
- Investigate the effects of radiation through the air stream.
- Continue the modeling efforts to include the study of a cross-flow fixed plate total enthalpy exchanger.

- Perform a detailed cost analysis and market study regarding the implementation of this technology into the air conditioning industry.

Due to several limitations encountered during this research, it was decided that several material properties that could have been slightly dependent on surrounding temperature and moisture content would be estimated at a given environment. Three material properties, W_{max}, D_{ab}, and K_y, were estimated from material moisture uptake data at an environment of 75 °F and 95 % relative humidity. These properties could be more accurately represented within each numerical model if enough moisture uptake data was recorded so that a surface function could be produced for each of these properties representing the behavior of the property with respect to temperature and surrounding moisture levels.

The governing energy equations for each model could be improved by considering heat and mass transfer effects that could take place axially along the dehumidifier. Intuitively these effects would be small compared to the effects already considered through the thickness of the dehumidifier, but their inclusion within the model would yield a more exact solution.

A study involving the possible contribution of radiative effects to the overall energy balance of the air stream would help in providing the most comprehensive representation of the heat transfer within the convective air stream.

A significant number of fixed plate energy exchangers operate in a cross-flow fashion. As a result, it is recommended that a numerical model representing this particular flow pattern be developed. Development of a three dimensional cross-flow model would provide a more complete collection of energy exchanger models. If a new desiccant is discovered in the future, the models will all be available to provide performance analyses.

Bibliography

Allcock, H. R., 1996, *Hydrophilic Polymers, Performance with Environmental Acceptibilty*, Chap. Water-Soluble Polyphosphazenes and Their Hydrogels, American Chemical Society, Washington, DC, pp. 3–29.

Anderna, A., Tillman, W., and Herdi, T., 1995, "Polymers as Advanced Materials for Desiccant Applications Part 3: Alkali Salts of PSSA and PolyAMPSA and Copolymers of PolyAMPSASS," in: *Proceedings of the 1995 ASHRAE Annual Meeting*, Chicago, IL, pp. 578–586.

Anon., 1994, "Integral Water-Based A/C Systems," *H & V Engineer*, Vol. 68, No. 726, pp. 11.

Anon., 1996, *ASHRAE Standards*, Vol. 102, Chap. 42, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, New York, NY, pp. 42.1–42.18.

Anon., 1997, "Two-Wheel Desiccant Dehumidification System," Tech. rep., U.S. Department of Energy, http://www.pnl.gov/fta/8_tdd.htm.

Banks, P. J., 1971, "Coupled Equilibrium Heat and Single Adsorbate Transfer in Fluid Flow Through a Porous Medium-I Characteristic Potentials and Specific Capacity Ratios," *Chemical Engineering Science*, Vol. 27, pp. 1143–1154.

Berthold, J., Desbrières, J., Rinuado, M., and Salmén, L., 1994, "Types of Absorbed Water in Relation to the Ionic Groups and Their Counter-Ions for Some Cellulose Derivatives," *Polymer*, Vol. 35, No. 26, pp. 5729–5736.

Cai, Y., Shanks, R. A., and Wlodarski, W. B., 1995, "Study of Water Vapor Sorption on Polymer Thin Films by a Quartz Crystal Resonator," *Material Science Forum*, Vol. 189, pp. 193–198.

Chant, E. and Jeter, S., 1995, "On the Use of the Parabolic Concentration Profile Assumption for a Rotary Desiccant Dehumidifier," *Journal of Solar Energy Engineering*, Vol. 117, No. 1, pp. 45–50.

Close, D. J. and Banks, P. J., 1971, "Coupled Equilibrium Heat and Single Adsorbate Transfer in Fluid Flow Through a Porous Medium -II Predictions for a Silica Gel Air-Drier Using Characteristic Charts," *Chemical Engineering Science*, Vol. 27, pp. 1157–1168.

Collier, R. K., Barlow, R. S., and Arnold, F. H., 1982, "An Overview of Open-Cycle Desiccant-Cooling Systems and Materials," *Journal of Solar Energy Engineering*, Vol. 104, pp. 28–34.

DiBlasio, R., 1995, "Conditioning a Research Facility," *Engineered Systems*, Vol. 12, No. 9.

Han, H., Gryte, C., and Ree, M., 1995, "Water Diffusion and Sorption in films of High Performance Poly(4,4 prime-oxydiphenylene pyromellitimide): Effects of Humidity, Imidization History, and film Thickness," *Polymer*, Vol. 36, No. 8, pp. 1662–1672.

Harland, R. S., 1993, "Ultimate Superabsorbents," in: *Proceeding of the ACS Division of Polymeric Materials Science and Engineeering*, Vol. 69, p. 564.

Harriman, L., 1994, "Basics of Commercial Desiccant Systems," *Heating, Piping, and Air Conditioning*, Vol. 66, No. 7, pp. 77–85.

Holman, J. P., 1981, *Heat Transfer*, fifth edition edn., McGraw-Hill Publishing Company.

Ishihara, I. and Matsumoto, R., 1995, "Heat and Mass Transfer Under Vaporization from Liquid Surface into Cold Gas Stream," in: *ASME/JSME Thermal Engineering Joint Conference - Proceedings*, Vol. 4, ASME, New York, pp. 477–482.

Jain, S., Dhar, P., and Kaushik, S., 1995, "Evaluation of solid-Desiccant-Based Evaporative Cooling Cycles for Typical Hot and Humid Climates," *International Journal of Refrigeration*, Vol. 18, No. 5, pp. 287–296.

Jurinak, J. J., Mitchell, J. W., and Beckman, W. A., 1984, "Open-Cycle Desiccant Air Conditioning as an Alternative to Vapor Compression Cooling in Residential Applications," *Journal of Solar Energy Engineering*, Vol. 106, pp. 252–260.

Ketelaars, A. A. J., Coumans, W. J., and Kerkhof, P. J. A. M., 1995, "Drying Kenetics: A Comparison of Diffusion Coefficints from Moisture Concentration Profiles and Drying Curves," *Chemical Engineering Science*, Vol. 50, No. 7, pp. 1187–1191.

Kettleborough, C. and Waugaman, D., 1995, "Alternative Desiccant Cooling Cycle," *ASME Journal of Solar Energy Engineering*, Vol. 117, No. 3, pp. 251–255.

Khan, A. R., Baker, N. S., and Wardle, A. P., 1988, "The Dynamic Characteristcis of a Countercurrent Plate Heat Exhchanger," *International Journal of Heat and Mass Transfer*, Vol. 31, No. 6, pp. 1269–1278.

Kudela, V., 1984, *Encyclopedia of Polymer Science and Engineering*, Vol. 7, Chap. Hydrogels, John Wiley & Sons, New York, NY, pp. 783–807.

Kulicke, W. M., Nottlemann, H., Aggour, Y. A., and Elsabee, M. Z., 1989, "Preperation, Characterization, and Rehological Behavior of Water-Swellable Plotmers," in: *Proceeding of the ACS Division of Polymeric Materials Science and Engineeering*, Vol. 61, pp. 393–397.

Lakshmanan, C. C. and Potter, O. E., 1990, "Dynamic Simulation of Plate Heat Exchangers," *Internation Journal of Heat and Mass Transfer*, Vol. 33, No. 5, pp. 995–1002.

Lomax, R. G., 1990, "Hydrophilic Polyurethane Coatings," *Journal of Coated Fabrics*, Vol. 20, pp. 88–107.

Luikov, A. V., Shashkov, A. G., Vasiliev, L. L., and Fraiman, Y. E., 1968, "Thermal Conductivity of Porous Systems," *International Journal of Heat and Mass Transfer*, Vol. 11, pp. 117–140.

Masuda, K. and Iwata, H., 1990, "Dewatering of Particulate Materials Utilizing Highly Water-Absorptive Polymers," *Powder Technology*, Vol. 63, pp. 113–119.

Mendizabal, E., Espinoza, D., Casteneda, A., Puig, J., Katime, I., and Velada, J., 1995, "Improved Water Absorbing Acrylamide Hydrogels with Itaconic Esters," in: *Annual Technical Conference-ANTEC, Conference Proceedings*, Vol. 2, Society of Plastics Engineers, Brookfield, CT, pp. 1960–1963.

Miyata, N., Yokoyama, M., and Sakata, I., 1995, "Properties of Highly Water-Absorptive Polymers," *Journal of Applied Polymer Science*, Vol. 55, pp. 201–208.

Moffat, R. J., 1988, "Describing the Uncertanties in Experimental Results," *Experimental Thermal and Fluid Science*, Vol. 1, pp. 3–17.

Otterbein, R. T., 1995, "Theory for Heat Exchangers with Liquid-Desiccant-Wetted Surfaces," in: *Proceeding of the 1995 ASHRAE Annual Meeting*, Chicago, IL, pp. 317–325.

Passiniemi, P., 1995, "Simple Method for Determining water Diffusion Coefficient in Conducting Polymers," *Synthetic Metals*, Vol. 69, No. 1-3, pp. 685–686.

Pó, R., 1994, "Water-Absorbent Polymers: A Patent Survey," *Journal of Macromolecular Science - Review in Macromolecular Chemistry and Physics*, Vol. C34(4), pp. 607–662.

Rao, R., Kumari, H. S., and Raju, K. S., 1995, "Moisture Diffusion Behavior of T300-914C Laminates," *Journal or Reinforced Plastics and Composites*, Vol. 14, No. 5, pp. 513–522.

Sarwar, M. and Majumdar, P., 1995, "Thermal Conductivity of Wet Composite Porous Media," *Heat Recovery Systems and CHP*, Vol. 15, No. 4, pp. 369–381.

Sharifi, F., Narandji, M. R. G., and Mehravaran, K., 1995, "Dynamic Simulation of Plate Heat Exchangers," *International Communications in Heat and MAss Transfer*, Vol. 22, No. 2, pp. 213–225.

Zheng, W. and Worek, W. M., 1993, "Numerical Simulation of Combined Heat and Mass Transfer Processes in a Rotary Dehumidifier," *Numerical Heat Transfer*, Vol. 23, pp. 211–232.

Zheng, W., Worek, W. M., and Novosel, D., 1995, "Performance Optimization of Rotary Dehumidifiers," *Journal of Solar Energy Engineering*, Vol. 117, pp. 40–44.

Appendix A

Evaluation of Heat, Mass, and Structural Properties of Polyvinyl Alcohol

A.1 Evaluating the Material Mass Transfer Coefficient, K_y

The mass transfer coefficient, a term analogous to the heat transfer coefficient h, was implemented into every model presented in this document. This term controls the amount of moisture transferred to the surface of the desiccant material. It is an expression of the material moisture uptake rate as a function of the existing material moisture content. A description of the steps taken to determine K_y from the moisture uptake experimental data is outlined below.

1. Determine the initial dry weight

2. Using the mass uptake data and the initial dry weight, calculate the amount of moisture for each recorded data point

3. Find the amount of moisture per unit area by dividing the value from the previous step by the surface area of the sample

4. Plot this information versus time and generate a line fit to the data

5. Take the derivative of the line fit to obtain information regarding how the uptake per

area changes with time

6. Determine numerical values for this derivative expression by inserting time data into the expression from the above step

7. Plot these data points against the amount of moisture in the material at the corresponding time steps

8. Generate another line fit through this data to obtain an expression for the mass transfer coefficient, the expression of how the moisture uptake rate varies with material moisture content

A.2 Evaluating the System Heat Transfer Coefficient, h

Heat transfer coefficients were determined mathematically using the traditional method of evaluating effective airflow diameter, polyvinyl alcohol material conductivity, and effective Nusselt number.

For both the rotary wheel total enthalpy exchanger and fixed plate total enthalpy exchanger models, the heat transfer coefficient was estimated using an equation from Holman (1981), Eq. A.1, h was calculated from information like channel diameter and the conductivity of air. When solving for h in the rotary wheel total enthalpy exchanger models, the effective Nusselt number was evaluated by assuming laminar flow through a tube. Nu_d for the fixed plate total enthalpy exchanger model was calculated by assuming flow through a rectangular duct with a small aspect ratio.

$$Nu_d = \frac{hd_o}{k_{da}} \tag{A.1}$$

A.3 Evaluating the Material Mass Diffusivity, D_{ab}

Fundamentally, diffusion is based on Fick's first law, Eq. A.2, which describes the macroscopic transport of molecules by a concentration gradient.

$$\frac{\partial W}{\partial t} = D_{ab}\frac{\partial^2 W}{\partial y^2} \tag{A.2}$$

An effective mass diffusion coefficient, D_{ab}, was calculated from an semi-infinite solution using the data from the environmental chamber. It is assumed that initially the concentration in the film is zero, the surface concentration is constant, and the operating times of interest are short enough that the material can be modeled as a semi-infinite plate. The semi-infinite equation is represented as

$$W = W_{surface}[1 - erf(\frac{x}{2\sqrt{D_{ab}t}})]. \tag{A.3}$$

A.4 Evaluating the Material Thermal Diffusivity, α

Thermal diffusivity was determined within the MSE Department by using the laser flash thermal diffusivity technique. The laser flash method is a transient technique for measuring the thermal diffusivity of materials. The front surface of the sample is irradiated with a pulse of energy from a high-power laser source. An infra-red detector measures the temperature change of the sample's back surface as a function of time. The thermal diffusivity of the material is obtained from the rise time of the measured temperature curve.

A.5 Evaluating the Material Specific Heat, C_d

The desiccant specific heat was determined with the use of a differential scanning calorimeter (DSC). This material characterization method yields peaks relating to endothermic and exothermic transitions in addition to showing changes in heat capacity. This method can also give quantitative information relating to enthalpic changes within the polymeric material. A DSC performs measurements by comparing temperature differences between a sample material and a reference material, typically powdered alumina. Employing a servo system to supply energy at a varying rate to both the sample and the reference, the DSC records the energy that must be supplied against the average temperature in order to keep the sample and the reference at equal temperatures. When an exothermic or endothermic change occurs in a sample material, power (energy) is applied or removed to one or both calorimeters to compensate for the energy change in the sample. Using plotted output, specific heat can be determined by measuring the energy difference associated with the slope before and after the material's glass transition temperature has taken place. The heat

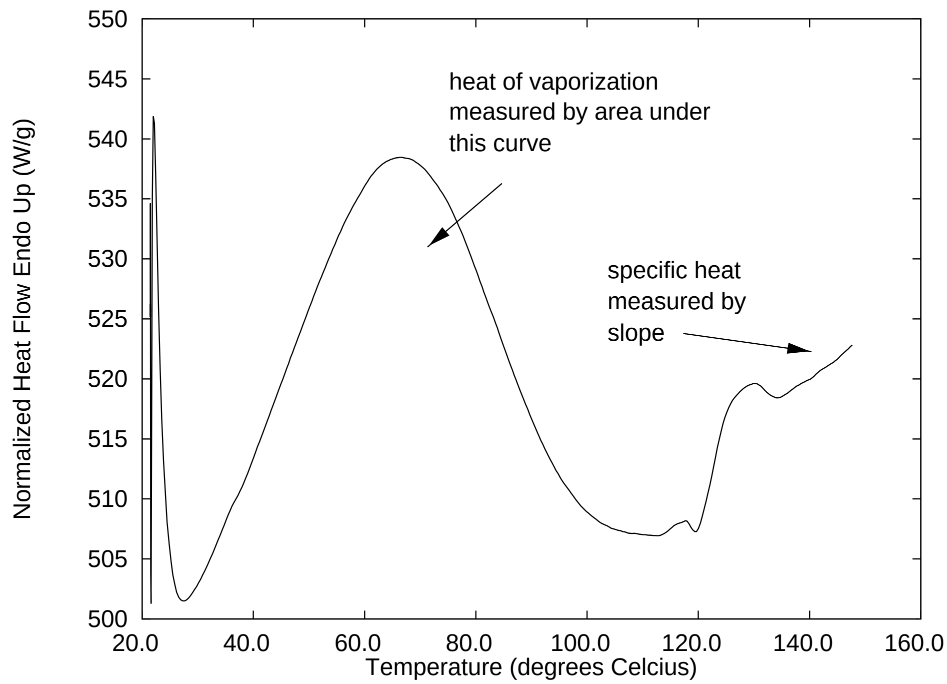

Figure A.1 Sample DSC Curve: Polyvinyl Alcohol Foam

of vaporization can be discovered by analyzing the area under the exothermic peak of the energy curve. A sample DSC curve, used to estimate the heat of vaporization and specific heat of polyvinyl alcohol foam, is included in Fig. A.1.

A.6 Evaluating the Desiccant Coating Thickness, a_d

The desiccant thickness played a role in determining the amount of desiccant used in the dehumidifier. As shown in Chapter 5, the amount of desiccant is directly related to the overall performance of the total enthalpy exchanger. Desiccant thicknesses for the polyvinyl alcohol foam applications were directly measured with micrometers. However, the desiccant thicknesses on the film-coated applications were too thin to measured by conventional methods. The procedure used to determine the desiccant film-coating thicknesses is outlined below.

1. Determine mass of desiccant coated on substrate
2. Measure coated surface area
3. Using desiccant coating density information, back out coating thickness value (this requires an assumption that the coating was uniform over the substrate surface)

Appendix B

Sample Data Obtained from the Environmental Chamber

To assure more accurate property estimates than those gathered from the desiccator box described in Chapter 3, more detailed adsorption rate/equilibrium moisture content experiments were run in the Cole-Parmer Automatically Controlled Environmental Chamber ®. Temperature and humidity conditions within the chamber were kept uniform throughout each experiment. A computer program was written to record data resulting from the moisture uptake experiments in the environmental chamber. Using an electronic balance, the program recorded the changing weight as a function of time. Sample moisture uptake data obtained from the environmental chamber are shown below.

The plots in Fig. B.1 represent the types of curves from which the material properties W_{max}, D_{ab}, and K_y were evaluated. For each material, several moisture uptake tests were conducted at various environmental conditions. Material properties represented as constants in the mathematical models were determined by averaging material properties observed at different conditions. Consequently, material property values represented in Fig. B.1 may not agree exactly with the material property values represented in Tables 3.2 and 5.6.

Figure B.1 can be used, however, to compare the nature of each material's moisture uptake slope with the slopes generated by the other materials. As outlined in Appendix A, moisture uptake slopes were used to determine both material diffusivity, D_{ab}, and the mass transfer coefficient, K_y. From this particular graph it is difficult to determine whether the polyvinyl alcohol foam or the LaRoche paper has the steeper uptake slope. However it can

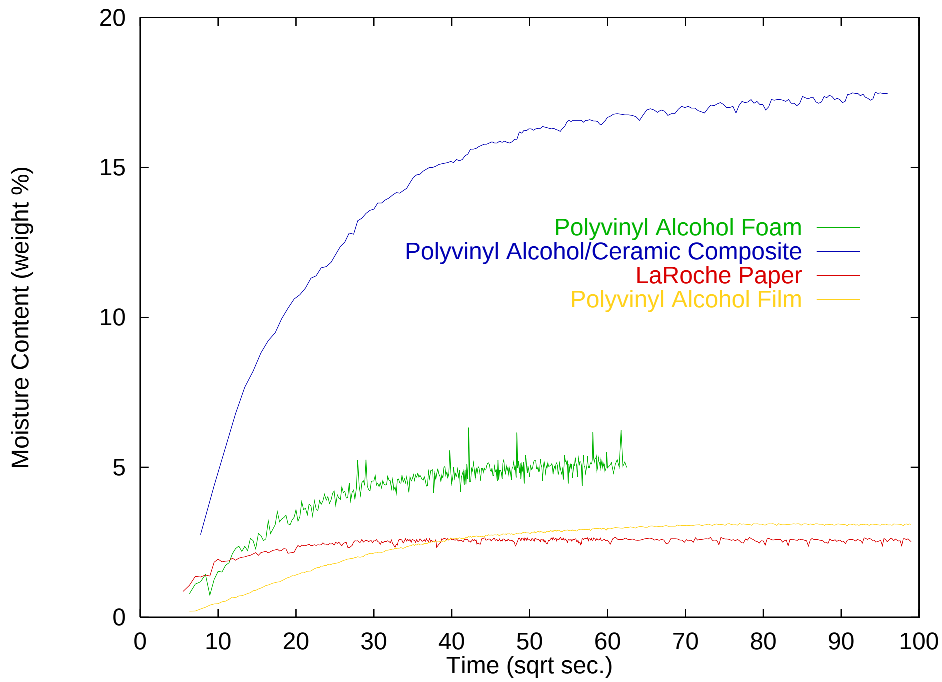

Figure B.1 Sample Sorption Curves: 70-75 °F and 95-97 % Relative Humidity

be determined that under the conditions of 70-75 °F and 95-97 % relative humidity that polyvinyl alcohol foam is capable of holding more water per dry mass than the LaRoche paper. It can also be observed that, under the sample conditions, the equilibrium moisture content of the polyvinyl alcohol/ceramic composite was the greatest of all the materials studied. The moisture uptake slope of the composite material also appears to be steeper than the other materials.

Under the sample conditions, the polyvinyl alcohol film absorbs moisture from the air at a much slower rate than the other materials. This behavior directly reflects on the ability of polyvinyl alcohol film to both accept and diffuse moisture, resulting in smaller values of K_y and D_{ab}. These two material properties were significantly smaller for polyvinyl alcohol film than any other material studied during this research. This observation helps explain why, regardless of the rotary wheel total enthalpy exchanger design, the latent performance of the polyvinyl alcohol film-coated rotary wheel total enthalpy exchanger was poor.

Appendix C

Polyvinyl Alcohol Film Coated Rotary Wheel Total Enthalpy Exchanger Performance Estimation Code

```
      PROGRAM ABSORB

**********************************************************************
*      Performance Estimates of A
*  Rotary Wheel Total Enthalpy Exchanger
*  Coated with a Non-Porous (Absorbing) Polymer Desiccant
*                                 by
*                           Christie Staton
*  1997
*
**********************************************************************
*
*
*    This program estimates the sensible, latent, and total performance
* efficiencies of a Rotary Wheel Total Enthalpy Exchanger coated with a
* non-porous (absorbing) polymer desiccant. Exit conditions for both the
* supply (process) and exhaust (regenerative) air streams are calculated
* along with the desiccant bed conditions of the front and back faces of
* the wheel.  A solution is obtained via an integration over half of the
* wheel face.
*
*    A choice of three desiccant channel geometries, tridiagonal,
* hexagonal, or sinusoidal can be implemented during the program run.  In
* addition to the channel geometry choices and rotational time inputted by
* the program user, the program requires six input data files.  These files
* are identified and described in detail in the subroutine PROPERTY.
*
*-----------------------------------------------------------------------
*    We begin with the MAIN DRIVER of the program.  This is the portion of
* the code form which all subroutines are called.  The MAIN DRIVER organizes
* the operation of the subroutines in such a manner as to get the
* performance estimation accomplished in the most efficient way.
```

```
*----------------------------------------------------------------------

      implicit none
*-----------------------------------------------------------------------
*   Declare all variables that are used in the MAIN DRIVER and elsewhere.
* Variables declare here are used throughout the program; they are used in
* several different subroutines.
*-----------------------------------------------------------------------

      integer z,i,j,k,l,nx,cyclenum,NMAX1,NMAX2
      real g,f,perw,perd,n,tautot,timetot
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Hv,h,velpro,
     $  velreg,Wmax,Patm
      double precision hgpres,hgprev,hgpro,hgregen,hgtot
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring
      double precision Twpros,Wpros,Yapros,Tapros,
     $   Twregs,Wregs,Yaregs,Taregs,Twtotpro,Wtotpro,Yatotpro,
     $   Tatotpro,Tatot,Ta,Ya,Yatot,Twtot,Wtot,Yatotreg,Tatotreg,
     $   Twtotreg,Wtotreg
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the MAIN DRIVER and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)
      parameter (NMAX2=2500)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $   known4(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),
     $   Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Yatotpro(NMAX2),
     $   Tatotpro(NMAX2),Twtotreg(NMAX2),Wtotreg(NMAX2),
     $   Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to various subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/heat/ hgpres,hgprev,hgpro,hgregen,hgtot
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
```

```
      common/coefficients/B3,B4,B5,B6,B7,B8
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
      common/solution/ Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot
*----------------------------------------------------------------------
*    We begin with the body of the MAIN DRIVER of the program, the portion
*  of the program from which all subroutines are called.
*    First, the subroutine PROPERTY is called. This subroutine reads
* user input data dealing with plate geometry and air stream conditions.
* The input data files containing all density, specific heat, boundary
* conditions of the supply and exhaust air, initial conditions of the
* desiccant bed, plate geometry, and air property information are read in
* the section of the program.
*-----------------------------------------------------------------------

      call PROPERTY
*-----------------------------------------------------------------------
*    At this point the variable that determines how long one wheel rotation
* will be is determined.  Variable tautot defines the total elapsed time
* (in seconds) for one full rotation modeled by the program.  This is
* determined by converting rotational speed (in RPM),timetot, to
* rotational time.  Next, the amount of time each side of the wheel spends
* in each air stream (assuming the wheel is divided in half) is calculated.
*-----------------------------------------------------------------------

      timetot=40.0
      tautot=60.0/timetot
      tautot=tautot/2.0
*-----------------------------------------------------------------------
*    Using the time step size defined in PROPERTY, determine how many time
* steps will occur in half a wheel rotation.  This number is defined by l.
*-----------------------------------------------------------------------

      l=0
      do g=REAL(dtau),tautot,REAL(dtau)
         l=l+1
      end do
*-----------------------------------------------------------------------
*    After determining the total time to be modeled as well as the time step
* to be used in the analysis, a time loop is introduced.  This loop is based
* upon the number of cycles to be analyzed.  This term is indicated in one
* of the input files read in subroutine PROPERTY.  Keep in mind that
* this model is analyzed by stepping along in distance within a specified
* space in time.
*-----------------------------------------------------------------------

      do j=1,cyclenum
*-----------------------------------------------------------------------
*    Next, variables representing total air and desiccant material exiting
* conditions for both the supply and exhaust side are initialized.  At every
* time step, these conditions are updated, and therefore must be reset as each
* time step is updated.
*-----------------------------------------------------------------------

 Tatot=0.d0
 Yatot=0.d0
        Ta=0.d0
        Ya=0.d0
 Tatotpro(j)=0.d0
 Yatotpro(j)=0.d0
 Twtotpro(j)=0.d0
 Wtotpro(j)=0.d0
        Tatotreg(j)=0.d0
 Yatotreg(j)=0.d0
 Twtotreg(j)=0.d0
 Wtotreg(j)=0.d0
```

```
        Tapros=0.d0
        Yapros=0.d0
        Twpros=0.d0
        Wpros=0.d0
        Taregs=0.d0
        Yaregs=0.d0
        Twregs=0.d0
        Wregs=0.d0

        z=0
        call BNDYPRO
        call COEFPRO

        do g=REAL(dtau),tautot,REAL(dtau)
           z=z+1
           if (z .eq. INT(tautot/(2.*REAL(dtau))) ) then
              call APPROXIMATE
           end if
           call PROCESS
           call XFER
           Tapros=Tapros + Tapres(nx)
           Yapros=Yapros + Yapres(nx)
           Twpros=Twpros + Twpres(nx)
           Wpros=Wpros + Wpres(nx)
        enddo

         Tatotpro(j)=Tapros/l
 Yatotpro(j)=Yapros/l
 Twtotpro(j)=Twpros/l
 Wtotpro(j)=Wpros/l
 Tatot=Tatotpro(j)
 Yatot=Yatotpro(j)
         Ta=Tatot
         Ya=Yatot
         Twtot=Twtotpro(j)
 Wtot=Wtotpro(j)

        z=0
        call BNDYREGEN
        call COEFREGEN

        do g=REAL(dtau),tautot,REAL(dtau)
           z=z+1
           if (z .eq. INT(tautot/(2.*REAL(dtau))) ) then
             call APPROXIMATE
           end if
           call REGENERATION
           call XFER
            Taregs=Taregs + Tapres(1)
            Yaregs=Yaregs + Yapres(1)
            Twregs=Twregs + Twpres(1)
            Wregs=Wregs + Wpres(1)
        enddo

        Tatotreg(j)=Taregs/l
Yatotreg(j)=Yaregs/l
Twtotreg(j)=Twregs/l
Wtotreg(j)=Wregs/l

     enddo

     call EFFICIENCY


     call OUTPUT

     END
```

```
**********************************************************************
      SUBROUTINE PROPERTY
**********************************************************************
*
*    The primary function of this subroutine is read in data dealing with
* channel geometry, efficiency evaluation, density, specific heat, boundary
* conditions of the supply and exhaust air, initial conditions of the
* desiccant bed, wheel geometry, and air property, and wheel performance
* information.  All of this information is read via a series of data input
* files.  Once the proper information is  obtained,  geometric information
* leading to the solution of performance efficiencies is calculated.
*----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to PROPERTY and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine PROPERTY.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      real g,f,perw,perd,n,tautot
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision pi,gravity,Dh,Red,kviscair,friction,tothead,
     $   spgravity
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the PROPERTY and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $   Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
```

```
*----------------------------------------------------------------------
*    Read in the input data files
*  performance.dat:
*   cyclenum=   # of cycles to be modeled
*  spheat.dat:
*   Cwr=  reference specific heat (sp. heat of water)
*   Cda=  specific heat of dry air
*   Cv=   specific heat of water vapor
*   Cd=  specific heat of desiccant
*   Cs=  specific heat of support substrate
*   Cw=  specific heat of wheel
*   spgravity=  specific gracity of water
*  density.dat:
*   rhod=  density of desiccant
*   rhos=  density of support substrate
*   rhoevery=  density of wheel
*   rhoair=  density of air
*   kviscair=  kinematic viscosity of air
*  design.dat:
*   Do=  outer diameter
*   Di=  inner diameter
*   wheeldepth=  depth of dehumidifier
*   thickd=  thickness of desiccant
*   thicks=  thickness of support substrate
*  nondim.dat:
*   Wmax=  maximum moisture capacity of desiccant
*   Patm=  atmospheric pressure
*   h=  heat transfer coefficient
*   velpro=  velocity of supply air
*   velreg=  velocity of exhaust air
*  boundary.dat:
*   Twic=  initial temperature condition of the dehumidifier
*   Wic=  initial moisture content condition of the dehumidifier
*   Tabcpro=  air temperature boundary condition for supply side
*   Yabcpro=  air moisture content boundary condition for supply side
*   Tabcregen=  air temperature boundary condition for exhaust side
*   Yabcregen=  air moisture content boundary condition for exhaust side
*----------------------------------------------------------------------

      pi=3.141592654d0
      gravity=9.81d0

      open(1,file='performance.dat')
      read(1,*) cyclenum
      open(2,file='spheat.dat')
      read(2,*) Cwr,Cda,Cv,Cd,Cs,Cw,spgravity
      open(3,file='density.dat')
      read(3,*) rhod,rhos,rhoevery,rhoair,kviscair
      open(4,file='design.dat')
      read(4,*) Do,Di,wheeldepth,thickd,thicks
      open(7,file='nondim.dat')
      read(7,*) Wmax,Patm,h,velpro,velreg
      open(8,file='boundary.dat')
      read(8,*) Twic,Wic,Tabcpro,Yabcpro,Tabcregen,Yabcregen
*----------------------------------------------------------------------

100   print *, 'Which channel geometry would you like to use?'
      print *
      print *, 'Enter the number that corresponds to the'
      print *, 'correct cross-sectional shape:'
      print *
      print *, '1 = equilateral triangular'
      print *, '2 = sinusoidal'
      print *, '3 = regular hexagonal'
      read *, response
      print *
*----------------------------------------------------------------------
```

```
*    Check to make sure the inputs are valid:
*-----------------------------------------------------------------------

      if (response.ne.1 .and. response.ne.2 .and. response.ne.3) then
      print *, 'Geometric input parameter is invalid, please try again.'
      print *, 'The input should be in the form of a 1, 2, or 3.'
      go to 100
      end if
*-----------------------------------------------------------------------
*    If the triangular channel geometry is chosen, assign proper values to
* the various area inputs.  Base, Areatot, Areaair, and Areas define
* channel width, total channel area, air area in the channel, and
* structural area in the channel, respectively.
*-----------------------------------------------------------------------

      if (response.eq.1) then
 print *
 print *, 'Please enter the channel height (meters):'
        read *, height
        height=DBLE(height)
 base=2.d0*height/DTAN(pi/3.d0)
        perimeter=3.d0*base
 Areatot=base*height/2.d0
 Areaair=(height-(2.d0*thickd+thicks))*(height-(2.d0*thickd+
     $  thicks))/DTAN(pi/3.d0)
 Areas=(height-thicks)*(height-thicks)/DTAN(pi/3.d0)
        Dh=4.d0*Areaair/(6.d0*(height-(2.d0*thickd+thicks))/
     $      DTAN(pi/3.d0))
        Red=(velpro*Dh)/kviscair
        friction=64.d0/Red
        tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
     $        wheeldepth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
     $        velpro**2/(2.d0*gravity))
        pressuredrop=rhoair*gravity*tothead
        pressuredrop=pressuredrop/spgravity
 go to 120
      end if
*-----------------------------------------------------------------------
*   If the sinusoidal channel geometry is chosen, assign proper values to
* the various area inputs.  Base, Areatot, Areaair, and Areas define
* channel width, total channel area, air area in the channel, and
* structural area in the channel, respectively.
*-----------------------------------------------------------------------

      if (response.eq.2) then
 print *
 print *, 'Please enter the channel wave amplitude (meters):'
 read *, height
        height=DBLE(height)
 print *
 print *, 'Please enter the channel wave length (meters):'
 read *, base
 base=2.d0*height/DTAN(pi/3.d0)
        perimeter=3.d0*base
 Areatot=base*height/2.d0
 Areaair=(height-(2.d0*thickd+thicks))*(height-(2.d0*thickd+
     $  thicks))/DTAN(pi/3.d0)
 Areas=(height-thicks)*(height-thicks)/DTAN(pi/3.d0)
        Dh=4.d0*Areaair/(6.d0*(height-(2.d0*thickd+thicks))/
     $      DTAN(pi/3.d0))
        Red=(velpro*Dh)/kviscair
        friction=64.d0/Red
        tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
     $        wheeldepth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
     $        velpro**2/(2.d0*gravity))
        pressuredrop=rhoair*gravity*tothead
        pressuredrop=pressuredrop/spgravity
 go to 120
```

```
      end if
*-----------------------------------------------------------------------
*   If the hexagonal channel geometry is chosen, assign proper values to
* the various area inputs.  Base, Areatot, Areaair, and Areas define
* channel width, total channel area, air area in the channel, and
* structural area in the channel, respectively.
*-----------------------------------------------------------------------

      if (response.eq.3) then
 print *
 print *, 'Please enter the distance between parallel sides
     $    (meters):'
 read *, height
         height=DBLE(height)
 base=3.d0*height/DTAN(pi/3.d0)
 Areatot=(1.5d0*height**2)/DTAN(pi/3.d0)
 Areaair=(1.5d0*(height-(2.d0*thickd+thicks))**2)/
     $   DTAN(pi/3.d0)
 Areas= (1.5d0*(height-thicks)**2)/DTAN(pi/3.d0)
         Dh=4.d0*Areaair/(6.d0*base)
         Red=(velpro*Dh)/kviscair
         friction=64.d0/Red
         tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
     $        wheeldepth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
     $        velpro**2/(2.d0*gravity))
         pressuredrop=rhoair*gravity*tothead
         pressuredrop=pressuredrop/spgravity
 go to 120
      end if

*-----------------------------------------------------------------------
*    Determine actual length of elements; nx = number of nodes along
* channel and dx = distance between nodes
*-----------------------------------------------------------------------

120   nx=21
      dx=wheeldepth/dble(nx-1)
*-----------------------------------------------------------------------
*    Calculate areas, volumes, weights, and mass fractions needed for
* numerical solution of performance estimates,
*-----------------------------------------------------------------------
*    Area of the desiccant(per channel):
*    Mass of desiccant within each channel:
*-----------------------------------------------------------------------

      Aread=Areas - Areaair
      Massdc=Aread*wheeldepth*rhod
*-----------------------------------------------------------------------
*    Percentage of dehumidifier in wheel, and desiccant in dehumidifier
*-----------------------------------------------------------------------

      perw=REAL((Areatot-Areaair)/Areatot)
      perd=REAL(Aread/(Areatot-Areaair))
*-----------------------------------------------------------------------
*    Tot volume of wheel, number of channels in wheel, mass of dehumidifier
*-----------------------------------------------------------------------

      Voltot=3.14d0*(Do**2-Di**2)*wheeldepth/4.d0
      Volw=Voltot*DBLE(perw)
      n=int((Voltot-Volw)/(Areaair*wheeldepth))
      Massw=Volw*(rhod*DBLE(perd) + rhos*(1.d0-DBLE(perd)))
      f=REAL(Voltot*DBLE(perw)*DBLE(perd)*rhod/Massw)
*-----------------------------------------------------------------------
*    Determine length of time steps to be used in the solution
*-----------------------------------------------------------------------

      dtau=0.01d0
*-----------------------------------------------------------------------
```

```
*  Assign initial values to each node along the channel depth
*-----------------------------------------------------------------------

      do i=1,nx
         Taring(i)=Tabcpro
         Yaring(i)=Yabcpro
         Twring(i)=Twic
 Wring(i)=Wic
      end do
         Taring(1)=Tabcpro
         Yaring(1)=Yabcpro
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

***********************************************************************
      SUBROUTINE APPROXIMATE
**********************************************************************

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring


      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring

      parameter (NMAX1=100)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)

      do i=1,nx
         Twring(i)=Twpres(i)
         Wring(i)=Wpres(i)
         Yaring(i)=Yapres(i)
         Taring(i)=Tapres(i)
         Yamaxring(i)=Yamax(i)

      end do

      return
      end

***********************************************************************
      SUBROUTINE BNDYPRO
***********************************************************************

c
c----------------------------------------------------------------------


      common/counters/ g,i,j,k,l,nx,cyclenum
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
```

```
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring


      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring

      parameter (NMAX1=100)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)

c________________________________________________________________________


c  Define initial conditions and boundary conditions at all nodes

      do i=1,nx
         Tapres(i)=Taring(i)
         Yapres(i)=Yaring(i)
         Twpres(i)=Twring(i)
         Wpres(i)=Wring(i)
         Yamax(i)=Yamaxring(i)
      end do
      Yapres(1)=Yabcpro
      Tapres(1)=Tabcpro
      return
      end




***********************************************************************
      SUBROUTINE COEFPRO
***********************************************************************
*_______________________________________________________________________
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to COEFPRO and various other subroutines easier.
*_______________________________________________________________________

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*_______________________________________________________________________
*   Declare all variables that are used in the subroutine COEFPRO.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*_______________________________________________________________________

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision rhod,rhos,rhoair,rhoevery,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $   Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*_______________________________________________________________________
*     Initialize parameters used in the dimensioning of all arrays used in
```

```
* the COEFPRO and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $   Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
*-----------------------------------------------------------------------
*   Assign air conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*-----------------------------------------------------------------------
*   Step along the x direction, along the plate depth
*-----------------------------------------------------------------------

      do i=1,nx
         Yaprev(i)=Yapres(i)
        Taprev(i)=Tapres(i)
         Twprev(i)=Twpres(i)
         Wprev(i)=Wpres(i)
         TempK(i)=2.7315d+2 + Taprev(i)
         u(i)=(6.4727d+2 - TempK(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempK(i))*
     $        (6.4727d+2 - TempK(i)))* (6.4727d+2 - TempK(i)))/
     $        (TempK(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempK(i))))
         PwsT(i)=2.18167d+2/(10.d0 **u(i))
 Yamaxprev(i)=6.22d-1*PwsT(i)/Patm
      end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

***********************************************************************
      SUBROUTINE PROCESS
***********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to PROCESS and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine PROCESS.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------
```

```
      integer i,j,k,l,nx,cyclenum,NMAX1
      real g,f,perw,perd,n,tautot
      double precision Ky
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the PROCESS and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension Ky(NMAX1)
      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $   known4(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
*-----------------------------------------------------------------------
*    Stepping along in x, the plate depth, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every iteration/time step.
*-----------------------------------------------------------------------

      do i=1,nx

         if (Wprev(i) .gt. Wmax) then
             Wprev(i)=Wmax
         end if

         TempK(i)=2.7313d+2 + Taprev(i)
         TempK(1)=2.7315d+2 + Tabcpro
         u(i)=(6.4727d+2 - TempK(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempK(i))*
     $        (6.4727d+2 - TempK(i)))* (6.4727d+2 - TempK(i)))/
     $        (TempK(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempK(i))))
         PwsT(i)=2.18167d+2/(10.d0 **u(i))
 Yamax(i)=6.22d-1*PwsT(i)/Patm
*-----------------------------------------------------------------------
*    Calculate the mass transfer coefficient, a moisture content dependent
* material property.
*-----------------------------------------------------------------------

         Ky(i)=10.d0*(6.d-7 - 1.d-5*(1.d+3*Wprev(i)*Massdc))
*-----------------------------------------------------------------------
*    Calculate the heat of vaporization, a temperature dependent
* material property.
*-----------------------------------------------------------------------

         Hv=2502.d0+Cv*Taprev(i)
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
```

```
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------
         B3(i)=Ky(i)/(thickd*rhod*DBLE(f))
         B4(i)=Hv*Ky(i)/(thickd*rhoevery*
     $         (DBLE(1.-f)*Cs+DBLE(f)*(Cd+Wprev(i)*Cwr)))
         B5(i)=h/(thickd*rhoevery*(DBLE(1.-f)*
     $         Cs+DBLE(f)*(Cd+Wprev(i)*Cwr)))
         B6(i)=perimeter*Ky(i)/(rhoair*velpro*Areaair)
         B7(i)=perimeter*Ky(i)*Hv/(rhoair*velpro*Areaair*(Cda+
     $         Yaprev(i)*Cv))
         B8(i)=h*perimeter/(rhoair*velpro*Areaair*(Cda+
     $         Yaprev(i)*Cv))
c   B3(i)=Cwr*Ky(i)/(h*DBLE(f))
c         B4(i)=Cwr/(DBLE(1.-f)*Cs+DBLE(f)*(Cd+Cwr*Wprev(i)))
c         B5(i)=Hv/(Cda*Le(i)*B4(i))
c         B6(i)=Cda*Ky(i)/h
c         B7(i)=0.d0
c         B8(i)=1.d0/(1.d0+Cv*Yaprev(i)/Cda)
       end do
*-----------------------------------------------------------------------
*    This closes the coefficient determination phase of the subroutine.  We
* now move on to the nodal solution of the supply-side phase.
*-----------------------------------------------------------------------
************************************************************************
*-----------------------------------------------------------------------
*  Maintain supply-side air boundary conditions
*-----------------------------------------------------------------------

      Yapres(1)=Yabcpro
      Tapres(1)=Tabcpro
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for first, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known3(1)=(1.d0-dtau*B3(1)/(2.d0*Wmax))*Wprev(1) + (B3(1)*
     $             dtau/(2.d0*Yamaxprev(1)))*Yaprev(1)

      known4(1)=Twprev(1)+dtau*B4(1)/2.d0*(Yaprev(1)/Yamaxprev(1)-
     $             Wprev(1)/Wmax) + B5(1)*dtau/2.d0*(Taprev(1) -
     $             Twprev(1))
*-----------------------------------------------------------------------
*    Solve for desiccant conditions on the supply-side surface for the
* first, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(1)=(known3(1) + dtau*B3(1)/2.d0 *Yapres(1)/Yamax(1))/
     $         (1.d0 + dtau/2.d0*B3(1)/Wmax)

      Twpres(1)=(known4(1) + dtau/2.d0*B4(1)*(Yapres(1)/Yamax(1) -
     $          Wpres(1)/Wmax) + dtau/2.d0*B5(1)*Tapres(1))/(1.d0 +
     $          dtau/2.d0*B5(1))
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for all interior
*  nodes.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      do i=2,nx-1

         known1(i)=(B6(i)*dx/(2.d0*Wmax))*Wpres(i-1) +
     $    (1.d0-B6(i)*dx/ (2.d0*Yamax(i-1)))*Yapres(i-1)

         known2(i)=B7(i)*dx/2.d0*(Wpres(i-1)/Wmax - Yapres(i-1)/
     $             Yamax(i-1)) + B8(i)*dx/2.d0*(Twpres(i-1) -
     $            Tapres(i-1)) +Tapres(i-1)
```

```
 known3(i)=(1.d0-dtau*B3(i)/(2.d0*Wmax))*Wprev(i) + (B3(i)*
    $             dtau/(2.d0*Yamaxprev(i)))*Yaprev(i)

        known4(i)=Twprev(i)+dtau*B4(i)/2.d0*(Yaprev(i)/Yamaxprev(i)-
    $             Wprev(i)/Wmax) + B5(i)*dtau/2.d0*(Taprev(i) -
    $             Twprev(i))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the supply-side
* surface for all interior nodes
*-----------------------------------------------------------------------

        Wpres(i)=(known3(i)+dtau/2.d0*B3(i)/Yamax(i)*(known1(i)/
    $            (1.d0+dx/2.d0*B6(i)/Yamax(i))))/(1.d0 + dtau*B3(i)/
    $            (2.d0*Wmax) - dtau/2.d0*B3(i)/Yamax(i)*dx/2.d0
    $            *B6(i)/Wmax/(1.d0+dx/2.d0*B6(i)/Yamax(i)))

        Yapres(i)=(known1(i) + dx/2.d0*B6(i)/Wmax*Wpres(i))/
    $             (1.d0+dx/2.d0*B6(i)/Yamax(i))

        Twpres(i)=(known4(i) + dtau*B4(i)/2.d0*(Yapres(i)/Yamax(i) -
    $             Wpres(i)/Wmax) + dtau*B5(i)/(2.d0*(1.d0+B8(i)*
    $             dx/2.d0))*(known2(i) + B7(i)*dx/2.d0*(Wpres(i)/
    $             Wmax - Yapres(i)/Yamax(i))))/(1.d0 + dtau*B5(i)/
    $             2.d0 - dtau*B5(i)*(dx*B8(i)/2.d0)/(2.d0*(1.d0+
    $             B8(i)*dx/2.d0)))

        Tapres(i)=(known2(i) + B7(i)*dx/2.d0*(Wpres(i)/Wmax -
    $             Yapres(i)/Yamax(i)) + B8(i)*dx/2.d0*Twpres(i))/
    $             (1.d0+B8(i)*dx/2.d0)
      end do
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for last, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known1(nx)=(B6(nx)*dx/(4.d0*Wmax))*Wpres(nx-1) +
    $    (1.d0-B6(nx)*dx/ (4.d0*Yamax(nx)))*Yapres(nx-1)

      known2(nx)=B7(nx)*dx/4.d0*(Wpres(nx-1)/Wmax - Yapres(nx-1)/
    $             Yamax(nx-1)) + B8(nx)*dx/4.d0*(Twpres(nx-1) -
    $            Tapres(nx-1)) +Tapres(nx-1)

      known3(nx)=(1.d0-dtau*B3(nx)/(2.d0*Wmax))*Wprev(nx) + (B3(nx)*
    $             dtau/(2.d0*Yamaxprev(nx)))*Yaprev(nx)

      known4(nx)=Twprev(nx)+dtau*B4(nx)/2.d0*(Yaprev(nx)/Yamaxprev(nx)-
    $             Wprev(nx)/Wmax) + B5(nx)*dtau/2.d0*(Taprev(nx) -
    $             Twprev(nx))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream  conditions on the supply-side
* surface for the last, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(nx)=(known3(nx)+dtau/2.d0*B3(nx)/Yamaxprev(nx)*(known1(nx)/
    $          (1.d0+dx/2.d0*B6(nx)/Yamax(nx))))/(1.d0 + dtau*B3(nx)/
    $          (2.d0*Wmax) - dtau/2.d0*B3(nx)/Yamaxprev(nx)*dx/4.d0
    $          *B6(nx)/Wmax/(1.d0+dx/4.d0*B6(nx)/Yamax(nx)))


      Yapres(nx)=(known1(nx) + dx/4.d0*B6(nx)/Wmax*Wpres(nx))/
    $           (1.d0+dx/4.d0*B6(nx)/Yamax(nx))

      Twpres(nx)=(known4(nx) + dtau*B4(nx)/2.d0*(Yapres(nx)/Yamax(nx) -
    $             Wpres(nx)/Wmax) + dtau*B5(nx)/(2.d0*(1.d0+B8(nx)*
    $             dx/2.d0))*(known2(nx) + B7(nx)*dx/2.d0*(Wpres(nx)/
    $             Wmax - Yapres(nx)/Yamax(nx))))/(1.d0 + dtau*B5(nx)/
```

```
     $             2.d0 - dtau*B5(nx)*(dx*B8(nx)/2.d0)/(2.d0*(1.d0+
     $             B8(nx)*dx/2.d0)))


      Tapres(nx)=(known2(nx) + B7(nx)*dx/4.d0*(Wpres(nx)/Wmax -
     $             Yapres(nx)/Yamax(nx)) + B8(nx)*dx/4.d0*Twpres(nx))/
     $             (1.d0+B8(nx)*dx/4.d0)
*_________________________________________________________________________
*   Return to the MAIN DRIVER
*_________________________________________________________________________

      return
      end

************************************************************************
      SUBROUTINE BNDYREGEN
************************************************************************

c
c_______________________________________________________________________


      common/counters/ g,i,j,k,l,nx,cyclenum
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring

      parameter (NMAX1=100)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
c_______________________________________________________________________


c  Define initial conditions and boundary conditions at all nodes

      do i=1,nx
         Twpres(i)=Twring(i)
         Wpres(i)=Wring(i)
         Yapres(i)=Yaring(i)
         Tapres(i)=Taring(i)
         Yamax(i)=Yamaxring(i)
      end do

      Tapres(nx)=Tabcregen
      Yapres(nx)=Yabcregen

      return
      end




***********************************************************************
      SUBROUTINE COEFREGEN
***********************************************************************
```

```
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to COEFREGEN and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine COEFREGEN.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------
      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision rhod,rhos,rhoair,rhoevery,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $   Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the COEFREGEN and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
*-----------------------------------------------------------------------
*   Assign air conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*-----------------------------------------------------------------------
*   Step along the x direction, along the wheel depth
*-----------------------------------------------------------------------

      do i=nx,1,-1
         Yaprev(i)=Yapres(i)
         Taprev(i)=Tapres(i)
         Twprev(i)=Twpres(i)
 Wprev(i)=Wpres(i)
         TempK(i)=2.7315d+2 + Taprev(i)
         u(i)=(6.4727d+2 - TempK(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempK(i))*
     $        (6.4727d+2 - TempK(i)))* (6.4727d+2 - TempK(i)))/
     $        (TempK(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempK(i))))
         PwsT(i)=2.18167d+2/(10.d0 **u(i))
 Yamaxprev(i)=6.22d-1*PwsT(i)/Patm
      end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

***********************************************************************
      SUBROUTINE REGENERATION
***********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to REGENERATION and various other subroutines easier.
```

```
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine REGENERATION.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      real g,f,perw,perd,n,tautot
      double precision Ky
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the REGENERATION and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension Ky(NMAX1)
      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $   known4(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
*-----------------------------------------------------------------------
*    Stepping along in x, the plate depth, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every time step.
*-----------------------------------------------------------------------

      do i=nx,1,-1

         if (Wprev(i) .gt. Wmax) then
          Wprev(i)=Wmax
```

```
        end if

        TempK(i)=2.7315d+2 + Taprev(i)
        TempK(1)=2.7315d+2 + Tabcpro
        u(i)=(6.4727d+2 - TempK(i))  *(3.2437814d0 +
    $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempK(i))*
    $        (6.4727d+2 - TempK(i)))* (6.4727d+2 - TempK(i)))/
    $        (TempK(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempK(i))))
        PwsT(i)=2.18167d+2/(10.d0 **u(i))
 Yamax(i)=6.22d-1*PwsT(i)/Patm
*-----------------------------------------------------------------------
*    Calculate the mass transfer coefficient, a moisture content dependent
* material property.
*-----------------------------------------------------------------------

        Ky(i)=10.d0*(6.d-7 - 1.d-5*(1.d+3*Wprev(i)*Massdc))
*-----------------------------------------------------------------------
*    Calculate the heat of vaporization, a temperature dependent
* material property.
*-----------------------------------------------------------------------

        Hv=2502.d0+Cv*Taprev(i)
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------
        B3(i)=Ky(i)/(thickd*rhod*DBLE(f))
        B4(i)=Hv*Ky(i)/(thickd*rhoevery*
    $         (DBLE(1.-f)*Cs+DBLE(f)*(Cd+Wprev(i)*Cwr)))
        B5(i)=h/(thickd*rhoevery*(DBLE(1.-f)*
    $         Cs+DBLE(f)*(Cd+Wprev(i)*Cwr)))
        B6(i)=perimeter*Ky(i)/(rhoair*velpro*Areaair)
        B7(i)=perimeter*Ky(i)*Hv/(rhoair*velpro*Areaair*(Cda+
    $         Yaprev(i)*Cv))
        B8(i)=h*perimeter/(rhoair*velpro*Areaair*(Cda+
    $         Yaprev(i)*Cv))
c        B3(i)=Cwr*Ky(i)/(h*DBLE(f))
c        B4(i)=Cwr/(DBLE(1.-f)*Cs+DBLE(f)*(Cd+Cwr*Wprev(i)))
c        B5(i)=Hv/(Cda*Le(i)*B4(i))
c        B6(i)=Cda*Ky(i)/h
c        B7(i)=0.d0
c        B8(i)=1.d0/(1.d0+Cv*Yaprev(i)/Cda)
      end do
*-----------------------------------------------------------------------
*    This closes the coefficient determination phase of the subroutine.  We
* now move on to the nodal solution of the supply-side phase.
*-----------------------------------------------------------------------
****************************************************************
*-----------------------------------------------------------------------
*  Maintain regenerative-side air boundary conditions
*-----------------------------------------------------------------------

      Yapres(nx)=Yabcregen
      Tapres(nx)=Tabcregen
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for last, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known3(nx)=(1.d0-dtau*B3(nx)/(2.d0*Wmax))*Wprev(nx) + (B3(nx)*
    $            dtau/(2.d0*Yamaxprev(nx)))*Yaprev(nx)

      known4(nx)=Twprev(nx)+dtau*B4(nx)/2.d0*(Yaprev(nx)/Yamaxprev(nx)-
    $            Wprev(nx)/Wmax) + B5(nx)*dtau/2.d0*(Taprev(nx) -
    $            Twprev(nx))
*-----------------------------------------------------------------------
```

```
*    Solve for desiccant conditions during the exhaust rotation for the
* last, channel edge node (an exterior node)
*----------------------------------------------------------------------

      Wpres(nx)=(known3(nx) + dtau*B3(nx)/2.d0 *Yapres(nx)/Yamax(nx))/
     $         (1.d0 + dtau/2.d0*B3(nx)/Wmax)

      Twpres(nx)=(known4(nx) + dtau/2.d0*B4(i)*(Yapres(nx)/Yamax(nx) -
     $          Wpres(nx)/Wmax) + dtau/2.d0*B5(nx)*Tapres(nx))/(1.d0 +
     $          dtau/2.d0*B5(nx))
*----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for all interior
*  nodes for the exhaust rotation of the analysis.  These expressions
*  contain information gathered from previous time steps.
*----------------------------------------------------------------------

      do i=nx-1,2,-1
         known1(i)=(B6(i)*dx/(2.d0*Wmax))*Wpres(i+1) +
     $    (1.d0-B6(i)*dx/ (2.d0*Yamax(i)))*Yapres(i+1)

         known2(i)=B7(i)*dx/2.d0*(Wpres(i+1)/Wmax - Yapres(i+1)/
     $             Yamax(i+1)) + B8(i)*dx/2.d0*(Twpres(i+1) -
     $            Tapres(i+1)) +Tapres(i+1)

 known3(i)=(1.d0-dtau*B3(i)/(2.d0*Wmax))*Wprev(i) + (B3(i)*
     $             dtau/(2.d0*Yamaxprev(i)))*Yaprev(i)

         known4(i)=Twprev(i)+dtau*B4(i)/2.d0*(Yaprev(i)/Yamaxprev(i)-
     $             Wprev(i)/Wmax) + B5(i)*dtau/2.d0*(Taprev(i) -
     $             Twprev(i))
*----------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the exhaust-side
* rotation for all interior nodes
*----------------------------------------------------------------------

         Wpres(i)=(known3(i)+dtau/2.d0*B3(i)/Yamaxprev(i)*(known1(i)/
     $            (1.d0+dx/2.d0*B6(i)/Yamax(i))))/(1.d0 + dtau*B3(i)/
     $            (2.d0*Wmax) - dtau/2.d0*B3(i)/Yamaxprev(i)*dx/2.d0
     $            *B6(i)/Wmax/(1.d0+dx/2.d0*B6(i)/Yamax(i)))

         Yapres(i)=(known1(i) + dx/2.d0*B6(i)/Wmax*Wpres(i))/
     $             (1.d0+dx/2.d0*B6(i)/Yamax(i))

         Twpres(i)=(known4(i) + dtau*B4(i)/2.d0*(Yapres(i)/Yamax(i) -
     $             Wpres(i)/Wmax) + dtau*B5(i)/(2.d0*(1.d0+B8(i)*
     $             dx/2.d0))*(known2(i) + B7(i)*dx/2.d0*(Wpres(i)/
     $             Wmax - Yapres(i)/Yamax(i))))/(1.d0 + dtau*B5(i)/
     $             2.d0 - dtau*B5(i)*(dx*B8(i)/2.d0)/(2.d0*(1.d0+
     $             B8(i)*dx/2.d0)))

         Tapres(i)=(known2(i) + B7(i)*dx/2.d0*(Wpres(i)/Wmax -
     $             Yapres(i)/Yamax(i)) + B8(i)*dx/2.d0*Twpres(i))/
     $             (1.d0+B8(i)*dx/2.d0)
      end do
*----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for first, channel
* edge node.  These expressions contain information gathered from previous
* time steps.
*----------------------------------------------------------------------

      known1(1)=(B6(1)*dx/(4.d0*Wmax))*Wpres(2) +
     $    (1.d0-B6(1)*dx/ (4.d0*Yamax(1)))*Yapres(2)

      known2(1)=B7(1)*dx/4.d0*(Wpres(2)/Wmax - Yapres(2)/
     $             Yamax(2)) + B8(1)*dx/4.d0*(Twpres(2) -
     $            Tapres(2)) +Tapres(2)

      known3(1)=(1.d0-dtau*B3(1)/(2.d0*Wmax))*Wprev(1) + (B3(1)*
```

```
     $            dtau/(2.d0*Yamaxprev(1)))*Yaprev(1)

      known4(1)=Twprev(1)+dtau*B4(1)/2.d0*(Yaprev(1)/Yamaxprev(1)-
     $            Wprev(1)/Wmax) + B5(1)*dtau/2.d0*(Taprev(1) -
     $            Twprev(1))
*_______________________________________________________________________
*    Solve for desiccant and air conditions on the exhaust-side rotation for
* the first, channel edge node (an exterior node)
*_______________________________________________________________________

      Wpres(1)=(known3(1)+dtau/2.d0*B3(1)/Yamaxprev(1)*(known1(1)/
     $           (1.d0+dx/2.d0*B6(1)/Yamax(1))))/(1.d0 + dtau*B3(1)/
     $           (2.d0*Wmax) - dtau/2.d0*B3(1)/Yamaxprev(1)*dx/4.d0
     $           *B6(1)/Wmax/(1.d0+dx/4.d0*B6(1)/Yamax(1)))

      Yapres(1)=(known1(1) + dx/4.d0*B6(1)/Wmax*Wpres(1))/
     $            (1.d0+dx/4.d0*B6(1)/Yamax(1))

      Twpres(1)=(known4(1) + dtau*B4(1)/2.d0*(Yapres(1)/Yamax(1) -
     $            Wpres(1)/Wmax) + dtau*B5(1)/(2.d0*(1.d0+B8(1)*
     $            dx/2.d0))*(known2(1) + B7(1)*dx/2.d0*(Wpres(1)/
     $            Wmax - Yapres(1)/Yamax(1))))/(1.d0 + dtau*B5(1)/
     $            2.d0 - dtau*B5(1)*(dx*B8(1)/2.d0)/(2.d0*(1.d0+
     $            B8(1)*dx/2.d0)))


      Tapres(1)=(known2(1) + B7(1)*dx/4.d0*(Wpres(1)/Wmax -
     $            Yapres(1)/Yamax(1)) + B8(1)*dx/4.d0*Twpres(1))/
     $            (1.d0+B8(1)*dx/4.d0)
*_______________________________________________________________________
*   Return to the MAIN DRIVER
*_______________________________________________________________________

      return
      end

**********************************************************************
      SUBROUTINE XFER
**********************************************************************
*_______________________________________________________________________
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to XFER and various other subroutines easier.
*_______________________________________________________________________

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/airprops/ TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
      common/solution/ Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot
*_______________________________________________________________________
*   Declare all variables that are used in the subroutine XFER.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*_______________________________________________________________________

      integer i,j,k,l,nx,cyclenum,NMAX1,NMAX2
      double precision TempKo,TempK,Yamax,Yamaxprev,u,PwsT,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring
      double precision Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot
*_______________________________________________________________________
*     Initialize parameters used in the dimensioning of all arrays used in
* the XFER and transferred between subroutines.
```

```
*-----------------------------------------------------------------------

      parameter (NMAX1=100)
      parameter (NMAX2=2500)

      dimension TempK(NMAX1),Yamax(NMAX1),Yamaxprev(NMAX1),u(NMAX1),
     $  PwsT(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Yatotpro(NMAX2),
     $   Tatotpro(NMAX2),Twtotreg(NMAX2),Wtotreg(NMAX2),
     $   Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    After subroutines are completed, air conditions labeled 'present'
* are transferred to 'previous' for nodal solution purposes.
*-----------------------------------------------------------------------

         do i=1,nx
            Twprev(i)=Twpres(i)
            Wprev(i)=Wpres(i)
            Yaprev(i)=Yapres(i)
            Taprev(i)=Tapres(i)
            Yamaxprev(i)=Yamax(i)
         end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------
      return
      end

**********************************************************************
      SUBROUTINE EFFICIENCY
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to EFFICIENCY and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/time/ tautot,dtau
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/heat/ hgpres,hgprev,hgpro,hgregen,hgtot
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,
     $  Yaprev,Taprev,Twring,Wring,Yaring,Taring,Yamaxring
      common/solution/ Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine EFFICIENCY.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1,NMAX2
      real tautot
      double precision dtau
      double precision eff1,eff2,eff3
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision hgpres,hgprev,hgpro,hgregen,hgtot,haprotot,
     $  haproregen
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Yapres,Tapres,Twprev,Wprev,Yaprev,
     $  Taprev,Twring,Wring,Yaring,Taring,Yamaxring
      double precision Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot
```

```
      double precision Ppro,Ptot,Pregen,Xpro,Xtot,Xregen,
     $  Ypro,Ytot,Yregen
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the EFFICIENCY and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)
      parameter (NMAX2=2500)

      dimension Twpres(NMAX1),Wpres(NMAX1),Yapres(NMAX1),Tapres(NMAX1),
     $  Twprev(NMAX1),Wprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),
     $   Twring(NMAX1),Wring(NMAX1),Yaring(NMAX1),
     $   Taring(NMAX1),Yamaxring(NMAX1)
      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Yatotpro(NMAX2),
     $   Tatotpro(NMAX2),Twtotreg(NMAX2),Wtotreg(NMAX2),
     $   Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    Determine coefficients used in solving for enthalpy
*-----------------------------------------------------------------------

 Xpro= 374.12d0-Tabcpro
 Xtot= 374.12d0-Ta
 Xregen= 374.12d0-Tabcregen
 Ypro= Xpro*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $        Xpro**2)*Xpro)/((Tabcpro+273.15d0)*(1.d0+
     $        2.1878462d-3*Xpro))
 Ytot= Xtot*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $   Xtot**2)*Xtot)/((Ta+273.15d0)*(1.d0+2.1878462d-3*
     $   Xtot))
 Yregen= Xregen*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $  Xregen**2)*Xregen)/((Tabcregen+273.15d0)*(1.d0+
     $  2.1878462d-3*Xregen))

 Ppro = 14.696d0*218.167d0/(10.0d0**Ypro)
 Ptot = 14.696d0*218.167d0/(10.0d0**Ytot)
 Pregen =  14.696d0*218.167d0/(10.0d0**Yregen)
 Xpro = DLOG10(Ppro)
 Xtot = DLOG10(Ptot)
 Xregen = DLOG10(Pregen)
*-----------------------------------------------------------------------
*   Calculate enthaply associated with the air streams
*-----------------------------------------------------------------------

 hgpro = (1105.9387d0+( 32.756807d0+(4.6198474d0+(0.20672996d0+
     $  (-0.5411693d0+(0.49241362d0-0.17884885d0*Xpro)*Xpro)
     $  *Xpro)*Xpro)*Xpro)*Xpro)/.4299d0
 hgtot =(1105.9387d0+( 32.756807d0+(4.6198474d0+(0.20672996d0+
     $  (-0.5411693d0+(0.49241362d0-0.17884885d0*Xtot)*Xtot)*
     $  Xtot)*Xtot)*Xtot)*Xtot)/.4299d0
 hgregen =(1105.9387d0+( 32.756807d0+(4.6198474d0+(0.20672996d0
     $  +(-0.5411693d0+(0.49241362d0-0.17884885d0*Xregen)
     $  *Xregen)*Xregen)*Xregen)*Xregen)*Xregen)/.4299d0

610   haprotot=Cda*(Tabcpro-Ta) + (Yabcpro*hgpro-
     $ Ya*hgtot)
      haproregen=Cda*(Tabcpro-Tabcregen) + (Yabcpro*hgpro-Yabcregen*
     $  hgregen)
*-----------------------------------------------------------------------
*    Calculate sensible efficiency
*-----------------------------------------------------------------------

      eff1= (Tabcpro-Ta)/(Tabcpro-Tabcregen)
*-----------------------------------------------------------------------
*    Calculate latent efficiency
*-----------------------------------------------------------------------

      eff2=(Yabcpro*hgpro-Ya*hgtot)/
```

```
     $  (Yabcpro*hgpro-Yabcregen*hgregen)
*-----------------------------------------------------------------------
*    Calculate total efficiency
*-----------------------------------------------------------------------

      eff3= haprotot/haproregen
*-----------------------------------------------------------------------
*    Open the output data file and record efficiency values
*-----------------------------------------------------------------------

      open(44,file='wheel_film.dat')
      write(44,*)
      write(44,*) 'After',cyclenum,'cycles,'
      write(44,*)
      write(44,*) 'Efficiencies are:'
      write(44,*)
      write(44,*) 'Sensible =',eff1
      write(44,*) 'Latent =',eff2
      write(44,*) 'Total =',eff3
      write(44,*)

*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE OUTPUT
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to OUTPUT and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Hv,h,velpro,velreg,Wmax,Patm
      common/solution/ Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine OUTPUT.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX2
      real g,f,perw,perd,n,tautot
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision Hv,h,velpro,velreg,Wmax,Patm
      double precision Twtotpro,Wtotpro,Yatotpro,Tatotpro,Twtotreg,
     $  Wtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,Twtot,Wtot

*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the OUTPUT and transferred between subroutines.
*-----------------------------------------------------------------------
```

```
      parameter (NMAX2=2500)

      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Yatotpro(NMAX2),
     $   Tatotpro(NMAX2),Twtotreg(NMAX2),Wtotreg(NMAX2),
     $   Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    Record elapsed time and pressure drop as well as final nodal spacing
* in the output file.
*-----------------------------------------------------------------------

      write(44,*) 'Rotation time = ',DBLE(tautot)*2.d0,'seconds'
      write(44,*)
      write(44,*) 'Pressure Drop = ',pressuredrop,'m water'
      write(44,*)
      write(44,*) 'dx = ',dx,'m'
      write(44,*)
*-----------------------------------------------------------------------
*    Record exit supply and exhaust air and desiccant data at every 100
* time steps, for a more detailed look at the rotary wheel exchanger behavior.
*-----------------------------------------------------------------------

      do j=1,cyclenum
 write(44,*) 'CYCLE #:',j
 write(44,*) 'SUPPLY'
 write(44,*) 'Tair (Celcius)= ',REAL(Tatotpro(j))
 write(44,*) 'Yair (kg water/kg air)= ',REAL(Yatotpro(j))
 write(44,*) 'Tbed (Celcius)= ',REAL(Twtotpro(j))
 write(44,*) 'Ybed (kg water/kg desiccant)= ',REAL(Wtotpro(j))
 write(44,*) 'EXHAUST'
 write(44,*) 'Tair (Celcius)= ',REAL(Tatotreg(j))
 write(44,*) 'Yair (kg water/kg air)= ',REAL(Yatotreg(j))
 write(44,*) 'Tbed (Celcius)= ',REAL(Twtotreg(j))
 write(44,*) 'Ybed (kg water/kg desiccant)= ',REAL(Wtotreg(j))
 write(44,*)
       end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end
```

Appendix D

Polyvinyl Alcohol Foam Coated Rotary Wheel Total Enthalpy Exchanger Performance Estimation Code

```
      PROGRAM ADSORB
**********************************************************************
*      Performance Estimates of A
*  Rotary Wheel Total Enthalpy Exchanger
*    Coated with a Porous (Adsorbing) Polymer Desiccant
*                              by
*                        Christie Staton
*  1997
*
**********************************************************************
*
*
*    This program estimates the sensible, latent, and total performance
* efficiencies of a Rotary Wheel Total Enthalpy Exchanger coated with a
* porous (absorbing) polymer desiccant. Exit conditions for both the supply
* (process) and exhaust (regenerative) air streams are calculated along
* with the desiccant bed conditions of the front and back faces of the
* wheel.  A solution is obtained via an integration over half of the
* wheel face.
*
*    A choice of three desiccant channel geometries, tridiagonal,
* hexagonal, or sinusoidal can be implemented during the program run.  In
* addition to the channel geometry choices and rotational time inputted by
* the program user, the program requires six input data files.  These files
* are identified and described in detail in the subroutine PROPERTY.
*
*------------------------------------------------------------------------
*    We begin with the MAIN DRIVER of the program.  This is the portion of
* the code form which all subroutines are called.  The MAIN DRIVER organizes
* the operation of the subroutines in such a manner as to get the
* performance estimation accomplished in the most efficient way.
*
```

```
*   Subroutines called within the main driver of the program:
*
*   PROPERTY: reads input data files and establishes wheel and desiccant
*             material information
*   BNDYPRO: establishes boundary conditions for process (supply) air
*            stream
*   COEFPRO: establishes coefficients used in PROCESS which are dependent
*            on temperature and/or moisture content
*   PROCESS: contains explicit finite-differencing equations used to solve
*            nodal conditions in both the air stream and the desiccant bed
*            along the process (supply) route
*   APPROXIMATE: called during the summation process during the finite-
*                differencing scheme and is used to establish nodal values
*                for the previous process when the next process (half
*                cycle) is engaged
*   XFER: called after each time step during PROCESS or REGENERATION
*             to establish nodal values for the previous time step
*   BNDYREGEN: establishes boundary conditions for regenerative (exhaust)
*              air stream
*   COEFREGEN: establishes coefficients used in REGENERATION which are
*              dependent on temperature and/or moisture content
*   REGENERATION: contains explicit finite-differencing equations used to
*                 solve nodal conditions in both the air stream and the
*                 desiccant bed along the regeneration (exhaust) route
*   EFFICIENCY: using integrated air stream and desiccant bed conditions,
*               sensible, latent, and total efficiencies are determined
*   OUTPUT: transfer efficiency information along with cycle-by-cycle
*           process and regenerative exit conditions to an output file
*----------------------------------------------------------------------

      implicit none
*-----------------------------------------------------------------------
*   Declare all variables that are used in the MAIN DRIVER and elsewhere.
* Variables declare here are used throughout the program; they are used in
* several different subroutines.
*-----------------------------------------------------------------------

      integer z,i,j,k,l,nx,cyclenum,NMAX1,NMAX2
      real g,f,perw,perd,n,tautot,timetot
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Q,Hv,h,velpro,
     $  velreg,Wmax,Patm
      double precision hgpres,hgprev,hgpro,hgregen,hgtot
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
      double precision Twpros,Wpros,Ywpros,Yapros,Tapros,
     $   Twregs,Wregs,Ywregs,Yaregs,Taregs,
     $   Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,Tatot,Ta,Ya,
     $   Yatot,Twtot,Wtot,Ywtot,Yatotreg,Tatotreg,Twtotreg,Wtotreg,
     $   Ywtotreg
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the MAIN DRIVER and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)
      parameter (NMAX2=2500)
```

```
      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $   known4(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Ywtotpro(NMAX2),
     $   Yatotpro(NMAX2),Tatotpro(NMAX2),TWtotreg(NMAX2),
     $   Wtotreg(NMAX2),Ywtotreg(NMAX2),Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to various subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/heat/ hgpres,hgprev,hgpro,hgregen,hgtot
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,
     $   Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
      common/solution/ Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,
     $   Yatot,Twtot,Wtot,Ywtot
*----------------------------------------------------------------------
*    We begin with the body of the MAIN DRIVER of the program, the portion
*  of the program from which all subroutines are called.
*    First, the subroutine PROPERTY is called. This subroutine reads
* user input data dealing with plate geometry and air stream conditions.
* The input data files containing all density, specific heat, boundary
* conditions of the supply and exhaust air, initial conditions of the
* desiccant bed, plate geometry, and air property information are read in
* the section of the program.
*-----------------------------------------------------------------------

      call PROPERTY
*-----------------------------------------------------------------------
*    At this point the variable that determines how long one wheel rotation
* will be is determined.  Variable tautot defines the total elapsed time
* (in seconds) for one full rotation modeled by the program.  This is
* determined by converting rotational speed (in RPM),timetot, to
* rotational time.  Next, the amount of time each side of the wheel spends
* in each air stream (assuming the wheel is divided in half) is calculated.
*-----------------------------------------------------------------------

      timetot=40.0
      tautot=60.0/timetot
      tautot=tautot/2.0
*-----------------------------------------------------------------------
*    Using the time step size defined in PROPERTY, determine how many time
```

```
* steps will occur in half a wheel rotation.  This number is defined by l.
*-----------------------------------------------------------------------

      l=0
      do g=dtau,tautot,dtau
         l=l+1
      end do
*-----------------------------------------------------------------------
*    After determining the total time to be modeled as well as the time step
* to be used in the analysis, a time loop is introduced.  This loop is based
* upon the number of cycles to be analyzed.  This term is indicated in one
* of the input files read in subroutine PROPERTY.  Keep in mind that
* this model is analyzed by stepping along in distance within a specified
* space in time.
*-----------------------------------------------------------------------

      do j=1,cyclenum
*-----------------------------------------------------------------------
*    Next, variables representing total air and desiccant material exiting
* conditions for both the supply and exhaust side are initialized.  At every
* time step, these conditions are updated, and therefore must be reset as each
* time step is updated.
*-----------------------------------------------------------------------

 Tatot=0.d0
 Yatot=0.d0
        Ta=0.d0
        Ya=0.d0
 Tatotpro(j)=0.d0
 Yatotpro(j)=0.d0
 Twtotpro(j)=0.d0
 Wtotpro(j)=0.d0
        Ywtotpro(j)=0.d0
        Tatotreg(j)=0.d0
 Yatotreg(j)=0.d0
 Twtotreg(j)=0.d0
 Wtotreg(j)=0.d0
        Ywtotreg(j)=0.d0

        Tapros=0.d0
        Yapros=0.d0
        Twpros=0.d0
        Wpros=0.d0
        Ywpros=0.d0
        Taregs=0.d0
        Yaregs=0.d0
        Twregs=0.d0
        Wregs=0.d0
        Ywregs=0.d0

        z=0
        call BNDYPRO
        call COEFPRO

        do g=dtau,tautot,dtau
           z=z+1
           if (z .eq. INT(tautot/(2.*REAL(dtau))) ) then
              call APPROXIMATE
           end if
           call PROCESS
           call XFER
           Tapros=Tapros + Tapres(nx)
           Yapros=Yapros + Yapres(nx)
           Twpros=Twpros + Twpres(nx)
           Wpros=Wpros + Wpres(nx)
           Ywpros=Ywpros + Ywpres(nx)
        enddo
```

```
         Tatotpro(j)=Tapros/l
 Yatotpro(j)=Yapros/l
 Twtotpro(j)=Twpros/l
 Wtotpro(j)=Wpros/l
         Ywtotpro(j)=Ywpros/l
 Tatot=Tatotpro(j)
 Yatot=Yatotpro(j)
         Ta=Tatot
         Ya=Yatot
         Twtot=Twtotpro(j)
 Wtot=Wtotpro(j)
         Ywtot=Ywtotpro(j)

        z=0
        call BNDYREGEN
        call COEFREGEN

        do g=dtau,tautot,dtau
           z=z+1
           if (z .eq. INT(tautot/(2.*REAL(dtau))) ) then
             call APPROXIMATE
           end if
           call REGENERATION
           call XFER
            Taregs=Taregs + Tapres(1)
            Yaregs=Yaregs + Yapres(1)
            Twregs=Twregs + Twpres(1)
            Wregs=Wregs + Wpres(1)
            Ywregs=Ywregs + Ywpres(1)
        enddo

        Tatotreg(j)=Taregs/l
Yatotreg(j)=Yaregs/l
Twtotreg(j)=Twregs/l
Wtotreg(j)=Wregs/l
        Ywtotreg(j)=Ywregs/l

     enddo

     call EFFICIENCY

     call OUTPUT

     END



************************************************************************
      SUBROUTINE PROPERTY
************************************************************************
*
*    The primary function of this subroutine is read in data dealing with
* channel geometry, efficiency evaluation, density, specific heat, boundary
* conditions of the supply and exhaust air, initial conditions of the
* desiccant bed, wheel geometry, and air property, and wheel performance
* information.  All of this information is read via a series of data input
* files.  Once the proper information is  obtained,  geometric information
* leading to the solution of performance efficiencies is calculated.
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to PROPERTY and various other subroutines easier.
*------------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
```

```
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,
     $   Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine PROPERTY.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      real g,f,perw,perd,n,tautot
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision pi,gravity,Dh,Red,kviscair,friction,tothead,
     $   spgravity
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Q,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the PROPERTY and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    Read in the input data files
*  performance.dat:
*   cyclenum=   # of cycles to be modeled
*  spheat.dat:
*   Cwr=  reference specific heat (sp. heat of water)
*   Cda=  specific heat of dry air
*   Cv=   specific heat of water vapor
*   Cd=  specific heat of desiccant
*   Cs=  specific heat of support substrate
*   Cw=  specific heat of wheel
*   spgravity=  specific gracity of water
*  density.dat:
*   rhod=  density of desiccant
*   rhos=  density of support substrate
*   rhoevery=  density of wheel
*   rhoair=  density of air
*   kviscair=  kinematic viscosity of air
```

```
*  design.dat:
*   Do=  outer diameter
*   Di=  inner diameter
*   wheeldepth=  depth of dehumidifier
*   thickd=  thickness of desiccant
*   thicks=  thickness of support substrate
*  nondim.dat:
*   Q=  heat of adsorption
*   Wmax=  maximum moisture capacity of desiccant
*   Patm=  atmospheric pressure
*   h=  heat transfer coefficient
*   velpro=  velocity of supply air
*   velreg=  velocity of exhaust air
*  boundary.dat:
*   Twic=  initial temperature condition of the dehumidifier
*   Wic=  initial moisture content condition of the dehumidifier
*   Tabcpro=  air temperature boundary condition for supply side
*   Yabcpro=  air moisture content boundary condition for supply side
*   Tabcregen=  air temperature boundary condition for exhaust side
*   Yabcregen=  air moisture content boundary condition for exhaust side
*-----------------------------------------------------------------------

      pi=3.141592654d0
      gravity=9.81d0

      open(1,file='performance.dat')
      read(1,*) cyclenum
      open(2,file='spheat.dat')
      read(2,*) Cwr,Cda,Cv,Cd,Cs,Cw,spgravity
      open(3,file='density.dat')
      read(3,*) rhod,rhos,rhoevery,rhoair,kviscair
      open(4,file='design.dat')
      read(4,*) Do,Di,wheeldepth,thickd,thicks
      open(7,file='nondim.dat')
      read(7,*) Q,Wmax,Patm,h,velpro,velreg
      open(8,file='boundary.dat')
      read(8,*) Twic,Wic,Tabcpro,Yabcpro,Tabcregen,Yabcregen
*-----------------------------------------------------------------------

100   print *, 'Which channel geometry would you like to use?'
      print *
      print *, 'Enter the number that corresponds to the'
      print *, 'correct cross-sectional shape:'
      print *
      print *, '1 = equilateral triangular'
      print *, '2 = sinusoidal'
      print *, '3 = regular hexagonal'
      read *, response
      print *
*-----------------------------------------------------------------------
*    Check to make sure the inputs are valid:
*-----------------------------------------------------------------------

      if (response.ne.1 .and. response.ne.2 .and. response.ne.3) then
      print *, 'Geometric input parameter is invalid, please try again.'
      print *, 'The input should be in the form of a 1, 2, or 3.'
      go to 100
      end if
*-----------------------------------------------------------------------
*    If the triangular channel geometry is chosen, assign proper values to
* the various area inputs.  Base, Areatot, Areaair, and Areas define
* channel width, total channel area, air area in the channel, and
* structural area in the channel, respectively.
*-----------------------------------------------------------------------

      if (response.eq.1) then
 print *
 print *, 'Please enter the channel height (meters):'
```

```
        read *, height
 base=2.d0*height/DTAN(pi/3.d0)
        perimeter=3.d0*base
 Areatot=base*height/2.d0
 Areaair=(height-(2.d0*thickd+thicks))*(height-(2.d0*thickd+
     $  thicks))/DTAN(pi/3.d0)
 Areas=(height-thicks)*(height-thicks)/DTAN(pi/3.d0)
        Dh=4.d0*Areaair/(6.d0*(height-(2.d0*thickd+thicks))/
     $      DTAN(pi/3.d0))
        Red=(velpro*Dh)/kviscair
        friction=64.d0/Red
        tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
     $        wheeldepth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
     $        velpro**2/(2.d0*gravity))
        pressuredrop=rhoair*gravity*tothead
        pressuredrop=pressuredrop/spgravity
 go to 120
     end if
*-----------------------------------------------------------------------
*   If the sinusoidal channel geometry is chosen, assign proper values to
* the various area inputs.  Base, Areatot, Areaair, and Areas define
* channel width, total channel area, air area in the channel, and
* structural area in the channel, respectively.
*-----------------------------------------------------------------------

     if (response.eq.2) then
 print *
 print *, 'Please enter the channel wave amplitude (meters):'
 read *, height
        height=DBLE(height)
 print *
 print *, 'Please enter the channel wave length (meters):'
 read *, base
 base=2.d0*height/DTAN(pi/3.d0)
        perimeter=3.d0*base
 Areatot=base*height/2.d0
 Areaair=(height-(2.d0*thickd+thicks))*(height-(2.d0*thickd+
     $  thicks))/DTAN(pi/3.d0)
 Areas=(height-thicks)*(height-thicks)/DTAN(pi/3.d0)
        Dh=4.d0*Areaair/(6.d0*(height-(2.d0*thickd+thicks))/
     $      DTAN(pi/3.d0))
        Red=(velpro*Dh)/kviscair
        friction=64.d0/Red
        tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
     $        wheeldepth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
     $        velpro**2/(2.d0*gravity))
        pressuredrop=rhoair*gravity*tothead
        pressuredrop=pressuredrop/spgravity
 go to 120
     end if
*-----------------------------------------------------------------------
*   If the hexagonal channel geometry is chosen, assign proper values to
* the various area inputs.  Base, Areatot, Areaair, and Areas define
* channel width, total channel area, air area in the channel, and
* structural area in the channel, respectively.
*-----------------------------------------------------------------------

     if (response.eq.3) then
 print *
 print *, 'Please enter the distance between parallel sides
     $    (meters):'
 read *, height
 base=3.d0*height/DTAN(pi/3.d0)
 Areatot=(1.5d0*height**2)/DTAN(pi/3.d0)
 Areaair=(1.5d0*(height-(2.d0*thickd+thicks))**2)/
     $   DTAN(pi/3.d0)
 Areas= (1.5d0*(height-thicks)**2)/DTAN(pi/3.d0)
        Dh=4.d0*Areaair/(6.d0*base)
```

```
        Red=(velpro*Dh)/kviscair
        friction=64.d0/Red
        tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
    $        wheeldepth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
    $        velpro**2/(2.d0*gravity))
        pressuredrop=rhoair*gravity*tothead
        pressuredrop=pressuredrop/spgravity
 go to 120
      end if
*-------------------------------------------------------------------------
*    Determine actual length of elements; nx = number of nodes along
* channel and dx = distance between nodes
*-------------------------------------------------------------------------

120   nx=21
      dx=wheeldepth/dble(nx-1)
*-------------------------------------------------------------------------
*    Calculate areas, volumes, weights, and mass fractions needed for
* numerical solution of performance estimates,
*-------------------------------------------------------------------------
*    Area of the desiccant(per channel):
*    Mass of desiccant within each channel:
*-------------------------------------------------------------------------

      Aread=Areas - Areaair
      Massdc=Aread*wheeldepth*rhod
*-------------------------------------------------------------------------
*    Percentage of dehumidifier in wheel, and desiccant in dehumidifier
*-------------------------------------------------------------------------

      perw=REAL((Areatot-Areaair)/Areatot)
      perd=REAL(Aread/(Areatot-Areaair))
*-------------------------------------------------------------------------
*    Tot volume of wheel, number of channels in wheel, mass of dehumidifier
*-------------------------------------------------------------------------

      Voltot=3.14d0*(Do**2-Di**2)*wheeldepth/4.d0
      Volw=Voltot*DBLE(perw)
      n=int((Voltot-Volw)/(Areaair*wheeldepth))
      Massw=Volw*(rhod*DBLE(perd) + rhos*(1.d0-DBLE(perd)))
      f=REAL(Voltot*DBLE(perw)*DBLE(perd)*rhod/Massw)
*-------------------------------------------------------------------------
*    Determine length of time steps to be used in the solution
*-------------------------------------------------------------------------

      dtau=0.01d0
*-------------------------------------------------------------------------
*    Assign initial values to each node along the channel depth
*-------------------------------------------------------------------------

      do i=1,nx
         Taring(i)=Tabcpro
         Yaring(i)=Yabcpro
         Twring(i)=Twic
 Wring(i)=Wic

         TempKw(i)=2.7315d+2 + Twic
         uw(i)=(6.4727d+2 - TempKw(i))  *(3.2437814d0 +
    $         (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKw(i))*
    $         (6.4727d+2 - TempKw(i)))* (6.4727d+2 - TempKw(i)))/
    $         (TempKw(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKw(i))))
         PwsTw(i)=2.18167d+2/(10.d0 **uw(i))
         Ywmaxring(i)=6.22d-1*PwsTw(i)/Patm

         Ywring(i)=Wring(i)*Ywmaxring(i)/Wmax
      end do
         Taring(1)=Tabcpro
*-------------------------------------------------------------------------
```

```
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

************************************************************************
      SUBROUTINE APPROXIMATE
************************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to APPROXIMATE various subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine APPROXIMATE.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the APPROXIMATE and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    Assign intermediate, quarter-cycle values to be used as initial
* conditions in the following subroutine (either PROCESS or REGENERATION)
*-----------------------------------------------------------------------

      do i=1,nx
         Twring(i)=Twpres(i)
         Wring(i)=Wpres(i)
         Ywring(i)=Ywpres(i)
         Yaring(i)=Yapres(i)
         Taring(i)=Tapres(i)
         Yamaxring(i)=Yamax(i)
         Ywmaxring(i)=Ywmax(i)
      end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------
```

```
      return
      end

**********************************************************************
      SUBROUTINE BNDYPRO
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to BNDYPRO various subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine BNDYPRO.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the BNDYPRO and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    Assign initial conditions before the beginning of PROCESS
*-----------------------------------------------------------------------

      do i=1,nx
         Tapres(i)=Taring(i)
         Yapres(i)=Yaring(i)
         Twpres(i)=Twring(i)
         Wpres(i)=Wring(i)
         Ywpres(i)=Ywring(i)
         Yamax(i)=Yamaxring(i)
      end do
*-----------------------------------------------------------------------
*    Maintain boundary conditions
*-----------------------------------------------------------------------

      Yapres(1)=Yabcpro
      Tapres(1)=Tabcpro
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------
```

```
      return
      end

**********************************************************************
      SUBROUTINE COEFPRO
**********************************************************************
*------------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to COEFPRO and various other subroutines easier.
*------------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*------------------------------------------------------------------------
*   Declare all variables that are used in the subroutine COEFPRO.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*------------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision rhod,rhos,rhoair,rhoevery,Q,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev, Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*------------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the COEFPRO and transferred between subroutines.
*------------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*------------------------------------------------------------------------
*   Assign air conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*------------------------------------------------------------------------
*   Step along the x direction, along the plate depth
*------------------------------------------------------------------------

      do i=1,nx
         Yaprev(i)=Yapres(i)
        Taprev(i)=Tapres(i)
         Twprev(i)=Twpres(i)
         Wprev(i)=Wpres(i)
         Ywprev(i)=Ywpres(i)
         TempKa(i)=2.7315d+2 + Taprev(i)
         TempKw(i)=2.7315d+2 + Twprev(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKa(i))))
```

```
        uw(i)=(6.4727d+2 - TempKw(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKw(i))*
     $        (6.4727d+2 - TempKw(i)))* (6.4727d+2 - TempKw(i)))/
     $        (TempKw(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKw(i))))
        PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
        PwsTw(i)=2.18167d+2/(10.d0 **uw(i))
 Yamaxprev(i)=6.22d-1*PwsTa(i)/Patm
        Ywmaxprev(i)=6.22d-1*PwsTw(i)/Patm
      end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE PROCESS
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to PROCESS and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine PROCESS.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      real g,f,perw,perd,n,tautot
      double precision Ky
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Q,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
```

```
* the PROCESS and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension Ky(NMAX1)
      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $   known4(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    Stepping along in x, the plate depth, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every iteration/time step.
*-----------------------------------------------------------------------
      do i=1,nx

         if (Wprev(i) .gt. Wmax) then
             Wprev(i)=Wmax
         end if

         TempKa(i)=2.7315d+2 + Taprev(i)
         TempKa(1)=2.7315d+2 + Tabcpro
         TempKw(i)=2.7315d+2 + Twprev(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $        TempKa(i))))
         uw(i)=(6.4727d+2 - TempKw(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKw(i))*
     $        (6.4727d+2 - TempKw(i)))* (6.4727d+2 - TempKw(i)))/
     $        (TempKw(i) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $        TempKw(i))))
         PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
         PwsTw(i)=2.18167d+2/(10.d0 **uw(i))
 Yamax(i)=6.22d-1*PwsTa(i)/Patm
         Ywmax(i)=6.22d-1*PwsTw(i)/Patm
*-----------------------------------------------------------------------
*    Calculate the mass transfer coefficient, a moisture content dependent
* material property.
*-----------------------------------------------------------------------

          Ky(i)=10.d0*(3.d-5*(Wprev(i)*Massdc)**2 - 3.d-4*
     $          (Wprev(i)*Massdc) + 6.d-4)
c         Ky(i)=10.d0*(5.8d-4 - 1.d-5*(1.d+3*Wprev(i)*Massdc))
*-----------------------------------------------------------------------
*    Calculate the heat of vaporization, a temperature dependent
* material property.
*-----------------------------------------------------------------------

         Hv=2502.d0-Cv/1.d+3*Taprev(i)
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------

 B3(i)=Ky(i)/(thickd*rhod*DBLE(f))
         B4(i)=Hv*Ky(i)/(thickd*rhoevery*
```

```
     $         (DBLE(1.-f)*Cs+DBLE(f)*(Cd+Wprev(i)*(Hv-Q))))
         B5(i)=h/(thickd*rhoevery*(DBLE(1.-f)*
     $         Cs+DBLE(f)*(Cd+Wprev(i)*(Hv-Q))))
         B6(i)=perimeter*Ky(i)/(rhoair*velpro*Areaair)
         B7(i)=perimeter*Ky(i)*Hv/(rhoair*velpro*Areaair*(Cda+
     $         Yaprev(i)*Cv))
         B8(i)=h*perimeter/(rhoair*velpro*Areaair*(Cda+
     $         Yaprev(i)*Cv))
       end do
*------------------------------------------------------------------------
*    This closes the coefficient determination phase of the subroutine.  We
* now move on to the nodal solution of the supply-side phase.
*------------------------------------------------------------------------
*************************************************************************
*------------------------------------------------------------------------
*  Maintain supply-side air boundary conditions
*------------------------------------------------------------------------

      Yapres(1)=Yabcpro
      Tapres(1)=Tabcpro
*------------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for first, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*------------------------------------------------------------------------

      known3(1)=Wprev(1) + B3(1)*dtau*(Yaprev(1)-Ywprev(1))

      known4(1)=Twprev(1) + dtau*B4(1)*(Yaprev(1)-Ywprev(1)) +
     $            B5(1)*dtau*(Taprev(1) - Twprev(1))
*------------------------------------------------------------------------
*    Solve for desiccant conditions on the supply-side surface for the
* first, plate edge node (an exterior node)
*------------------------------------------------------------------------

      Wpres(1)=(known3(1) + B3(1)*dtau*Yapres(1))/(1.d0 + B3(1)*
     $         dtau*Ywmax(1)/Wmax)

      Ywpres(1)=Wpres(1)*Ywmax(1)/Wmax

      Twpres(1)=(known4(1) + B4(1)*dtau*(Yapres(1)-Ywpres(1)) +
     $            B5(1)*dtau*Tapres(1))/(1.d0 + B5(1)*dtau)
*------------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for all interior
*  nodes.  These expressions contain information gathered from previous
* time steps.
*------------------------------------------------------------------------

      do i=2,nx-1

         known1(i)=(B6(i)*dx/2.d0)*Ywpres(i-1) +
     $    (1.d0-B6(i)*dx/ 2.d0)*Yapres(i-1)

         known2(i)=B7(i)*dx/2.d0*(Ywpres(i-1) - Yapres(i-1)) +
     $            B8(i)*dx/2.d0*(Twpres(i-1) - Tapres(i-1)) +
     $            Tapres(i-1)

 known3(i)=Wprev(i) + B3(i)*dtau*(Yaprev(i)-Ywprev(i))

         known4(i)=Twprev(i) + dtau*B4(i)*(Yaprev(i)-Ywprev(i)) +
     $            B5(i)*dtau*(Taprev(i) - Twprev(i))
*------------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the supply-side
* surface for all interior nodes
*------------------------------------------------------------------------

        Wpres(i)=(known3(i) + B3(i)*dtau*known1(i)/(1.d0+B6(i)*dx/
     $           2.d0))/(1.d0 - (B3(i)*B6(i)*Ywmax(i)*dtau*dx/
```

```
     $           (2.d0*Wmax))/(1.d0 + B6(i)*dx/2.d0) + B3(i)*dtau*
     $           Ywmax(i)/Wmax)

         Yapres(i)=(known1(i) + B6(i)*dx/2.d0 * (Ywmax(i)/Wmax *
     $             Wpres(i)))/(1.d0 + B6(i)*dx/2.d0)

         Ywpres(i)=(Ywmax(i)/Wmax *Wpres(i))

         Twpres(i)=(known4(i) + B4(i)*dtau*(Yapres(i)-Ywpres(i)) +
     $             B5(i)*dtau/(1.d0 + B8(i)*dx/2.d0) * (known2(i) +
     $             B7(i)*dx/2.d0 * (Ywpres(i)-Yapres(i))))/(1.d0 +
     $             B5(i)*dtau - B5(i)*B8(i)*dtau*dx/(2.d0*(1.d0 +
     $             B8(i)*dx/2.d0)))

         Tapres(i)=(known2(i) + B7(i)*dx/2.d0*(Ywpres(i) - Yapres(i))
     $             + B8(i)*dx/2.d0*Twpres(i))/(1.d0+B8(i)*dx/2.d0)
      end do
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for last, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known1(nx)=(B6(nx)*dx/4.d0)*Ywpres(nx-1) +
     $    (1.d0-B6(nx)*dx/4.d0)*Yapres(nx-1)

      known2(nx)=B7(nx)*dx/4.d0*(Ywpres(nx-1) - Yapres(nx-1)) +
     $             B8(nx)*dx/4.d0*(Twpres(nx-1) - Tapres(nx-1)) +
     $             Tapres(nx-1)

      known3(nx)=Wprev(nx) + B3(nx)*dtau*(Yaprev(nx)-Ywprev(nx))

      known4(nx)=Twprev(nx) + dtau*B4(nx)*(Yaprev(nx)-Ywprev(nx)) +
     $             B5(nx)*dtau*(Taprev(nx) - Twprev(nx))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream  conditions on the supply-side
* surface for the last, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(nx)=(known3(nx) + B3(nx)*dtau*known1(nx)/(1.d0+B6(nx)*
     $           dx/2.d0))/(1.d0 - (B3(nx)*B6(nx)*Ywmax(nx)*dtau*dx/
     $           (2.d0*Wmax))/(1.d0 + B6(nx)*dx/2.d0) + B3(nx)*dtau*
     $           Ywmax(nx)/Wmax)

      Yapres(nx)=(known1(nx) + B6(nx)*dx/4.d0 * (Ywmax(nx)/Wmax *
     $             Wpres(nx)))/(1.d0 + B6(nx)*dx/4.d0)

      Ywpres(nx)=(Ywmax(nx)/Wmax *Wpres(nx))

      Twpres(nx)=(known4(nx) + B4(nx)*dtau*(Yapres(nx)-Ywpres(nx)) +
     $             B5(nx)*dtau/(1.d0 + B8(nx)*dx/2.d0) * (known2(nx) +
     $             B7(nx)*dx/2.d0 * (Ywpres(nx)-Yapres(nx))))/(1.d0 +
     $             B5(nx)*dtau - B5(nx)*B8(nx)*dtau*dx/(2.d0*(1.d0 +
     $             B8(nx)*dx/2.d0)))

      Tapres(nx)=(known2(nx) + B7(nx)*dx/4.d0*(Ywpres(nx) -
     $              Yapres(nx)) + B8(nx)*dx/4.d0*Twpres(nx))/
     $              (1.d0+B8(nx)*dx/4.d0)
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE BNDYREGEN
**********************************************************************
```

```
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to BNDYREGEN various subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ g,i,j,k,l,nx,cyclenum
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine BNDYREGEN.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the BNDYREGEN and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    Assign initial conditions before the beginning of REGENERATION
*-----------------------------------------------------------------------

      do i=1,nx
         Tapres(i)=Taring(i)
         Yapres(i)=Yaring(i)
         Twpres(i)=Twring(i)
         Wpres(i)=Wring(i)
         Ywpres(i)=Ywring(i)
         Yamax(i)=Yamaxring(i)
         Ywmax(i)=Ywmaxring(i)
      end do

      Tapres(nx)=Tabcregen
      Yapres(nx)=Yabcregen
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE COEFREGEN
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
```

```
* the MAIN DRIVER to COEFREGEN and various other subroutines easier.
*----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine COEFREGEN.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision rhod,rhos,rhoair,rhoevery,Q,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the COEFREGEN and transferred between subroutines.
*----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*----------------------------------------------------------------------
*   Assign air conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*----------------------------------------------------------------------
*   Step along the x direction, along the wheel depth
*----------------------------------------------------------------------

      do i=nx,1,-1
         Yaprev(i)=Yapres(i)
        Taprev(i)=Tapres(i)
         Twprev(i)=Twpres(i)
         Wprev(i)=Wpres(i)
         Ywprev(i)=Ywpres(i)
         TempKa(i)=2.7315d+2 + Taprev(i)
         TempKw(i)=2.7315d+2 + Twprev(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKa(i))))
         uw(i)=(6.4727d+2 - TempKw(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKw(i))*
     $        (6.4727d+2 - TempKw(i)))* (6.4727d+2 - TempKw(i)))/
     $        (TempKw(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKw(i))))
         PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
         PwsTw(i)=2.18167d+2/(10.d0 **uw(i))
 Yamaxprev(i)=6.22d-1*PwsTa(i)/Patm
         Ywmaxprev(i)=6.22d-1*PwsTw(i)/Patm
      end do
```

```
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

***********************************************************************
      SUBROUTINE REGENERATION
***********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to REGENERATION and various other subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/density/ rhod,rhos,rhoair,rhoevery
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine REGENERATION.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      real g,f,perw,perd,n,tautot
      double precision Ky
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision rhod,rhos,rhoair,rhoevery,Q,Hv,h,velpro,
     $   velreg,Wmax,Patm
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the REGENERATION and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension Ky(NMAX1)
      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
```

```
     $   B8(NMAX1)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $   known4(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    Stepping along in x, the plate depth, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every time step.
*-----------------------------------------------------------------------

      do i=nx,1,-1
         if (Wprev(i) .gt. Wmax) then
          Wprev(i)=Wmax
         end if
         TempKa(i)=2.7315d+2 + Taprev(i)
         TempKa(1)=2.7315d+2 + Tabcpro
         TempKw(i)=2.7315d+2 + Twprev(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $        TempKa(i))))
         uw(i)=(6.4727d+2 - TempKw(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKw(i))*
     $        (6.4727d+2 - TempKw(i)))* (6.4727d+2 - TempKw(i)))/
     $        (TempKw(i) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $        TempKw(i))))
         PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
         PwsTw(i)=2.18167d+2/(10.d0 **uw(i))
 Yamax(i)=6.22d-1*PwsTa(i)/Patm
         Ywmax(i)=6.22d-1*PwsTw(i)/Patm
*-----------------------------------------------------------------------
*    Calculate the mass transfer coefficient, a moisture content dependent
* material property.
*-----------------------------------------------------------------------
          Ky(i)=10.d0*(3.d-5*(Wprev(i)*Massdc)**2 - 3.d-4*
     $         (Wprev(i)*Massdc) + 6.d-4)
c         Ky(i)=10.d0*(5.8d-4 - 1.d-5*(1.d+3*Wprev(i)*Massdc))
*-----------------------------------------------------------------------
*    Calculate the heat of vaporization, a temperature dependent
* material property.
*-----------------------------------------------------------------------

         Hv=2502.d0-Cv/1.d+3*Taprev(i)
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------

         B3(i)=DBLE(n)*perimeter*Ky(i)*wheeldepth/(Massw*DBLE(f))
         B4(i)=Hv*Ky(i)*DBLE(n)*perimeter*wheeldepth/(Massw*
     $         (DBLE(1.-f)*Cs+DBLE(f)*(Cd+Cwr*Wprev(i))))
         B5(i)=DBLE(n)*perimeter*h*wheeldepth/(Massw*(DBLE(1.-f)*
     $         Cs+DBLE(f)*(Cd+Cwr*Wprev(i))))
         B6(i)=perimeter*Ky(i)/(rhoair*velpro*Areaair)
         B7(i)=perimeter*Ky(i)*Hv/(rhoair*velpro*Areaair*(Cda+
     $         Yaprev(i)*Cv))
         B8(i)=h*perimeter/(rhoair*velpro*Areaair*(Cda+
     $         Yaprev(i)*Cv))
      end do
*-----------------------------------------------------------------------
*    This closes the coefficient determination phase of the subroutine.  We
* now move on to the nodal solution of the supply-side phase.
```

```
*-----------------------------------------------------------------------
****************************************************************
*-----------------------------------------------------------------------
*  Maintain regenerative-side air boundary conditions
*-----------------------------------------------------------------------

      Yapres(nx)=Yabcregen
      Tapres(nx)=Tabcregen
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for last, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known3(nx)=Wprev(nx) + B3(nx)*dtau*(Yaprev(nx)-Ywprev(nx))

      known4(nx)=Twprev(nx) + dtau*B4(nx)*(Yaprev(nx)-Ywprev(nx)) +
     $           B5(nx)*dtau*(Taprev(nx) - Twprev(nx))
*-----------------------------------------------------------------------
*    Solve for desiccant conditions during the exhaust rotation for the
* last, channel edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(nx)=(known3(nx) + B3(nx)*dtau*Yapres(nx))/(1.d0 +
     $          B3(nx)*dtau*Ywmax(nx)/Wmax)

      Ywpres(nx)=Wpres(nx)*Ywmax(nx)/Wmax

      Twpres(nx)=(known4(nx) + B4(nx)*dtau*(Yapres(nx)-Ywpres(nx)) +
     $           B5(nx)*dtau*Tapres(nx))/(1.d0 + B5(nx)*dtau)
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for all interior
*  nodes for the exhaust rotation of the analysis.  These expressions
*  contain information gathered from previous time steps.
*-----------------------------------------------------------------------

      do i=nx-1,2,-1
         known1(i)=(B6(i)*dx/2.d0)*Ywpres(i+1) +
     $    (1.d0-B6(i)*dx/ 2.d0)*Yapres(i+1)

         known2(i)=B7(i)*dx/2.d0*(Ywpres(i+1) - Yapres(i+1)) +
     $             B8(i)*dx/2.d0*(Twpres(i+1) - Tapres(i+1)) +
     $             Tapres(i+1)

 known3(i)=Wprev(i) + B3(i)*dtau*(Yaprev(i)-Ywprev(i))

         known4(i)=Twprev(i) + dtau*B4(i)*(Yaprev(i)-Ywprev(i)) +
     $             B5(i)*dtau*(Taprev(i) - Twprev(i))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the exhaust-side
* rotation for all interior nodes
*-----------------------------------------------------------------------

         Wpres(i)=(known3(i) + B3(i)*dtau*known1(i)/(1.d0+B6(i)*dx/
     $           2.d0))/(1.d0 - (B3(i)*B6(i)*Ywmax(i)*dtau*dx/
     $           (2.d0*Wmax))/(1.d0 + B6(i)*dx/2.d0) + B3(i)*dtau*
     $           Ywmax(i)/Wmax)

         Yapres(i)=(known1(i) + B6(i)*dx/2.d0 * (Ywmax(i)/Wmax *
     $             Wpres(i)))/(1.d0 + B6(i)*dx/2.d0)

         Ywpres(i)=(Ywmax(i)/Wmax *Wpres(i))

         Twpres(i)=(known4(i) + B4(i)*dtau*(Yapres(i)-Ywpres(i)) +
     $             B5(i)*dtau/(1.d0 + B8(i)*dx/2.d0) * (known2(i) +
     $             B7(i)*dx/2.d0 * (Ywpres(i)-Yapres(i))))/(1.d0 +
     $             B5(i)*dtau - B5(i)*B8(i)*dtau*dx/(2.d0*(1.d0 +
     $             B8(i)*dx/2.d0)))
```

```
        Tapres(i)=(known2(i) + B7(i)*dx/2.d0*(Ywpres(i) - Yapres(i))
     $            + B8(i)*dx/2.d0*Twpres(i))/(1.d0+B8(i)*dx/2.d0)
      end do
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for first, channel
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known1(1)=(B6(1)*dx/4.d0)*Ywpres(2) +
     $    (1.d0-B6(1)*dx/4.d0)*Yapres(2)

      known2(1)=B7(1)*dx/4.d0*(Ywpres(2) - Yapres(2)) +
     $            B8(1)*dx/4.d0*(Twpres(2) - Tapres(2)) +
     $            Tapres(2)

      known3(1)=Wprev(1) + B3(1)*dtau*(Yaprev(1)-Ywprev(1))

      known4(1)=Twprev(1) + dtau*B4(1)*(Yaprev(1)-Ywprev(1)) +
     $            B5(1)*dtau*(Taprev(1) - Twprev(1))
*-----------------------------------------------------------------------
*    Solve for desiccant and air conditions on the exhaust-side rotation for
* the first, channel edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(1)=(known3(1) + B3(1)*dtau*known1(1)/(1.d0+B6(1)*
     $          dx/2.d0))/(1.d0 - (B3(1)*B6(1)*Ywmax(1)*dtau*dx/
     $          (2.d0*Wmax))/(1.d0 + B6(1)*dx/2.d0) + B3(1)*dtau*
     $          Ywmax(1)/Wmax)

      Yapres(1)=(known1(1) + B6(1)*dx/4.d0 * (Ywmax(1)/Wmax *
     $            Wpres(1)))/(1.d0 + B6(1)*dx/4.d0)

      Ywpres(1)=(Ywmax(1)/Wmax *Wpres(1))

      Twpres(1)=(known4(1) + B4(1)*dtau*(Yapres(1)-Ywpres(1)) +
     $            B5(1)*dtau/(1.d0 + B8(1)*dx/2.d0) * (known2(1) +
     $            B7(1)*dx/2.d0 * (Ywpres(1)-Yapres(1))))/(1.d0 +
     $            B5(1)*dtau - B5(1)*B8(1)*dtau*dx/(2.d0*(1.d0 +
     $            B8(1)*dx/2.d0)))

      Tapres(1)=(known2(1) + B7(1)*dx/4.d0*(Ywpres(1) -
     $             Yapres(1)) + B8(1)*dx/4.d0*Twpres(1))/
     $             (1.d0+B8(1)*dx/4.d0)
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

***********************************************************************
      SUBROUTINE XFER
***********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to XFER and various other subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ g,i,j,k,l,nx,cyclenum
      common/airprops/ TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine XFER.
* Variables declared here are used throughout the subroutine as well as
```

```
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1
      double precision TempKo,TempKa,TempKw,Yamax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw,ratio
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the XFER and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)

      dimension TempKo(NMAX1),TempKa(NMAX1),TempKw(NMAX1),Yamax(NMAX1),
     $   Ywmax(NMAX1),Yamaxprev(NMAX1),Ywmaxprev(NMAX1),ua(NMAX1),
     $   uw(NMAX1),PwsTa(NMAX1),PwsTw(NMAX1),ratio(NMAX1)
      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),Yapres(NMAX1),
     $   Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),Ywprev(NMAX1),
     $   Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),Wring(NMAX1),
     $   Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),Yamaxring(NMAX1),
     $   Ywmaxring(NMAX1)
*-----------------------------------------------------------------------
*    After subroutines are completed, air conditions labeled 'present'
* are transferred to 'previous' for nodal solution purposes.
*-----------------------------------------------------------------------

         do i=1,nx
            Twprev(i)=Twpres(i)
            Wprev(i)=Wpres(i)
            Ywprev(i)=Ywpres(i)
            Yaprev(i)=Yapres(i)
            Taprev(i)=Tapres(i)
            Yamaxprev(i)=Yamax(i)
            Ywmaxprev(i)=Ywmax(i)
         end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE EFFICIENCY
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to EFFICIENCY and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/time/ tautot,dtau
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/heat/ hgpres,hgprev,hgpro,hgregen,hgtot
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Ywpres,Yapres,Tapres,
     $   Twprev,Wprev,Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,
     $   Yaring,Taring,Yamaxring,Ywmaxring
      common/solution/ Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,
     $   Twtot,Wtot,Ywtot
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine EFFICIENCY.
* Variables declared here are used throughout the subroutine as well as
```

```
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX1,NMAX2
      real tautot
      double precision dtau
      double precision eff1,eff2,eff3
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision hgpres,hgprev,hgpro,hgregen,hgtot,haprotot,
     $  haproregen
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Yapres,Tapres,Twprev,Wprev,
     $   Ywprev,Yaprev,Taprev,Twring,Wring,Ywring,Yaring,Taring,
     $   Yamaxring,Ywmaxring
      double precision Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,
     $   Twtot,Wtot,Ywtot
      double precision Ppro,Ptot,Pregen,Xpro,Xtot,Xregen,
     $  Ypro,Ytot,Yregen
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the EFFICIENCY and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=100)
      parameter (NMAX2=2500)

      dimension Twpres(NMAX1),Wpres(NMAX1),Ywpres(NMAX1),
     $   Yapres(NMAX1),Tapres(NMAX1),Twprev(NMAX1),Wprev(NMAX1),
     $   Ywprev(NMAX1),Yaprev(NMAX1),Taprev(NMAX1),Twring(NMAX1),
     $   Wring(NMAX1),Ywring(NMAX1),Yaring(NMAX1),Taring(NMAX1),
     $   Yamaxring(NMAX1),Ywmaxring(NMAX1)
      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Ywtotpro(NMAX2),
     $   Yatotpro(NMAX2),Tatotpro(NMAX2),Twtotreg(NMAX2),
     $   Wtotreg(NMAX2),Ywtotreg(NMAX2),Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    Determine coefficients used in solving for enthalpy
*-----------------------------------------------------------------------

 Xpro= 374.12d0-Tabcpro
 Xtot= 374.12d0-Ta
 Xregen= 374.12d0-Tabcregen
 Ypro= Xpro*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $        Xpro**2)*Xpro)/((Tabcpro+273.15d0)*(1.d0+
     $        2.1878462d-3*Xpro))
 Ytot= Xtot*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $   Xtot**2)*Xtot)/((Ta+273.15d0)*(1.d0+2.1878462d-3*
     $   Xtot))
 Yregen= Xregen*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $  Xregen**2)*Xregen)/((Tabcregen+273.15d0)*(1.d0+
     $  2.1878462d-3*Xregen))

 Ppro = 14.696d0*218.167d0/(10.d0**Ypro)
 Ptot = 14.696d0*218.167d0/(10.d0**Ytot)
 Pregen =  14.696d0*218.167d0/(10.d0**Yregen)
 Xpro = DLOG10(Ppro)
 Xtot = DLOG10(Ptot)
 Xregen = DLOG10(Pregen)
*-----------------------------------------------------------------------
*   Calculate enthaply associated with the air streams
*-----------------------------------------------------------------------

 hgpro = (1105.9387d0+( 32.756807d0+(4.6198474d0+(0.20672996d0+
     $  (-0.5411693d0+(0.49241362d0-0.17884885d0*Xpro)*Xpro)
     $  *Xpro)*Xpro)*Xpro)*Xpro)/.4299d0
 hgtot =(1105.9387d0+( 32.756807d0+(4.6198474d0+(0.20672996d0+
     $  (-0.5411693d0+(0.49241362d0-0.17884885d0*Xtot)*Xtot)*
     $  Xtot)*Xtot)*Xtot)*Xtot)/.4299d0
```

```
 hgregen =(1105.9387d0+( 32.756807d0+(4.6198474d0+(0.20672996d0+
     $  (-0.5411693d0+(0.49241362d0-0.17884885d0*Xregen)
     $  *Xregen)* Xregen)*Xregen)*Xregen)*Xregen)/.4299d0

610   haprotot=Cda*(Tabcpro-Ta) + (Yabcpro*hgpro-
     $ Ya*hgtot)
      haproregen=Cda*(Tabcpro-Tabcregen) + (Yabcpro*hgpro-Yabcregen*
     $  hgregen)
*-----------------------------------------------------------------------
*    Calculate sensible efficiency
*-----------------------------------------------------------------------

      eff1= (Tabcpro-Ta)/(Tabcpro-Tabcregen)
*-----------------------------------------------------------------------
*    Calculate latent efficiency
*-----------------------------------------------------------------------

      eff2=(Yabcpro*hgpro-Ya*hgtot)/
     $  (Yabcpro*hgpro-Yabcregen*hgregen)
*-----------------------------------------------------------------------
*    Calculate total efficiency
*-----------------------------------------------------------------------

      eff3= haprotot/haproregen
*-----------------------------------------------------------------------
*    Open the output data file and record efficiency values
*-----------------------------------------------------------------------

      open(44,file='wheel_foam.dat')
      write(44,*)
      write(44,*) 'After',cyclenum,'cycles,'
      write(44,*)
      write(44,*) 'Efficiencies are:'
      write(44,*)
      write(44,*) 'Sensible =',eff1
      write(44,*) 'Latent =',eff2
      write(44,*) 'Total =',eff3
      write(44,*)
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end

***********************************************************************
      SUBROUTINE OUTPUT
***********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to OUTPUT and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ g,i,j,k,l,nx,cyclenum
      common/geometry/ Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,n,pressuredrop,perimeter
      common/specificheat/ Cwr,Cda,Cv,Cd,Cs,Cw
      common/time/ tautot,dtau
      common/distance/ dx,ztot
      common/properties/ Q,Hv,h,velpro,velreg,Wmax,Patm
      common/solution/ Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,
     $   Twtot,Wtot,Ywtot
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine OUTPUT.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
```

```
*-----------------------------------------------------------------------

      integer i,j,k,l,nx,cyclenum,NMAX2
      real g,f,perw,perd,n,tautot
      double precision Do,Di,wheeldepth,height,base,thickd,thicks,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cda,Cv,Cd,Cs,Cw
      double precision dx,ztot,dtau
      double precision Q,Hv,h,velpro,velreg,Wmax,Patm
      double precision Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot,
     $   Twtot,Wtot,Ywtot
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the OUTPUT and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX2=2500)

      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Ywtotpro(NMAX2),
     $   Yatotpro(NMAX2),Tatotpro(NMAX2),Twtotreg(NMAX2),
     $   Wtotreg(NMAX2),Ywtotreg(NMAX2),Yatotreg(NMAX2),
     $   Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    Record elapsed time and pressure drop as well as final nodal spacing
* in the output file.
*-----------------------------------------------------------------------

      write(44,*) 'Rotation time = ',DBLE(tautot)*2.d0,'seconds'
      write(44,*)
      write(44,*) 'Pressure Drop = ',pressuredrop,'m water'
      write(44,*)
      write(44,*) 'dx = ',dx,'m'
      write(44,*)
*-----------------------------------------------------------------------
*    Record exit supply and exhaust air and desiccant data at every 100
* time steps, for a more detailed look at the rotary wheel exchanger behavior.
*-----------------------------------------------------------------------

       do j=1,cyclenum,100
 write(44,*) 'CYCLE #:',j
 write(44,*) 'SUPPLY'
 write(44,*) 'Tair (Celcius)= ',REAL(Tatotpro(j))
 write(44,*) 'Yair (kg water/kg air)= ',REAL(Yatotpro(j))
 write(44,*) 'Tbed (Celcius)= ',REAL(Twtotpro(j))
 write(44,*) 'Ybed (kg water/kg desiccant)= ',REAL(Wtotpro(j))
        write(44,*) 'Ypores (kg water/kg air)= ',REAL(Ywtotpro(j))
 write(44,*) 'EXHAUST'
 write(44,*) 'Tair (Celcius)= ',REAL(Tatotreg(j))
 write(44,*) 'Yair (kg water/kg air)= ',REAL(Yatotreg(j))
 write(44,*) 'Tbed (Celcius)= ',REAL(Twtotreg(j))
 write(44,*) 'Ybed (kg water/kg desiccant)= ',REAL(Wtotreg(j))
        write(44,*) 'Ypores (kg water/kg air)= ',REAL(Ywtotpro(j))
 write(44,*)
       end do
*-----------------------------------------------------------------------
*   Return to the MAIN DRIVER
*-----------------------------------------------------------------------

      return
      end
```

Appendix E

Polyvinyl Alcohol Fixed Plate Total Enthalpy Exchanger Performance Estimation Code

```
      PROGRAM PLATE
**********************************************************************
*      Performance Estimates of A
*  Counterflow Plate Total Enthalpy Exchanger
*    Using a Porous (Adsorbing) Polymer Desiccant Sheet
*                              by
*                        Christie Staton
*  1997
*
**********************************************************************
*
*
*    This program estimates the sensible, latent, and total performance
* efficiencies of a  Counterflow Fixed Plate Total Enthalpy Exchanger
* using a open-cell selectively diffusive polymer desiccant sheet.
*  A solution is obtained via an analysis of the exiting air
* conditions.
*
*_______________________________________________________________________
*    We begin with the MAIN DRIVER of the program.  This is the portion of
* the code form which all subroutines are called.  The MAIN DRIVER organizes
* the operation of the subroutines in such a manner as to get the
* performance estimation accomplished in the most efficient way.
*_______________________________________________________________________

      implicit none
*________________________________________________________________________
*   Declare all variables that are used in the MAIN DRIVER and elsewhere.
* Variables declare here are used throughout the program; they are used in
* several different subroutines.
*________________________________________________________________________

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX2,NMAX3
      integer proconverge,regconverge
      real f,timetot,dt
      double precision width,depth,height,thickd,perw,perd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
```

```
     $   pressuredrop,perimeter
      double precision Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      double precision dx,dy
      double precision rhod,rhos,rhowater,rhopro,rhoreg,rhoevery,Q,
     $   hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      double precision hgpro,hgregen,hgtot
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8,B9,B10,B11
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
      double precision Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Tatot,Ta,Ya,Yatot,Yatotreg,Tatotreg,Twtotreg,Wtotreg,Ywtotreg
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the MAIN DRIVER and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=20000)
      parameter (NMAX2=90000)
      parameter (NMAX3=25)

      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1),B9(NMAX1,NMAX3),B10(NMAX1,NMAX3),B11(NMAX1,NMAX3)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $    known4(NMAX1)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Ywtotpro(NMAX2),
     $   Yatotpro(NMAX2),Tatotpro(NMAX2),TWtotreg(NMAX2),
     $   Wtotreg(NMAX2),Ywtotreg(NMAX2),Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to various subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/criteria/ proconverge,regconverge
      common/geometry/ width,depth,height,thickd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,pressuredrop,perimeter
      common/specificheat/ Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      common/time/ timetot,dt
      common/distance/ dx,dy
      common/properties/ Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      common/density/ rhod,rhos,rhowater,rhopro,rhoreg,rhoevery
      common/heat/ hgpro,hgregen,hgtot
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8,B9,B10,B11
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
      common/solution/ Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot
*----------------------------------------------------------------------
*    We begin with the body of the MAIN DRIVER of the program, the portion
```

```
*  of the program from which all subroutines are called.
*    First a flag variable, ll, is initialized.  This flag is an indicator
* used in the stability criteria check for the finite differencing scheme.
* It is further discussed in subroutine PROCESS.
*    The subroutine PROPERTY is called. This subroutine reads user input
* data dealing with plate geometry and air stream conditions.  The
* input data files containing all density, specific heat, boundary
* conditions of the supply and exhaust air, initial conditions of the
* desiccant bed, plate geometry, and air property information are read in
* the section of the program.
*-----------------------------------------------------------------------

      ll=0
10    call PROPERTY
*-----------------------------------------------------------------------
*   Since flag ll is used in subroutine PROPERTY, is has to be reset after
* every visit to that subroutine.  At this point the variable that determines
* how long the run will simulate is also determined.  Variable timetot defines
* the total elapsed time (in seconds) modeled by the program.
*-----------------------------------------------------------------------

      ll=0
      timetot=300.
*-----------------------------------------------------------------------
*    After determining the total time to be modeled as well as the time step
* to be used in the analysis, a time loop is introduced.  Keep in mind that
* this model is analyzed by stepping along in distance within a specified space
* in time.  This first loop adjusts the time steps after each analysis along
* the length of the supply and exhaust sides of the plate have been completed
* and met the convergence criteria as outlined in subroutine CONVERGENCE.
*-----------------------------------------------------------------------

      do j=INT(dt/dt),INT(timetot/dt),INT(dt/dt)
*-----------------------------------------------------------------------
*    Flags z,l,and zz are set along with variable proconverge and regconverge.
* These variables are also flags used to determine whether convergence has
* been met within the supply and exhaust exiting conditions.
*-----------------------------------------------------------------------

         z=0
         l=0
         zz=0
         proconverge=0
         regconverge=0
*-----------------------------------------------------------------------
*    Next, variables representing total air and desiccant material exiting
* conditions for both the supply and exhaust side are initialized.  At every
* time step, these conditions are updated, and therefore must be reset as each
* time step is updated.
*-----------------------------------------------------------------------

 Tatot=0.d0
 Yatot=0.d0
         Ta=0.d0
         Ya=0.d0
 Tatotpro(j)=0.d0
 Yatotpro(j)=0.d0
 Twtotpro(j)=0.d0
 Wtotpro(j)=0.d0
         Ywtotpro(j)=0.d0
         Tatotreg(j)=0.d0
 Yatotreg(j)=0.d0
 Twtotreg(j)=0.d0
 Wtotreg(j)=0.d0
         Ywtotreg(j)=0.d0
*-----------------------------------------------------------------------
*    The fist step in solving for the conditions associated with the supply
* air stream within the particular time step is the calling of subroutine
```

```
* COEFPRO.  This subroutine updates air properties such as maximum moisture
* capacity at each time step.  Z is a counter that helps keep track of which
* air stream is being analyzed.  If z=1, the supply side is being solved.
* If z=2, the exhaust side is being solved.  This is referenced in subroutine
* CONVERGE.
*-----------------------------------------------------------------------

 100    call COEFPRO
        z=1
*-----------------------------------------------------------------------
*    Subroutine PROCESS actually solves, stepping in distance, for air and
* desiccant conditions at each distance step along the length of the supply-
* side plate.  In addition, this subroutine also solves for desiccant
* conditions within the thickness of the bed at each distance step.
*-----------------------------------------------------------------------

        call PROCESS
*-----------------------------------------------------------------------
*    In PROCESS there is a stability criteria check.  If the indicator ll is
* set, the solution must be backed up and appropriate changes made to the step
* sizes.  Once these are made, the solution must begin again.
*-----------------------------------------------------------------------

        if (ll .eq. 1) then
          go to 10
        end if
*-----------------------------------------------------------------------
*    Subroutine CONVERGENCE is used at this moment to check for convergence
* within the exiting supply air conditions.
*-----------------------------------------------------------------------

        call CONVERGENCE
*-----------------------------------------------------------------------
*    Before going to the next solution segment, PROCESS, air conditions
* values need to be transferred from a status of 'present' condition to
* 'previous' condition.  These conditions must be specifically defined
* in order to perform the finite difference solution correctly.
*-----------------------------------------------------------------------

        call XFER
*-----------------------------------------------------------------------
*    After convergence of the supply side solution has been determined,
* conditions for air and desiccant are saved for that particular time step.
*-----------------------------------------------------------------------
         Tatotpro(j)=Tapros(nx)
 Yatotpro(j)=Yapros(nx)
 Twtotpro(j)=Twpres(nx,1)
 Wtotpro(j)=Wpres(nx,1)
         Ywtotpro(j)=Ywpres(nx,1)
 Tatot=Tatotpro(j)
 Yatot=Yatotpro(j)
         Ta=Tatot
         Ya=Yatot
*-----------------------------------------------------------------------
*    The fist step in solving for the conditions associated with the exhaust
* air stream within the particular time step is the calling of subroutine
* COEFREGEN.  This subroutine updates air properties such as maximum moisture
* capacity at each time step.  Z is a counter that helps keep track of which
* air stream is being analyzed.  If z=1, the supply side is being solved.
* If z=2, the exhaust side is being solved.  This is referenced in subroutine
* CONVERGE.
*-----------------------------------------------------------------------

 200    call COEFREGEN
        z=2
*-----------------------------------------------------------------------
*    Subroutine REGENERATION actually solves, stepping in distance, for air
* and desiccant conditions at each distance step along the length of the
```

```
* supply-side plate.  In addition, this subroutine also solves for desiccant
* conditions within the thickness of the bed at each distance step.
*-----------------------------------------------------------------------

         call REGENERATION
*-----------------------------------------------------------------------
*    Subroutine CONVERGENCE is used at this moment to check for convergence
* within the exiting exhaust air conditions.
*-----------------------------------------------------------------------

         call CONVERGENCE
*-----------------------------------------------------------------------
*    Before going to the next solution segment, PROCESS, air conditions
* values need to be transferred from a status of 'present' condition to
* 'previous' condition.  These conditions must be specifically defined
* in order to perform the finite difference solution correctly.
*-----------------------------------------------------------------------

         call XFER
*-----------------------------------------------------------------------
*    If one or both of the exiting air stream conditions have not converged,
* it will be determined here.  If convergence has not been met, the solution
* is performed again.  If convergence has been met, the solution proceeds
* to the next time step.
*-----------------------------------------------------------------------

        if ((proconverge .eq. 0)  .or. (regconverge .eq. 0)) then
            zz=1
c            print *, 'go to 100'
c            print *, j,proconverge,regconverge
            go to 100
         end if
*-----------------------------------------------------------------------
*    After convergence of the supply side solution has been determined,
* conditions for air and desiccant are saved for that particular time step.
*-----------------------------------------------------------------------

         Tatotreg(j)=Taregs(1)
 Yatotreg(j)=Yaregs(1)
 Twtotreg(j)=Twpres(1,ny)
 Wtotreg(j)=Wpres(1,ny)
         Ywtotreg(j)=Ywpres(1,ny)
       end do
*-----------------------------------------------------------------------
*     Once the time loop has been completed it is time to determine the
* performance efficiencies resulting form the final exiting air stream
* conditions.  These efficiencies are calculated in subroutine  EFFICIENCY.
*-----------------------------------------------------------------------

      call EFFICIENCY
*-----------------------------------------------------------------------
*    Once the efficiency conditions have been determined for the pre-
* determined test time, timetot, outlet conditions of both
* the supply stream and the exhaust stream are written to an output data
* file.  Outlet conditions include the humidity ratios of both the air stream
* and the desiccant bed as well as the temperatures of both the air stream
* and the desiccant bed.
*-----------------------------------------------------------------------

      call OUTPUT
*-----------------------------------------------------------------------
*    The completion of the MAIN DRIVER signifies the completion of the
* Counterflow Plate Total Enthalpy Exchange performance estimation program.
*-----------------------------------------------------------------------

      END
```

```
**********************************************************************
      SUBROUTINE PROPERTY
**********************************************************************
*
*  The primary function of this subroutine is read in data dealing with
* channel geometry, specific heat, boundary conditions of the supply and
* exhaust air, initial conditions of the desiccant bed, plate geometry,
* and air property information.  All of this information is read through
* direct input via a series of data input files.  Once the proper information
* is  obtained,  geometric information leading to the solution of performance
* efficiencies is calculated.
*_______________________________________________________________________
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to PROPERTY and various other subroutines easier.
*_______________________________________________________________________

      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/geometry/ width,depth,height,thickd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,pressuredrop,perimeter
      common/specificheat/ Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      common/time/ timetot,dt
      common/distance/ dx,dy
      common/properties/ Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      common/density/ rhod,rhos,rhowater,rhopro,rhoreg,rhoevery
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*_______________________________________________________________________
*   Declare all variables that are used in the subroutine PROPERTY.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*_______________________________________________________________________

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      real f,timetot,dt
      double precision width,depth,height,thickd,perw,perd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision gravity,Dh,Red,kviscair,friction,tothead,
     $   spgravity
      double precision Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      double precision dx,dy
      double precision rhod,rhos,rhowater,rhopro,rhoreg,rhoevery,Q,
     $   hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*_______________________________________________________________________
*     Initialize parameters used in the dimensioning of all arrays used in
* the PROPERTY and transferred between subroutines.
*_______________________________________________________________________

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
```

```
     $    Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $    Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $    Yaregv(NMAX1)
*-----------------------------------------------------------------------
*    If the stability criteria on subroutine PROCESS has not been met,
* an adjustment is made to the distancing step size along the depth of
* the plate.
*-----------------------------------------------------------------------

      if (ll .eq. 1) then
         nx=INT(1.055*REAL(nx))
         dx=depth/dble(nx-1)
c         print *, 'changing nx',nx
         go to 120
      end if
*-----------------------------------------------------------------------
*    Read in the input data files
*  spheat.dat:
*   Cwr=  reference specific heat (sp. heat of water)
*   Cdapro=  specific heat of dry supply air
*   Cdapro=  specific heat of dry exhaust air
*   Cv=   specific heat of water vapor
*   Cd=  specific heat of desiccant
*   Cs=  specific heat of support substrate
*   Cw=  specific heat of wheel
*   spgravity=  specific gracity of water
*   gravity=  gravity
*  density.dat:
*   rhod=  density of desiccant
*   rhos=  density of support material, if any
*   rhoevery=  density of dehumidifier
*   rhowater=  density of water
*   rhopro=  density of supply air
*   rhoreg=  density of exhaust air
*   kviscair=  kinematic viscosity of air
*  design.dat:
*   height=  height of air channel
*   width=  width of plate
*   depth=  depth of dehumidifier
*   thickd=  thickness of desiccant
*   velpro=  velocity of supply air
*   velreg=  velocity of exhaust air
*   Patm=  atmospheric pressure
*  property.dat:
*   Q=  heat of adsorption
*   Wmax=  maximum moisture capacity of desiccant
*   Dab=  mass diffusivity
*   alpha=  thermal diffusivity
*  boundary.dat:
*   Twic=  initial temperature condition of the dehumidifier
*   Wic=  initial moisture content condition of the dehumidifier
*   Tabcpro=  air temperature boundary condition for supply side
*   Yabcpro=  air moisture content boundary condition for supply side
*   Tabcregen=  air temperature boundary condition for exhaust side
*   Yabcregen=  air moisture content boundary condition for exhaust side
*-----------------------------------------------------------------------

       print *, 'reading input files'
       open(2,file='spheat.dat')
       read(2,*) Cwr,Cdapro,Cdareg,Cv,Cd,Cs,spgravity,gravity
       open(3,file='density.dat')
       read(3,*) rhod,rhos,rhoevery,rhowater,rhopro,rhoreg,kviscair
       open(4,file='design.dat')
       read(4,*) height,width,depth,thickd,velpro,velreg,Patm
       open(7,file='property.dat')
       read(7,*) Q,Wmax,Dab,alpha
       open(8,file='boundary.dat')
       read(8,*) Twic,Wic,Tabcpro,Yabcpro,Tabcregen,Yabcregen
```

```
*-----------------------------------------------------------------------
*    Using information obtained in the data files regarding plate
* geometry, desiccant and air properties, etc.  Determine area available for
* heat and mass transfer and pressure drop.
*-----------------------------------------------------------------------

        perimeter=2.d0*width
 Areatot=width*(thickd+height)
 Areaair=width*height
        Dh=4.d0*Areaair/perimeter
        Red=(velpro*Dh)/kviscair
        friction=64.d0/Red
        tothead=(4.d-1*velpro**2/(2.d0*gravity)) + (friction*
     $       depth/Dh*velpro**2/(2.d0*gravity)) + (1.d0*
     $       velpro**2/(2.d0*gravity))
        pressuredrop=rhopro*gravity*tothead
        pressuredrop=pressuredrop/spgravity
*-----------------------------------------------------------------------
*    Determine average heat transfer coefficient for both air streams:
* equations obtained from Incropera and Dewitt.
*-----------------------------------------------------------------------

      hpro=7.5d0*alpha*rhod*Cd/(4.d0*Areaair/perimeter)
      hreg=7.5d0*alpha*rhod*Cd/(4.d0*Areaair/perimeter)
*-----------------------------------------------------------------------
*    Determine actual length of elements and length of time steps
* Example: dx = step length along length of plate; nx = number of nodes
* along length of plate.
*-----------------------------------------------------------------------

      dt=0.0125
      nx=20
      dx=depth/dble(nx-1)
120   ny=4
140   dy=thickd/dble(ny-1)
*-----------------------------------------------------------------------
*    The second stability criteria is checked here.  If it is not met,
* an adjustment is made in the time step size, dt.
*-----------------------------------------------------------------------

       if (alpha*DBLE(dt)/(DBLE(dy)**2) .ge. .5d0) then
         dt=REAL(.5*dt)
c         print *, 'changing dt',dt
c         print *, alpha*DBLE(dt)/(DBLE(dy)**2)
         go to 140
       end if
*       print *, 'Fourier=',alpha*DBLE(dt)/(DBLE(dy)**2)
*-----------------------------------------------------------------------
*    Calculate areas, volumes, weights, and mass fractions needed to complete
* the solution.
*-----------------------------------------------------------------------
*    Area of the desiccant(per plate):
*-----------------------------------------------------------------------

      Aread=width*thickd
      Massdc=Aread*depth*rhod
*-----------------------------------------------------------------------
*    Calculate the percentage of desiccant in dehumidifier.  For now, it
* is assumed that only desiccant is used in the construction of the
* dehumidifier - ie. there is no support material.
*-----------------------------------------------------------------------

      perd=Aread/(Areatot-Areaair)
*-----------------------------------------------------------------------
*    Tot volume of plate, mass of dehumidifier, mass fraction of desiccant
* in the dehumidifier.
*-----------------------------------------------------------------------
```

```
      Voltot=Aread*depth
      Volw=Voltot
      Massw=Volw*(rhod*perd + rhos*(1.d0-perd))
      f=REAL(Voltot*perd*rhod/Massw)
*-----------------------------------------------------------------------
* Assign initial values to each node in the plate
*-----------------------------------------------------------------------
*   Along the depth...
*-----------------------------------------------------------------------
      do i=1,nx
         Tapros(i)=Tabcpro
         Yapros(i)=Yabcpro
         Taregs(i)=Tabcregen
         Yaregs(i)=Yabcregen
*-----------------------------------------------------------------------
*   And through the thickness...
*-----------------------------------------------------------------------

         do ii=1,ny
            TempKw(i,ii)=2.7315d+2 + Twic
            uw(i,ii)=(6.4727d+2 - TempKw(i,ii))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKw(i,ii))*
     $        (6.4727d+2 - TempKw(i,ii)))* (6.4727d+2 - TempKw(i,ii)))/
     $        (TempKw(i,ii) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $        TempKw(i,ii))))
            PwsTw(i,ii)=2.18167d+2/(10.d0 **uw(i,ii))
            Ywmax(i,ii)=6.22d-1*PwsTw(i,ii)/Patm

            Twpres(i,ii)=Twic
    Wpres(i,ii)=Wic
            Ywpres(i,ii)=Wpres(i,ii)*Ywmax(i,ii)/Wmax
         end do
      end do

*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE COEFPRO
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to COEFPRO and various other subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/properties/ Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine COEFPRO.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      double precision Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
```

```
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the COEFPRO and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
*-----------------------------------------------------------------------
*   Assign air conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*-----------------------------------------------------------------------
*   Step along the x direction, along the plate depth
*-----------------------------------------------------------------------

      do i=1,nx
            Yaprov(i)=Yapros(i)
            Taprov(i)=Tapros(i)
            TempKa(i)=2.7315d+2 + Taprov(i)
            ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $           (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $           (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $           (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $           TempKa(i))))
            PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
            Yamaxprev(i)=6.22d-1*PwsTa(i)/Patm
*-----------------------------------------------------------------------
*   If this is the first time COEFPRO has been called within the time step,
* assign desiccant conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*-----------------------------------------------------------------------
*   Step along the y direction, along the material thickness
*-----------------------------------------------------------------------

         if (zz .eq. 0) then
            do ii=1,ny
               Twprev(i,ii)=Twpres(i,ii)
               Wprev(i,ii)=Wpres(i,ii)
               Ywprev(i,ii)=Ywpres(i,ii)

               TempKw(i,ii)=2.7315d+2 + Twprev(i,ii)
               uw(i,ii)=(6.4727d+2 - TempKw(i,ii))  *(3.2437814d0 +
     $                 (5.86826d-03 + 1.1702379d-08*(6.4727d+2 -
     $                 TempKw(i,ii))*(6.4727d+2 - TempKw(i,ii)))*
     $                 (6.4727d+2 - TempKw(i,ii)))/(TempKw(i,ii) *
     $                 (1.d0+2.1878462d-03 *(6.4727d+2 - TempKw(i,ii))))
               PwsTw(i,ii)=2.18167d+2/(10.d0 **uw(i,ii))
               Ywmaxprev(i,ii)=6.22d-1*PwsTw(i,ii)/Patm
             end do
          end if
      end do
*-----------------------------------------------------------------------
*   Make sure that boundary conditions for the supply air stream  continue
* to be used.
*-----------------------------------------------------------------------

      Yapros(1)=Yabcpro
      Tapros(1)=Tabcpro
```

```
*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE PROCESS
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to PROCESS and various other subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/geometry/ width,depth,height,thickd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,pressuredrop,perimeter
      common/specificheat/ Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      common/time/ timetot,dt
      common/distance/ dx,dy
      common/properties/ Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      common/density/ rhod,rhos,rhowater,rhopro,rhoreg,rhoevery
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8,B9,B10,B11
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine PROCESS.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      real f,timetot,dt
      double precision Ky,velmass
      double precision width,depth,height,thickd,perw,perd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      double precision dx,dy
      double precision rhod,rhos,rhowater,rhopro,rhoreg,rhoevery,Q,Hv,
     $   hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8,B9,B10,B11
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the PROCESS and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension Ky(NMAX1),Hv(NMAX1),velmass(NMAX1,NMAX3)
      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
```

```
     $   B8(NMAX1),B9(NMAX1,NMAX3),B10(NMAX1,NMAX3),B11(NMAX1,NMAX3)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $    known4(NMAX1)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
*-----------------------------------------------------------------------
*    Stepping along in x, the plate depth, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every iteration/time step.
*-----------------------------------------------------------------------

      do i=1,nx
         TempKa(i)=2.7315d+2 + Taprov(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKa(i))))
         PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
         Yapromax(i)=6.22d-1*PwsTa(i)/Patm
         if (j .eq. 1) then
             Yaregmax(i)=Yapromax(i)
         end if
*-----------------------------------------------------------------------
*    Stepping along in y, the plate thickness, calculate the maximum air
* moisture content.  This is a property that is temperature dependent and
* therefore must be re-calculated after every iteration/time step.
*-----------------------------------------------------------------------

         do ii=1,ny
            TempKw(i,ii)=2.7315d+2 + Twpres(i,ii)
            uw(i,ii)=(6.4727d+2 - TempKw(i,ii))  *(3.2437814d0 +
     $               (5.86826d-03 + 1.1702379d-08*(6.4727d+2 -
     $               TempKw(i,ii))*(6.4727d+2 - TempKw(i,ii)))*
     $               (6.4727d+2 - TempKw(i,ii)))/(TempKw(i,ii) *
     $               (1.d0+2.1878462d-03 *(6.4727d+2 - TempKw(i,ii))))
            PwsTw(i,ii)=2.18167d+2/(10.d0 **uw(i,ii))
            Ywmax(i,ii)=6.22d-1*PwsTw(i,ii)/Patm
          end do
*-----------------------------------------------------------------------
*    Calculate the mass transfer coefficient, a moisture content dependent
* material property.
*-----------------------------------------------------------------------

         Ky(i)=10.d0*(6855.d-1*(Wprev(i,1)*Massdc)**2 - 362.d-1*
     $         (Wprev(i,1)*Massdc)+.0873d0)
*-----------------------------------------------------------------------
*    Calculate the heat of vaporization, a temperature dependent
* material property.
*-----------------------------------------------------------------------

         Hv(i)=2502.d0-Cv/1.d+3*Taprov(i)
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------

 B3(i)=Ky(i)/(thickd*rhod*DBLE(f))
         B4(i)=Hv(i)*Ky(i)/(thickd*rhoevery*(DBLE(1.-f)*Cs+DBLE(f)*
     $         (Cd*Twprev(i,1)+Wprev(i,1)*(Hv(i)-Q))))
         B5(i)=hpro/(thickd*rhoevery*(DBLE(1.-f)*Cs+DBLE(f)*
     $         (Cd*Twprev(i,1)+Wprev(i,1)*(Hv(i)-Q))))
         B6(i)=perimeter*Ky(i)/(rhopro*velpro*Areaair)
         B7(i)=perimeter*Ky(i)*Hv(i)/(rhopro*velpro*Areaair*
     $         (Cdapro+Yaprov(i)*Cv))
```

```
        B8(i)=hpro*perimeter/(rhopro*velpro*Areaair*(Cdapro+
     $         Yaprov(i)*Cv))
*-----------------------------------------------------------------------
*   Determine the speed at which the  moisture travels through the
* thickness of the desiccant plate on a node-by-node basis.  If this is
* the initial time step, a very small value is assigned since the initial
* bed conditions indicate no mass gradient through the plate thickness.
*-----------------------------------------------------------------------
          do ii=1,ny
            velmass(i,ii)=Massdc*Dab*(Wpres(i,ii)-Wpres(i,ii+1))/(dy*
     $                 Voltot*rhowater)
            if (j .eq. 1) then
               velmass(i,ii)=1.d-12
            end if
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------

            B9(i,ii)=DBLE(dt)*Dab/(dy**2)
            B10(i,ii)=alpha*DBLE(dt)/(dy**2)
            B11(i,ii)=velmass(i,ii)*DBLE(dt)/(2.d0*dy)
          end do
       end do
*-----------------------------------------------------------------------
*   Test for stability criteria.  If it is not met, flag variable ll.
*-----------------------------------------------------------------------

       do i=1,nx
          do ii=1,ny
             if (alpha*DBLE(dt)/DBLE(dy**2) .le.
     $                    velmass(i,ii)/(2.d0*dx)) then
               ll=1
            end if
          end do
       end do
*-----------------------------------------------------------------------
*    This closes the coefficient determination phase of the subroutine.  We
* now move on to the nodal solution of the supply-side phase.
*-----------------------------------------------------------------------
************************************************************************
*-----------------------------------------------------------------------
*  Maintain supply-side air boundary conditions
*-----------------------------------------------------------------------

      Yapros(1)=Yabcpro
      Tapros(1)=Tabcpro
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for first, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known3(1)=Wprev(1,1) + B3(1)*DBLE(dt)*(Yaprov(1)-Ywprev(1,1))

      known4(1)=Twprev(1,1) + DBLE(dt)*B4(1)*(Yaprov(1)-Ywprev(1,1)) +
     $            B5(1)*DBLE(dt)*(Taprov(1) - Twprev(1,1))
*-----------------------------------------------------------------------
*    Solve for desiccant conditions on the supply-side surface for the
* first, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(1,1)=(known3(1) + B3(1)*DBLE(dt)*Yapros(1))/(1.d0 + B3(1)*
     $         DBLE(dt)*Ywmax(1,1)/Wmax)

      Ywpres(1,1)=Wpres(1,1)*Ywmax(1,1)/Wmax
```

```
      Twpres(1,1)=(known4(1) + B4(1)*DBLE(dt)*(Yapros(1)-Ywpres(1,1)) +
     $             B5(1)*DBLE(dt)*Tapros(1))/(1.d0 + B5(1)*DBLE(dt))
*-----------------------------------------------------------------------
*   Solve for the desiccant conditions through the thickness of the plate
* at the position of the first, plate edge node (an exterior node).
*-----------------------------------------------------------------------

      do ii=2,ny-1
         Wpres(1,ii)=Wprev(1,ii)+B9(1,ii)*(Wprev(1,ii+1)+
     $               Wprev(1,ii-1) -2.d0*Wprev(1,ii))

         Twpres(1,ii)=Twprev(1,ii)*(1.d0 -2.d0*B10(1,ii)) +
     $                (B10(1,ii)-B11(1,ii))*Twprev(1,ii+1) +
     $                (B10(1,ii)+B11(1,ii))*Twprev(1,ii-1)

         Ywpres(1,ii)=(Ywmax(1,ii)/Wmax *Wpres(1,ii))
       end do
*-----------------------------------------------------------------------
*    Solve for desiccant conditions on the exhaust-side surface for the
* first, plate edge node (an exterior node)
*-----------------------------------------------------------------------

         Twpres(1,ny)=((hreg*dx)/(alpha*rhod*Cd)*
     $                Taregs(1) + Twpres(1,ny-1))/((hreg*dx
     $                )/(alpha*rhod*Cd) + 1.d0)

         Wpres(1,ny)=(Wpres(1,ny-1)+(Ky(1)*Wmax*dx/(rhowater*Dab))*
     $                 Yaregs(1)/Yaregmax(1))/(1.d0 + Ky(1)*dx/
     $                 (rhowater*Dab))

         Ywpres(1,ny)=(Ywmax(1,ny)/Wmax *Wpres(1,ny))
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for all interior
*  nodes.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      do i=2,nx-1
         known1(i)=(B6(i)*dx/2.d0)*Ywpres(i-1,1) +
     $    (1.d0-B6(i)*dx/ 2.d0)*Yapros(i-1)

         known2(i)=B7(i)*dx/2.d0*(Ywpres(i-1,1) - Yapros(i-1)) +
     $             B8(i)*dx/2.d0*(Twpres(i-1,1) - Tapros(i-1)) +
     $             Tapros(i-1)

 known3(i)=Wprev(i,1) + B3(i)*DBLE(dt)*(Yaprov(i)-Ywprev(i,1))

         known4(i)=Twprev(i,1) + DBLE(dt)*B4(i)*(Yaprov(i)-
     $             Ywprev(i,1)) + B5(i)*DBLE(dt)*(Taprov(i) -
     $             Twprev(i,1))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the supply-side
* surface for all interior nodes
*-----------------------------------------------------------------------

         Wpres(i,1)=(known3(i) + B3(i)*DBLE(dt)*known1(i)/(1.d0+B6(i)*
     $           dx/2.d0))/(1.d0 - (B3(i)*B6(i)*Ywmax(i,1)*DBLE(dt)*dx/
     $           (2.d0*Wmax))/(1.d0 + B6(i)*dx/2.d0) + B3(i)*DBLE(dt)*
     $           Ywmax(i,1)/Wmax)

         Yapros(i)=(known1(i) + B6(i)*dx/2.d0 * (Ywmax(i,1)/Wmax *
     $             Wpres(i,1)))/(1.d0 + B6(i)*dx/2.d0)

         Ywpres(i,1)=(Ywmax(i,1)/Wmax *Wpres(i,1))

         Twpres(i,1)=(known4(i) + B4(i)*DBLE(dt)*(Yapros(i)-
     $             Ywpres(i,1)) + B5(i)*DBLE(dt)/(1.d0 + B8(i)*dx/2.d0)
     $             * (known2(i) + B7(i)*dx/2.d0 * (Ywpres(i,1)-
```

```
     $            Yapros(i))))/(1.d0 + B5(i)*DBLE(dt) - B5(i)*B8(i)*
     $            DBLE(dt)*dx/(2.d0*(1.d0 + B8(i)*dx/2.d0)))

         Tapros(i)=(known2(i) + B7(i)*dx/2.d0*(Ywpres(i,1) - Yapros(i))
     $            + B8(i)*dx/2.d0*Twpres(i,1))/(1.d0+B8(i)*dx/2.d0)
*-----------------------------------------------------------------------
*   Solve for the desiccant conditions through the thickness of the plate
* at the position of all interior nodes.
*-----------------------------------------------------------------------
         do ii=2,ny-1

            Wpres(i,ii)=Wprev(i,ii)+B9(i,ii)*(Wprev(i,ii+1)+
     $              Wprev(i,ii-1) - 2.d0*Wprev(i,ii))

            Twpres(i,ii)=Twprev(i,ii)*(1.d0 -2.d0*B10(i,ii)) +
     $               (B10(i,ii)-B11(i,ii))*Twprev(i,ii+1) +
     $               (B10(i,ii)+B11(i,ii))*Twprev(i,ii-1)
         end do
*-----------------------------------------------------------------------
*    Solve for desiccant conditions on the exhaust-side surface for all
* interior nodes
*-----------------------------------------------------------------------

         Twpres(i,ny)=((hreg*dx)/(alpha*rhod*Cd)*
     $               Taregs(i) + Twpres(i,ny-1))/((hreg*dx
     $               )/(alpha*rhod*Cd) + 1.d0)

          Wpres(i,ny)=(Wpres(i,ny-1)+(Ky(i)*Wmax*dx/(rhowater*Dab))*
     $                Yaregs(i)/Yaregmax(i))/(1.d0 + Ky(i)*dx/
     $                (rhowater*Dab))

         Ywpres(i,ny)=(Ywmax(i,ny)/Wmax *Wpres(i,ny))
      end do
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for last, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known1(nx)=(B6(nx)*dx/4.d0)*Ywpres(nx-1,1) +
     $    (1.d0-B6(nx)*dx/4.d0)*Yapros(nx-1)

      known2(nx)=B7(nx)*dx/4.d0*(Ywpres(nx-1,1) - Yapros(nx-1)) +
     $            B8(nx)*dx/4.d0*(Twpres(nx-1,1) - Tapros(nx-1)) +
     $            Tapros(nx-1)

      known3(nx)=Wprev(nx,1) + B3(nx)*DBLE(dt)*(Yaprov(nx)-Ywprev(nx,1))

      known4(nx)=Twprev(nx,1) + DBLE(dt)*B4(nx)*(Yaprov(nx)-
     $          Ywprev(nx,1)) + B5(nx)*DBLE(dt)*(Taprov(nx) -
     $          Twprev(nx,1))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream  conditions on the supply-side
* surface for the last, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(nx,1)=(known3(nx) + B3(nx)*DBLE(dt)*known1(nx)/(1.d0+B6(nx)*
     $          dx/2.d0))/(1.d0 - (B3(nx)*B6(nx)*Ywmax(nx,1)*DBLE(dt)*
     $          dx/(2.d0*Wmax))/(1.d0 + B6(nx)*dx/2.d0) + B3(nx)*
     $          DBLE(dt)*Ywmax(nx,1)/Wmax)

      Yapros(nx)=(known1(nx) + B6(nx)*dx/4.d0 * (Ywmax(nx,1)/Wmax *
     $            Wpres(nx,1)))/(1.d0 + B6(nx)*dx/4.d0)

      Ywpres(nx,1)=(Ywmax(nx,1)/Wmax *Wpres(nx,1))

      Twpres(nx,1)=(known4(nx) + B4(nx)*DBLE(dt)*(Yapros(nx)-
     $            Ywpres(nx,1)) + B5(nx)*DBLE(dt)/(1.d0 + B8(nx)*dx/
```

```
     $             2.d0) * (known2(nx) + B7(nx)*dx/2.d0 * (Ywpres(nx,1)-
     $             Yapros(nx))))/(1.d0 + B5(nx)*DBLE(dt) - B5(nx)*
     $             B8(nx)*DBLE(dt)*dx/(2.d0*(1.d0 +
     $             B8(nx)*dx/2.d0)))

      Tapros(nx)=(known2(nx) + B7(nx)*dx/4.d0*(Ywpres(nx,1) -
     $              Yapros(nx)) + B8(nx)*dx/4.d0*Twpres(nx,1))/
     $              (1.d0+B8(nx)*dx/4.d0)
*------------------------------------------------------------------------
*   Solve for the desiccant conditions through the thickness of the plate
* at the position of the last, plate edge node (an exterior node).
*------------------------------------------------------------------------
      do ii=2,ny-1
          Wpres(nx,ii)=Wprev(nx,ii)+B9(nx,ii)*(Wprev(nx,ii+1)+
     $               Wprev(nx,ii-1) - 2.d0*Wprev(nx,ii))

            Twpres(nx,ii)=Twprev(nx,ii)*(1.d0 -2.d0*B10(nx,ii)) +
     $                (B10(nx,ii)-B11(nx,ii))*Twprev(nx,ii+1) +
     $                (B10(nx,ii)+B11(nx,ii))*Twprev(nx,ii-1)

         Ywpres(nx,ii)=(Ywmax(nx,ii)/Wmax *Wpres(nx,ii))
      end do
*------------------------------------------------------------------------
*    Solve for desiccant conditions on the exhaust-side surface for the
* last, plate edge node (an exterior node)
*------------------------------------------------------------------------

      Twpres(nx,ny)=((hreg*dx)/(alpha*rhod*Cd)*
     $                Taregs(nx) + Twpres(nx,ny-1))/((hreg*dx
     $                )/(alpha*rhod*Cd) + 1.d0)

       Wpres(nx,ny)=(Wpres(nx,ny-1)+(Ky(nx)*Wmax*dx/(rhowater*Dab))*
     $                 Yaregs(nx)/Yaregmax(nx))/(1.d0 + Ky(nx)*dx/
     $                 (rhowater*Dab))

       Ywpres(nx,ny)=(Ywmax(nx,ny)/Wmax *Wpres(nx,ny))
*------------------------------------------------------------------------
*   Return to the main driver
*------------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE COEFREGEN
**********************************************************************
*------------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to COEFREGEN and various other subroutines easier.
*------------------------------------------------------------------------
      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/properties/ Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*------------------------------------------------------------------------
*   Declare all variables that are used in the subroutine COEFREGEN.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*------------------------------------------------------------------------

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      double precision Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
```

```
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the COEFREGEN and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
*-----------------------------------------------------------------------
*   Assign air conditions solved for in the previous time step as 'present'
* to 'previous' status for finite difference solution purposes.
*-----------------------------------------------------------------------
*   Step along the x direction, along the plate depth
*-----------------------------------------------------------------------
*   Note:  bed conditions are not updated here according to solution method.
* bad conditions are not updated until the end/beginning of each time step
* due to diffusion occurring within the bed.
*-----------------------------------------------------------------------

      do i=nx,1,-1
         Yaregv(i)=Yaregs(i)
         Taregv(i)=Taregs(i)
         TempKa(i)=2.7315d+2 + Taregv(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 -
     $       TempKa(i))))
         PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
         Yamaxprev(i)=6.22d-1*PwsTa(i)/Patm
      end do
*-----------------------------------------------------------------------
*   Make sure that boundary conditions for the exhaust air stream  continue
* to be used.
*-----------------------------------------------------------------------

      Taregs(nx)=Tabcregen
      Yaregs(nx)=Yabcregen
*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------

      return
      end

************************************************************************
      SUBROUTINE REGENERATION
************************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to REGENERATION and various other subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/geometry/ width,depth,height,thickd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,pressuredrop,perimeter
```

```
      common/specificheat/ Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      common/time/ timetot,dt
      common/distance/ dx,dy
      common/properties/ Q,hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      common/density/ rhod,rhos,rhowater,rhopro,rhoreg,rhoevery
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/coefficients/B3,B4,B5,B6,B7,B8,B9,B10,B11
      common/knowns/ known1,known2,known3,known4
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine REGENERATION.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      real f,timetot,dt
      double precision Ky,velmass
      double precision width,depth,height,thickd,perw,perd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      double precision dx,dy
      double precision rhod,rhos,rhowater,rhopro,rhoreg,rhoevery,Q,Hv,
     $   hpro,hreg,velpro,velreg,Wmax,alpha,Dab,Patm
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision B3,B4,B5,B6,B7,B8,B9,B10,B11
      double precision known1,known2,known3,known4
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the REGENERATION and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension Ky(NMAX1),Hv(NMAX1),velmass(NMAX1,NMAX3)
      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension B3(NMAX1),B4(NMAX1),B5(NMAX1),B6(NMAX1),B7(NMAX1),
     $   B8(NMAX1),B9(NMAX1,NMAX3),B10(NMAX1,NMAX3),B11(NMAX1,NMAX3)
      dimension known1(NMAX1),known2(NMAX1),known3(NMAX1),
     $    known4(NMAX1)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
*-----------------------------------------------------------------------
*    Stepping along in x, the plate depth, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every iteration/time step.
*-----------------------------------------------------------------------

       do i=nx,1,-1
         TempKa(i)=2.7315d+2 + Taregv(i)
         ua(i)=(6.4727d+2 - TempKa(i))  *(3.2437814d0 +
     $        (5.86826d-03 + 1.1702379d-08*(6.4727d+2 - TempKa(i))*
```

```
     $        (6.4727d+2 - TempKa(i)))* (6.4727d+2 - TempKa(i)))/
     $        (TempKa(i) *(1.d0+2.1878462d-03 *(6.4727d+2 - TempKa(i))))
         PwsTa(i)=2.18167d+2/(10.d0 **ua(i))
         Yaregmax(i)=6.22d-1*PwsTa(i)/Patm
*-----------------------------------------------------------------------
*    Stepping along in y, the plate thickness, calculate the maximum air moisture
* content.  This is a property that is temperature dependent and therefore
* must be re-calculated after every iteration/time step.
*-----------------------------------------------------------------------

         do ii=ny,1,-1
            TempKw(i,ii)=2.7315d+2 + Twprev(i,ii)
            uw(i,ii)=(6.4727d+2 - TempKw(i,ii))  *(3.2437814d0 +
     $             (5.86826d-03 + 1.1702379d-08*(6.4727d+2 -
     $             TempKw(i,ii))*(6.4727d+2 - TempKw(i,ii)))*
     $             (6.4727d+2 - TempKw(i,ii)))/(TempKw(i,ii) *
     $             (1.d0+2.1878462d-03 *(6.4727d+2 - TempKw(i,ii))))
            PwsTw(i,ii)=2.18167d+2/(10.d0 **uw(i,ii))
            Ywmax(i,ii)=6.22d-1*PwsTw(i,ii)/Patm
          end do
*-----------------------------------------------------------------------
*    Calculate the mass transfer coefficient, a moisture content dependent
* material property.
*-----------------------------------------------------------------------

         Ky(i)=10.d0*(6855.d-1*(Wprev(i,ny)*Massdc)**2 - 362.d-1*
     $        (Wprev(i,ny)*Massdc)+.0873d0)
*-----------------------------------------------------------------------
*    Calculate the heat of vaporization, a temperature dependent
* material property.
*-----------------------------------------------------------------------

         Hv(i)=2502.d0-Cv/1.d+3*Taregv(i)
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------

         B3(i)=Ky(i)/(thickd*rhod*DBLE(f))
         B4(i)=Hv(i)*Ky(i)/(thickd*rhoevery*(DBLE(1.-f)*Cs+DBLE(f)*
     $        (Cd*Twprev(i,ny)+Wprev(i,ny)*(Hv(i)-Q))))
         B5(i)=hreg/(thickd*rhoevery*(DBLE(1.-f)*Cs+DBLE(f)*
     $        (Cd*Twprev(i,ny)+Wprev(i,ny)*(Hv(i)-Q))))
         B6(i)=perimeter*Ky(i)/(rhoreg*velreg*Areaair)
         B7(i)=perimeter*Ky(i)*Hv(i)/(rhoreg*velreg*Areaair*
     $        (Cdareg+Yaregv(i)*Cv))
         B8(i)=hreg*perimeter/(rhoreg*velreg*Areaair*(Cdareg+
     $        Yaregv(i)*Cv))
*-----------------------------------------------------------------------
*   Determine the speed at which the moisture travels through the
* thickness of the desiccant plate on a node-by-node basis.  If this is
* the initial time step, a very small value is assigned since the initial
* bed conditions indicate no mass gradient through the plate thickness.
*-----------------------------------------------------------------------

         do ii=ny,1,-1
             velmass(i,ii)=Massdc*Dab*(Wpres(i,ii)-Wprev(i,ii-1))/(dy*
     $                 Voltot*rhowater)

             if (j .eq. 1) then
                velmass(i,ii)=-1.d-12
             end if
*-----------------------------------------------------------------------
*    Determine coefficients used within the finite-differencing scheme
* These coefficients are used later in this subroutine to solve the
* nodal equations shown further in this subroutine.
*-----------------------------------------------------------------------
```

```
          B9(i,ii)=DBLE(dt)*Dab/(dy**2)
          B10(i,ii)=alpha*DBLE(dt)/(dy**2)
          B11(i,ii)=velmass(i,ii)*DBLE(dt)/(2.d0*dy)
        end do
      end do

*-----------------------------------------------------------------------
*    This closes the coefficient determination phase of the subroutine.  We
* now move on to the nodal solution of the supply-side phase.
*-----------------------------------------------------------------------
************************************************************************
*-----------------------------------------------------------------------
*  Maintain regenerative-side air boundary conditions
*-----------------------------------------------------------------------

      Yaregs(nx)=Yabcregen
      Taregs(nx)=Tabcregen
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for last, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      known3(nx)=Wprev(nx,ny) + B3(nx)*DBLE(dt)*(Yaregv(nx)-
     $           Ywprev(nx,ny))

      known4(nx)=Twprev(nx,ny) + DBLE(dt)*B4(nx)*(Yaregv(nx)-
     $           Ywprev(nx,ny)) + B5(nx)*DBLE(dt)*(Taregv(nx) -
     $           Twprev(nx,ny))
*-----------------------------------------------------------------------
*    Solve for desiccant conditions on the exhaust-side surface for the
* last, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(nx,ny)=(known3(nx) + B3(nx)*DBLE(dt)*Yaregs(nx))/(1.d0 +
     $          B3(nx)*DBLE(dt)*Ywmax(nx,ny)/Wmax)

      Ywpres(nx,ny)=Wpres(nx,ny)*Ywmax(nx,ny)/Wmax

      Twpres(nx,ny)=(known4(nx) + B4(nx)*DBLE(dt)*(Yaregs(nx)-
     $              Ywpres(nx,ny)) + B5(nx)*DBLE(dt)*Taregs(nx))/
     $              (1.d0 + B5(nx)*DBLE(dt))
*-----------------------------------------------------------------------
*   Solve for the desiccant conditions through the thickness of the plate
* at the position of the last, plate edge node (an exterior node).
*-----------------------------------------------------------------------

      do ii=ny-1,2,-1
         Wpres(nx,ii)=Wprev(nx,ii)+B9(nx,ii)*(Wprev(nx,ii+1)+
     $               Wprev(nx,ii-1) -2.d0* Wprev(nx,ii))

         Twpres(nx,ii)=Twprev(nx,ii)*(1.d0 -2.d0*B10(nx,ii)) +
     $                (B10(nx,ii)-B11(nx,ii))*Twprev(nx,ii+1) +
     $                (B10(nx,ii)+B11(nx,ii))*Twprev(nx,ii-1)

         Ywpres(nx,ii)=(Ywmax(nx,ii)/Wmax *Wpres(nx,ii))
      end do
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream  conditions on the supply-side
* surface for the last, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Twpres(nx,1)=((hreg*dx)/(alpha*rhod*Cd)*
     $                Tapros(nx) + Twpres(nx,2))/((hreg*dx
     $                )/(alpha*rhod*Cd) + 1.d0)

      Wpres(nx,1)=(Wpres(nx,2) + (Ky(nx)*Wmax*dx/(rhowater*Dab))*
```

```
     $                 Yapros(nx)/Yapromax(nx))/(1.d0 + Ky(nx)*dx/
     $                 (rhowater*Dab))

      Ywpres(nx,1)=(Ywmax(nx,1)/Wmax *Wpres(nx,1))
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for all interior
*  nodes.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------

      do i=nx-1,2,-1
         known1(i)=(B6(i)*dx/2.d0)*Ywpres(i+1,ny) +
     $    (1.d0-B6(i)*dx/ 2.d0)*Yaregs(i+1)

         known2(i)=B7(i)*dx/2.d0*(Ywpres(i+1,ny) - Yaregs(i+1)) +
     $            B8(i)*dx/2.d0*(Twpres(i+1,ny) - Taregs(i+1)) +
     $            Taregs(i+1)

 known3(i)=Wprev(i,ny) + B3(i)*DBLE(dt)*(Yaregv(i)-Ywprev(i,ny))

         known4(i)=Twprev(i,ny) + DBLE(dt)*B4(i)*(Yaregv(i)-
     $            Ywprev(i,ny)) + B5(i)*DBLE(dt)*(Taregv(i) -
     $            Twprev(i,ny))
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the exhaust-side
* surface for all interior nodes
*-----------------------------------------------------------------------

         Wpres(i,ny)=(known3(i) + B3(i)*DBLE(dt)*known1(i)/(1.d0+
     $           B6(i)*dx/2.d0))/(1.d0 - (B3(i)*B6(i)*Ywmax(i,ny)*
     $           DBLE(dt)*dx/(2.d0*Wmax))/(1.d0 + B6(i)*dx/2.d0) +
     $           B3(i)*DBLE(dt)*Ywmax(i,ny)/Wmax)

         Yaregs(i)=(known1(i) + B6(i)*dx/2.d0 * (Ywmax(i,ny)/Wmax *
     $            Wpres(i,ny)))/(1.d0 + B6(i)*dx/2.d0)

         Ywpres(i,ny)=(Ywmax(i,ny)/Wmax *Wpres(i,ny))

         Twpres(i,ny)=(known4(i) + B4(i)*DBLE(dt)*(Yaregs(i)-
     $            Ywpres(i,ny)) + B5(i)*DBLE(dt)/(1.d0 + B8(i)*dx/2.d0)
     $            * (known2(i) + B7(i)*dx/2.d0 * (Ywpres(i,ny)-
     $            Yaregs(i))))/(1.d0 + B5(i)*DBLE(dt) - B5(i)*B8(i)*
     $            DBLE(dt)*dx/(2.d0*(1.d0 + B8(i)*dx/2.d0)))

         Taregs(i)=(known2(i) + B7(i)*dx/2.d0*(Ywpres(i,ny) - Yaregs(i))
     $            + B8(i)*dx/2.d0*Twpres(i,ny))/(1.d0+B8(i)*dx/2.d0)
*-----------------------------------------------------------------------
*    Solve for desiccant conditions through the plate thickness
* for all interior nodes
*-----------------------------------------------------------------------

         do ii=ny-1,2,-1
             Wpres(i,ii)=Wprev(i,ii)+B9(i,ii)*(Wprev(i,ii+1)+
     $             Wprev(i,ii-1) -2.d0* Wprev(i,ii))

             Twpres(i,ii)=Twprev(i,ii)*(1.d0 -2.d0*B10(i,ii)) +
     $              (B10(i,ii)-B11(i,ii))*Twprev(i,ii+1) +
     $              (B10(i,ii)+B11(i,ii))*Twprev(i,ii-1)

             Ywpres(i,ii)=(Ywmax(i,ii)/Wmax *Wpres(i,ii))
         end do
*-----------------------------------------------------------------------
*    Solve for desiccant and air stream conditions on the supply-side
* surface for all interior nodes
*-----------------------------------------------------------------------

         Twpres(i,1)=((hreg*dx)/(alpha*rhod*Cd)*
     $                Tapros(i) + Twpres(i,2))/((hreg*dx
```

```
     $               )/(alpha*rhod*Cd) + 1.d0)

          Wpres(i,1)=(Wpres(i,2) + (Ky(i)*Wmax*dx/(rhowater*Dab))*
     $                Yapros(i)/Yapromax(i))/(1.d0 + Ky(i)*dx/
     $                (rhowater*Dab))

         Ywpres(i,1)=(Ywmax(i,1)/Wmax *Wpres(i,1))
      end do
*-----------------------------------------------------------------------
*   Define explicitly known portion of nodal equations for first, plate
* edge node.  These expressions contain information gathered from previous
* time steps.
*-----------------------------------------------------------------------
      known1(1)=(B6(1)*dx/4.d0)*Ywpres(2,ny) +
     $    (1.d0-B6(1)*dx/4.d0)*Yaregs(2)

      known2(1)=B7(1)*dx/4.d0*(Ywpres(2,ny) - Yaregs(2)) +
     $             B8(1)*dx/4.d0*(Twpres(2,ny) - Taregs(2)) +
     $             Taregs(2)

      known3(1)=Wprev(1,ny) + B3(1)*DBLE(dt)*(Yaregv(1)-Ywprev(1,ny))

      known4(1)=Twprev(1,ny) + DBLE(dt)*B4(1)*(Yaregv(1)-Ywprev(1,ny)) +
     $             B5(1)*DBLE(dt)*(Taregv(1) - Twprev(1,ny))
*-----------------------------------------------------------------------
*    Solve for desiccant and air conditions on the exhaust-side surface for
* the first, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      Wpres(1,ny)=(known3(1) + B3(1)*DBLE(dt)*known1(1)/(1.d0+B6(1)*
     $           dx/2.d0))/(1.d0 - (B3(1)*B6(1)*Ywmax(1,ny)*DBLE(dt)*dx/
     $           (2.d0*Wmax))/(1.d0 + B6(1)*dx/2.d0) + B3(1)*DBLE(dt)*
     $           Ywmax(1,ny)/Wmax)

      Yaregs(1)=(known1(1) + B6(1)*dx/4.d0 * (Ywmax(1,ny)/Wmax *
     $             Wpres(1,ny)))/(1.d0 + B6(1)*dx/4.d0)

      Ywpres(1,ny)=(Ywmax(1,ny)/Wmax *Wpres(1,ny))

      Twpres(1,ny)=(known4(1) + B4(1)*DBLE(dt)*(Yaregs(1)-
     $             Ywpres(1,ny)) + B5(1)*DBLE(dt)/(1.d0 + B8(1)*dx/
     $             2.d0) * (known2(1) + B7(1)*dx/2.d0 * (Ywpres(1,ny)-
     $             Yaregs(1))))/(1.d0 + B5(1)*DBLE(dt) - B5(1)*B8(1)*
     $             DBLE(dt)*dx/(2.d0*(1.d0 + B8(1)*dx/2.d0)))

      Taregs(1)=(known2(1) + B7(1)*dx/4.d0*(Ywpres(1,ny) -
     $              Yaregs(1)) + B8(1)*dx/4.d0*Twpres(1,ny))/
     $              (1.d0+B8(1)*dx/4.d0)
*-----------------------------------------------------------------------
*    Solve for desiccant conditions through the plate thickness for the
* first, plate edge node (an exterior node)
*-----------------------------------------------------------------------

      do ii=ny-1,2,-1
          Wpres(1,ii)=Wprev(1,ii)+B9(1,ii)*(Wprev(1,ii+1)+
     $               Wprev(1,ii-1) -2.d0* Wprev(1,ii))

         Twpres(1,ii)=Twprev(1,ii)*(1.d0 -2.d0*B10(1,ii)) +
     $               (B10(1,ii)-B11(1,ii))*Twprev(1,ii+1) +
     $               (B10(1,ii)+B11(1,ii))*Twprev(1,ii-1)

         Ywpres(1,ii)=(Ywmax(1,ii)/Wmax *Wpres(1,ii))
      end do
*-----------------------------------------------------------------------
*    Solve for desiccant and air conditions on the supply-side surface for
* the first, plate edge node (an exterior node)
*-----------------------------------------------------------------------
```

```
      Twpres(1,1)=((hreg*dx)/(alpha*rhod*Cd)*
     $                Tapros(1) + Twpres(1,2))/((hreg*dx
     $                )/(alpha*rhod*Cd) + 1.d0)

       Wpres(1,1)=(Wpres(1,2) + (Ky(1)*Wmax*dx/(rhowater*Dab))*
     $                 Yapros(1)/Yapromax(1))/(1.d0 + Ky(1)*dx/
     $                 (rhowater*Dab))

      Ywpres(1,1)=(Ywmax(1,1)/Wmax *Wpres(1,1))
*_________________________________________________________________________
*   Return to the main driver
*_________________________________________________________________________

      return
      end


**********************************************************************
      SUBROUTINE CONVERGENCE
**********************************************************************
*_________________________________________________________________________
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to CONVERGENCE and various other subroutines easier.
*_________________________________________________________________________

      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/criteria/proconverge,regconverge
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*_________________________________________________________________________
*   Declare all variables that are used in the subroutine CONVERGENCE.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*_________________________________________________________________________

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      integer check1,check2,proconverge,regconverge
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*_________________________________________________________________________
*     Initialize parameters used in the dimensioning of all arrays used in
* the CONVERGENCE and transferred between subroutines.
*_________________________________________________________________________

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
*_________________________________________________________________________
*    Initialize variables used in helping determine whether there is a
* convergence problem with the PROCESS or REGENERATIVE solution schemes.
* Initialize counter l as well.
*_________________________________________________________________________

      check1=0
      check2=0
      l=l+1
*_________________________________________________________________________
*    If program has just finished PROCESS subroutine, check for convergence
*_________________________________________________________________________

      if (z .eq. 1) then
         go to 800
      end if
```

```
*-----------------------------------------------------------------------
*    If program has just finished PROCESS subroutine, check for convergence
*-----------------------------------------------------------------------

      if (z .eq. 2) then
         go to 810
      end if
*-----------------------------------------------------------------------
*    Check to see if convergence has been met with regards to temperature
* in the exiting supply air stream.
*-----------------------------------------------------------------------

 800  if (ABS((Taprov(nx)-Tapros(nx))/Taprov(nx)) .le. 1.d-2) then
          check1=1
      end if
*-----------------------------------------------------------------------
*    Check to see if convergence has been met with regards to humidity ratio
* in the exiting supply air stream.
*-----------------------------------------------------------------------

      if (ABS((Yaprov(nx)-Yapros(nx))/Yaprov(nx)) .le. 1.d-2) then
         check2=1
      end if
*-----------------------------------------------------------------------
*    Determine whether convergence has been met for supply air with regards
* to both humidity ratio and temperature and flag accordingly.  This flag will
* be picked up in the MAIN DRIVER and handled by either restarting the program
* if convergence has not been met, or continuing on with the solution if
* convergence has been achieved.
*-----------------------------------------------------------------------

      if (((check1 .eq. 1) .and. (check2 .eq. 1)) .or. j .eq. 1) then
         proconverge=1
      end if
      go to 820
*-----------------------------------------------------------------------
*    Check to see if convergence has been met with regards to temperature
* in the exiting exhaust air stream.
*-----------------------------------------------------------------------

 810  if (ABS((Taregv(1)-Taregs(1))/Taregv(1)) .le. 1.d-2) then
          check1=1
      end if
*-----------------------------------------------------------------------
*    Check to see if convergence has been met with regards to humidity ratio
* in the exiting exhaust air stream.
*-----------------------------------------------------------------------

      if (ABS((Yaregv(1)-Yaregs(1))/Yaregv(1)) .le. 1.d-2) then
         check2=1
      end if
*-----------------------------------------------------------------------
*    Determine whether convergence has been met for exhaust air with regards
* to both humidity ratio and temperature and flag accordingly.  This flag will
* be picked up in the MAIN DRIVER and handled by either restarting the program
* if convergence has not been met, or continuing on with the solution if
* convergence has been achieved.
*-----------------------------------------------------------------------

      if (((check1 .eq. 1) .and. (check2 .eq. 1)) .or. j .eq. 1) then
         regconverge=1
      end if
*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------
 820  return
      end
```

```
**********************************************************************
      SUBROUTINE XFER
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to XFER and various other subroutines easier.
*-----------------------------------------------------------------------
      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/airprops/ TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      common/nodal_solution/ Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,
     $   Tapros,Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine XFER.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX1,NMAX3
      double precision TempKa,TempKw,Yapromax,Yaregmax,Ywmax,Yamaxprev,
     $   Ywmaxprev,ua,uw,PwsTa,PwsTw
      double precision Twpres,Wpres,Ywpres,Twprev,Wprev,Ywprev,Tapros,
     $   Yapros,Taprov,Yaprov,Taregs,Yaregs,Taregv,Yaregv
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the XFER and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX1=20000)
      parameter (NMAX3=25)

      dimension TempKa(NMAX1),TempKw(NMAX1,NMAX3),Yapromax(NMAX1),
     $   Yaregmax(NMAX1),Ywmax(NMAX1,NMAX3),Yamaxprev(NMAX1),
     $   Ywmaxprev(NMAX1,NMAX3),ua(NMAX1),uw(NMAX1,NMAX3),PwsTa(NMAX1),
     $   PwsTw(NMAX1,NMAX3)
      dimension Twpres(NMAX1,NMAX3),Wpres(NMAX1,NMAX3),
     $   Ywpres(NMAX1,NMAX3),Twprev(NMAX1,NMAX3),Wprev(NMAX1,NMAX3),
     $   Ywprev(NMAX1,NMAX3),Yapros(NMAX1),Tapros(NMAX1),Yaprov(NMAX1),
     $   Taprov(NMAX1),Taregs(NMAX1),Yaregs(NMAX1),Taregv(NMAX1),
     $   Yaregv(NMAX1)
*-----------------------------------------------------------------------
*    After subroutines are completed, air conditions labeled 'present'
* are transferred to 'previous' for nodal solution purposes.
*-----------------------------------------------------------------------
*    If finished with PROCESS, redefine...
*-----------------------------------------------------------------------

         do i=1,nx
            if (z .eq. 1) then
               Yaprov(i)=Yapros(i)
               Taprov(i)=Tapros(i)
               Yamaxprev(i)=Yapromax(i)
             end if
*-----------------------------------------------------------------------
*    IF finished with REGENERATIVE, redefine...
*-----------------------------------------------------------------------

             if (z .eq. 2) then
               Yaregv(i)=Yaregs(i)
               Taregv(i)=Taregs(i)
               Yamaxprev(i)=Yaregmax(i)
             end if
         end do
*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------
      return
      end
```

```
*********************************************************************
      SUBROUTINE EFFICIENCY
*********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to EFFICIENCY and various other subroutines easier.
*-----------------------------------------------------------------------
      common/specificheat/ Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      common/heat/ hgpro,hgregen,hgtot
      common/conditions/ Twic,Wic,Yabcpro,Tabcpro,Yabcregen,
     $  Tabcregen
      common/solution/ Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine EFFICIENCY.
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer NMAX2
      double precision eff1,eff2,eff3
      double precision Cwr,Cdapro,Cdareg,Cv,Cd,Cs
      double precision hgpro,hgregen,hgtot,haprotot,
     $  haproregen
      double precision Twic,Wic,Yabcpro,Tabcpro,Yabcregen,Tabcregen
      double precision Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot
      double precision Ppro,Ptot,Pregen,Xpro,Xtot,Xregen,Ypro,Ytot,
     $   Yregen
*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the EFFICIENCY and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX2=90000)

      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Ywtotpro(NMAX2),
     $   Yatotpro(NMAX2),Tatotpro(NMAX2),Twtotreg(NMAX2),
     $   Wtotreg(NMAX2),Ywtotreg(NMAX2),Yatotreg(NMAX2),Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    Determine coefficients used in solving for enthalpy
*-----------------------------------------------------------------------

 Xpro= 374.12d0-Tabcpro
 Xtot= 374.12d0-Ta
 Xregen= 374.12d0-Tabcregen
 Ypro= Xpro*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $        Xpro**2)*Xpro)/((Tabcpro+273.15d0)*(1.d0+
     $        2.1878462d-3*Xpro))
 Ytot= Xtot*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $   Xtot**2)*Xtot)/((Ta+273.15d0)*(1.0d0+2.1878462d-3*
     $   Xtot))
 Yregen= Xregen*(3.2437814d0+(5.86826d-3+1.1702379d-8*
     $  Xregen**2)*Xregen)/((Tabcregen+273.15d0)*(1.d0+
     $  2.1878462d-3*Xregen))

 Ppro = 14.696d0*218.167d0/(10.0d0**Ypro)
 Ptot = 14.696d0*218.167d0/(10.0d0**Ytot)
 Pregen =  14.696d0*218.167d0/(10.0d0**Yregen)
 Xpro = DLOG10(Ppro)
 Xtot = DLOG10(Ptot)
 Xregen = DLOG10(Pregen)
*-----------------------------------------------------------------------
*   Calculate enthaply associated with the air streams
*-----------------------------------------------------------------------

 hgpro = (1105.9387d0+( 32.756807d0+(4.6198474d0+
```

```
     $           (0.20672996d0+(-0.5411693d0+(0.49241362d0-
     $           0.17884885d0*Xpro)*Xpro)*Xpro)*Xpro)*Xpro)*
     $           Xpro)/.4299d0
 hgtot = (1105.9387d0+( 32.756807d0+(4.6198474d0+
     $           (0.20672996d0+(-0.5411693d0+(0.49241362d0-
     $           0.17884885d0*Xtot)*Xtot)*Xtot)*Xtot)*Xtot)*
     $           Xtot)/.4299d0
 hgregen =(1105.9387d0+( 32.756807d0+(4.6198474d0+
     $           (0.20672996d0+(-0.5411693d0+(0.49241362d0-
     $           0.17884885d0*Xregen)*Xregen)*Xregen)*Xregen)*Xregen)*
     $           Xregen)/.4299d0

610   haprotot=Cdapro*(Tabcpro-Ta) + (Yabcpro*hgpro-Ya*hgtot)
      haproregen=Cdareg*(Tabcpro-Tabcregen) + (Yabcpro*hgpro-Yabcregen*
     $  hgregen)
*-----------------------------------------------------------------------
*    Calculate sensible efficiency
*-----------------------------------------------------------------------

      eff1= (Tabcpro-Ta)/(Tabcpro-Tabcregen)
*-----------------------------------------------------------------------
*    Calculate latent efficiency
*-----------------------------------------------------------------------

      eff2=(Yabcpro*hgpro-Ya*hgtot)/
     $  (Yabcpro*hgpro-Yabcregen*hgregen)
*-----------------------------------------------------------------------
*    Calculate total efficiency
*-----------------------------------------------------------------------

      eff3= haprotot/haproregen
*-----------------------------------------------------------------------
*    Open the output data file and record efficiency values
*-----------------------------------------------------------------------

      open(44,file='plate.dat')
      write(44,*)
      write(44,*) 'Efficiencies are:'
      write(44,*)
      write(44,*) 'Sensible =',eff1
      write(44,*) 'Latent =',eff2
      write(44,*) 'Total =',eff3
      write(44,*)
*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------

      return
      end

**********************************************************************
      SUBROUTINE OUTPUT
**********************************************************************
*-----------------------------------------------------------------------
*     Common statements are implemented to make transferring variables from
* the MAIN DRIVER to OUTPUT and various other subroutines easier.
*-----------------------------------------------------------------------

      common/counters/ z,zz,i,ii,j,l,ll,nx,ny
      common/geometry/ width,depth,height,thickd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,f,
     $   perw,perd,pressuredrop,perimeter
      common/time/ timetot,dt
      common/distance/ dx,dy
      common/solution/ Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot
*-----------------------------------------------------------------------
*   Declare all variables that are used in the subroutine OUTPUT.
```

```
* Variables declared here are used throughout the subroutine as well as
* through out the program (if they are included in COMMON statements).
*-----------------------------------------------------------------------

      integer z,zz,i,ii,j,l,ll,nx,ny,NMAX2
      real f,timetot,dt
      double precision width,depth,height,thickd,perw,perd,
     $  Areatot,Areaair,Areas,Aread,Voltot,Volw,Massw,Massdc,
     $   pressuredrop,perimeter
      double precision dx,dy
      double precision Twtotpro,Wtotpro,Ywtotpro,Yatotpro,Tatotpro,
     $   Twtotreg,Wtotreg,Ywtotreg,Yatotreg,Tatotreg,Tatot,Ta,Ya,Yatot

*-----------------------------------------------------------------------
*     Initialize parameters used in the dimensioning of all arrays used in
* the OUTPUT and transferred between subroutines.
*-----------------------------------------------------------------------

      parameter (NMAX2=90000)

      dimension Twtotpro(NMAX2),Wtotpro(NMAX2),Ywtotpro(NMAX2),
     $   Yatotpro(NMAX2),Tatotpro(NMAX2),Twtotreg(NMAX2),
     $   Wtotreg(NMAX2),Ywtotreg(NMAX2),Yatotreg(NMAX2),
     $   Tatotreg(NMAX2)
*-----------------------------------------------------------------------
*    Record elapsed time and pressure drop as well as final nodal spacing
* in the output file.
*-----------------------------------------------------------------------

      write(44,*) 'Elapsed time = ',timetot,'seconds'
      write(44,*)
      write(44,*) 'Pressure Drop = ',pressuredrop,'m water'
      write(44,*)
      write(44,*) 'dx = ',dx,'m'
      write(44,*) 'dy=',dy,'m'
      write(44,*)
*-----------------------------------------------------------------------
*    Record exit supply and exhaust air and desiccant data at every 100
* time steps, for a more detailed look at the plate exchanger behavior.
*-----------------------------------------------------------------------

        do j=INT(dt/dt),INT(timetot/dt),100*INT(dt/dt)
 write(44,*) 'Time step value :',j
 write(44,*) 'SUPPLY'
 write(44,*) 'Tair (Celcius)= ',REAL(Tatotpro(j))
 write(44,*) 'Yair (kg water/kg air)= ',REAL(Yatotpro(j))
 write(44,*) 'Tbed (Celcius)= ',REAL(Twtotpro(j))
 write(44,*) 'Ybed (kg water/kg desiccant)= ',REAL(Wtotpro(j))
         write(44,*) 'Ypores (kg water/kg air)= ',REAL(Ywtotpro(j))
 write(44,*) 'EXHAUST'
 write(44,*) 'Tair (Celcius)= ',REAL(Tatotreg(j))
 write(44,*) 'Yair (kg water/kg air)= ',REAL(Yatotreg(j))
 write(44,*) 'Tbed (Celcius)= ',REAL(Twtotreg(j))
 write(44,*) 'Ybed (kg water/kg desiccant)= ',REAL(Wtotreg(j))
         write(44,*) 'Ypores (kg water/kg air)= ',REAL(Ywtotpro(j))
 write(44,*)
       end do
*-----------------------------------------------------------------------
*   Return to the main driver
*-----------------------------------------------------------------------

      return

      end
```

Vita

Christie Staton was raised in York County, Virginia. Upon graduating from Tabb High School in 1991, she enrolled at Virginia Tech. During her undergraduate career, she participated in the Cooperative Education Program which provided her with the opportunity to work at NASA's Langley Research Center for four semesters. She graduated with her B.S. degree in Mechanical Engineering in May of 1996. The submittal of this thesis fulfills her graduate requirements for a M.S. degree in Mechanical Engineering from Virginia Tech.

Permanent Address: 1406 Ascot Lane
Blacksburg, VA 24060

This thesis was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Greg Walker, Department of Mechanical Engineering, Virginia Tech.