MATHEMATICAL MODELING OF EVAPORATIVE COOLING OF MOISTURE BEARING EPOXY COMPOSITE PLATES

A Thesis

by

GREGORY STEVEN PAYETTE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2006

Major Subject: Mechanical Engineering

MATHEMATICAL MODELING OF EVAPORATIVE COOLING OF MOISTURE BEARING EPOXY COMPOSITE PLATES

A Thesis

by

GREGORY STEVEN PAYETTE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Roger Morgan
	J. N. Reddy
Committee Member,	Jose Roesset
Head of Department,	Dennis O'Neal

May 2006

Major Subject: Mechanical Engineering

ABSTRACT

Mathematical Modeling of Evaporative Cooling of

Moisture Bearing Epoxy Composite Plates. (May 2006)

Gregory Steven Payette, B.S., University of Idaho

Co-Chairs of Advisory Committee: Dr. R. Morgan
Dr. J. Reddy

Research is performed to assess the potential of surface moisture evaporative cooling from composite plates as a means of reducing the external temperature of military aircraft. To assess the feasibility of evaporative cooling for this application, a simplified theoretical model of the phenomenon is formulated. The model consists of a flat composite plate at an initial uniform temperature, T_0. The plate also possesses an initial moisture (molecular water) content, M_0. The plate is oriented vertically and at t=0 s, one surface is exposed to a free stream of air at an elevated temperature. The other surface is exposed to stagnant air at the same temperature as the plate's initial temperature.

The equations associated with energy and mass transport for the model are developed from the conservation laws per the continuum mechanics hypothesis. Constitutive equations and assumptions are introduced to express the two nonlinear partial differential equations in terms of the temperature, T, and the partial density of molecular water, ρ_w. These equations are approximated using a weak form Galerkin finite element formulation and the α–family of time approximation. An algorithm and

accompanying computer program written in the Matlab programming language are presented for solving the nonlinear algebraic equations at successive time steps. The Matlab program is used to generate results for plates possessing a variety of initial moisture concentrations, M_0, and diffusion coefficients, D.

Surface temperature profiles, over time, of moisture bearing specimens are compared with the temperature profiles of dry composite plates. It is evident from the results that M_0 and D affect the surface temperature of a moist plate. Surface temperature profiles are shown to decrease with increasing M_0 and/or D. In particular, dry and moist specimens are shown to differ in final temperatures by as much as 30°C over a 900 s interval when M_0 = 30% and D is on the order of 10^{-8}m^2/s (T_0 = 25°C, h = 60 W/m^2°C, T_∞ = 90°C).

To my wife Cami and our son Nathan

And to my parents

ACKNOWLEDGEMENTS

I would like to thank my thesis committee co–chair Dr. Roger Morgan for the opportunity to work for him on this project. His supervision, leadership and sense of humor made working for him a delightful experience. I would also like to thank Dr. J. N. Reddy who also served as a co–chair of my committee. I was always grateful for the technical support that he was able to provide. In spite of his demanding schedule he always made time for my questions.

I must also thank my father Steven Payette. It was his example that started me on the path of engineering and his encouragement that convinced me to pursue an advanced degree.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

OVERVIEW

Limiting the external temperature of current stealth aircraft is of major concern to the United States Air Force. As the name suggests, a primary function of stealth vehicles is to avoid detection. Detection is often associated with the visibility of an aircraft to radar signature devices. The design of current stealth aircraft is such that they have extremely small radar signatures which are often comparable to those of small birds. However, stealth aircraft are becoming more vulnerable to detection due to recent improvements in the IR signature detection devices of land–to–air missile systems. If an aircraft deviates from its surroundings by only 1°C, detection can be achieved at military useful ranges [1]. Present stealth aircraft utilize cooling pipes within the wings and engine exhausts to limit external temperatures and thereby the associated IR signatures. However, more efficient fast, instantaneous cooling mechanisms are needed. Several alternative methods for reducing external temperatures have been proposed. In the present work the feasibility of one such method is mathematically assessed; namely the utilization of evaporative moisture surface cooling via the embedding of inorganic hydrate and moisture bearing fiber and particulate fillers into the surface layers of stealth aircraft.

This thesis follows the style of *Journal of Composite Materials*.

EVAPORATIVE COOLING

Evaporative cooling is deemed to be an appropriate alternative mechanism for the cooling of stealth aircraft due to its simplicity, as well as its success as a cooling mechanism in other applications. Evaporative cooling of liquid water occurs when the surface of a body of water or moist object is exposed to an open environment which is commonly air. Under these conditions the water will begin to evaporate. This is due to the natural tendency of liquid water to seek to achieve phase equilibrium with the moisture content of the environment [2]. As water evaporates the latent heat of the vaporized water, or heat of vaporization, is absorbed from the body and the surrounding environment. In the absence of other mechanisms of heat transfer (i.e., convection and radiation), a net cooling effect of the body's surface is experienced. This phenomenon has been utilized for cooling for thousands of years in biological systems and engineering designs.

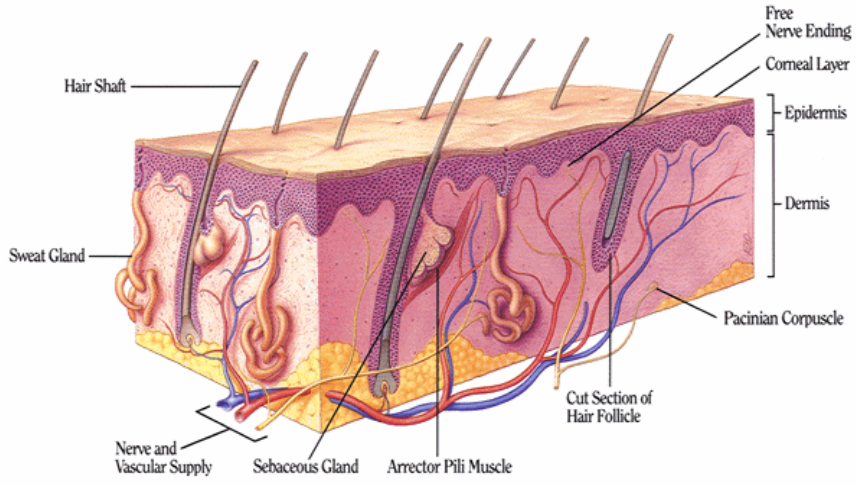

Figure 1. The structure of skin.

In biological systems the eccrine sweat gland in the skin executes the primary function of evaporative moisture cooling (see Figures 1 and 2). These sweat glands are simple coils of colloidal epithelium, which produce 0.5-10 liters/day of 99% pure water. Engineering systems that have successfully utilized evaporative cooling include evaporative coolers (swamp coolers), cooling towers, sweat cooling devices, and processes associated with the cooling and drying of food products just to name a few. In each of these systems evaporative cooling has been demonstrated to be a very useful mechanism for cooling.

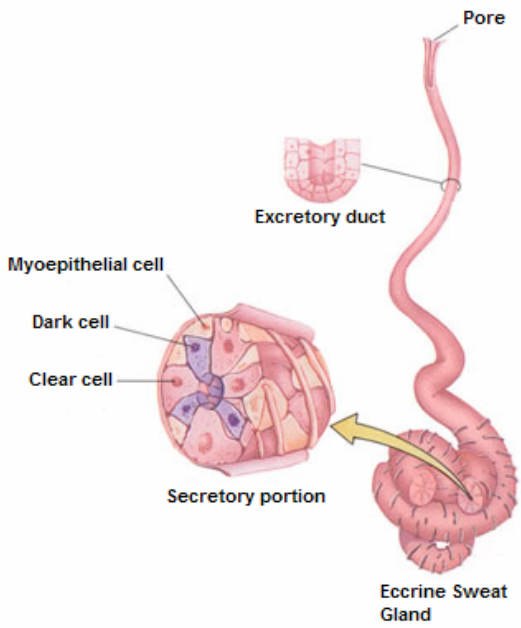

Figure 2. Eccrine sweat gland.

In addition to the aforementioned applications of evaporative cooling, it is believed that high moisture bearing fibers (up to 30 wt% moisture) of polyamidobenzimidazole were used by the former Soviet Union in their intercontinental

ballistic missile rocket motor casings. The purpose of these composites was to limit thermal–induced damage from laser threats by countering the potential heat of lasers with the latent heat of the imbedded moisture [3]. Subsequent studies have demonstrated that such composites can dissipate up to an order of magnitude greater thermal energy compared to dry composites [4].

OBJECTIVE

It has been proposed that high moisture particulate fillers be incorporated into high temperature polyimides, namely AFR-PEPA-N polyimides. It is anticipated that these polyimides will be able to be utilized as outer surfaces at critical locations (locations of high heat exposure) on military stealth aircraft. When these sections are exposed to high temperature airflows, the imbedded molecular moisture will be transported by diffusion to the outer surface of the composite where evaporation will occur. It is expected that the latent heat released due to evaporation will tend to reduce the effects of the high temperature airflows on the surface temperatures of stealth aircraft.

The main objective of the present study, therefore, is to mathematically evaluate the proposed alternative method for reducing the external temperatures of military aircraft. This is accomplished by

1. Formulating a simplified physical model of the evaporative cooling phenomenon.

2. Deriving the governing differential equations describing the energy and mass transfer per the simplified model from the conservation laws of continuum mechanics.
3. Simplifying the governing equations through appropriate assumptions and constitutive relationships.
4. Developing Galerkin weak form finite element models of the governing equations.
5. Creating a computer algorithm for solving the governing equations.
6. Utilizing the computer algorithm to generate results for dry composite specimens and moisture bearing composite specimens.
7. Evaluating the results to assess the potential of surface moisture evaporative cooling of moisture bearing composites as a means of reducing the external temperature of military aircraft.

CHAPTER II

DEVELOPMENT OF GOVERNING EQUATIONS

PHYSICAL MODEL

To address the potential of evaporative cooling as a means of reducing the external temperature of stealth aircraft, it is first necessary to develop a physical model of the phenomena and the corresponding governing equations associated with that model. In the interest of computational time it is convenient to develop a model which is as simple as possible yet complex enough to accurately assess the potential of evaporative cooling for the current application. Since subsequent experimental research will also be performed to assess the feasibility of the present study it also important to produce a model whose predictions may be compared with experimental data.

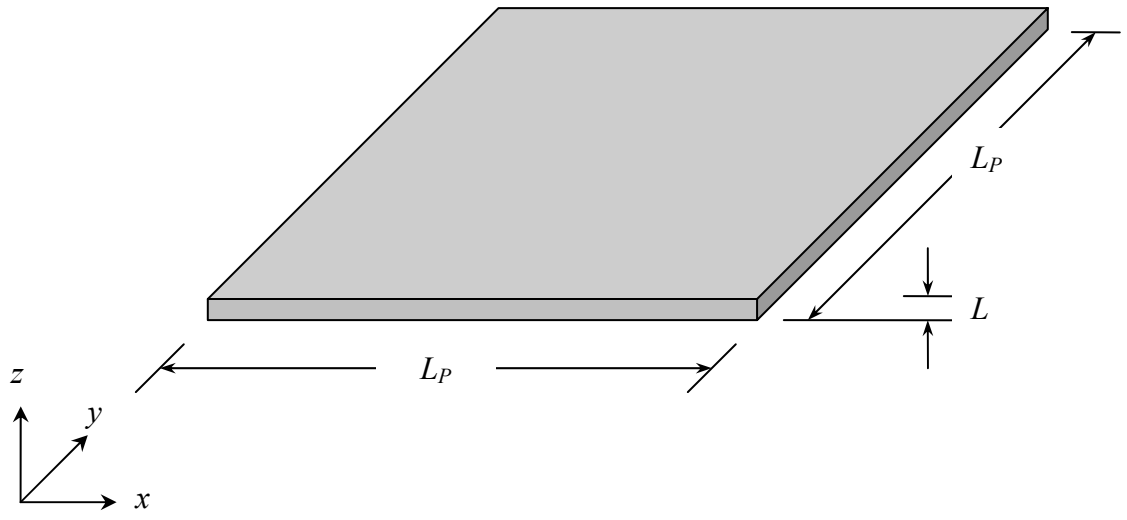

Figure 3. Thin flat plate where $L \ll L_P$.

The model chosen for the present study is a flat plate of dimensions $L_P \times L_P \times L$. An illustration of this model is presented in Figure 3. The plate possesses a uniform initial temperature and molecular moisture content. At a given time, $t=0$, the plate is exposed on both surfaces to streams of air containing known free stream temperatures and moisture contents. As a result heat and moisture will begin to transfer through the plate (see Figure 4).

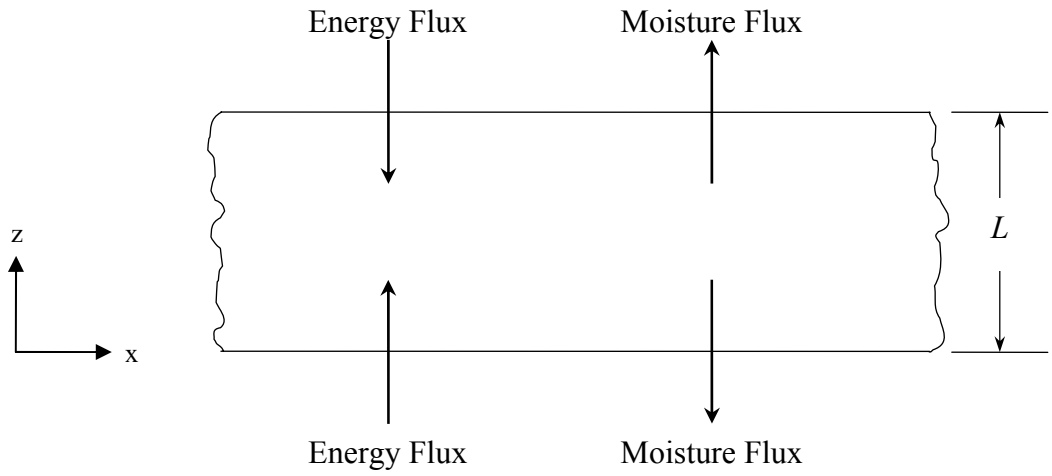

Figure 4. Heat and moisture transport.

ASSUMPTIONS

It is necessary to impose assumptions upon the physical model to simplify the mathematical equations that will be formulated. The overall effect of these assumptions on the behavior of the plate is considered to be negligible in the analysis. The assumptions are as follows:

1. The moisture bearing composite plate is isotropic. This allows for the use of Fourier's law of heat conduction.
2. The convective heat and mass transfer coefficients on both sides of the plate are uniform. Thus at any z location within the plate and for all values of time, t, the temperature and molecular moisture content will be uniform in the x and y directions (i.e., independent of x or y). This reduces the governing equations to one dimensional form.
3. Heat transfer at the plate surfaces due to radiation is negligible.
4. Diffusion of molecular moisture within the composite plate may be described by Fick's Law of Diffusion.
5. The rate of evaporation is small and does not significantly affect the boundary layer of the air flowing over the plate. As a result the convective heat and mass transfer coefficients are considered to be independent of the rate of mass transfer.
6. The temperature of the free stream air is less than 100°C. In addition, evaporation is assumed to occur at the plate surface and not within the plate.
7. The density of the water vapor at the surface of the plate is approximated as

$$\rho_s = \rho_{sat}\left(T_s\right) m_w \tag{2.1}$$

where m_w is the mass fraction of water just below the plate surface (inside the plate) and ρ_s is the partial density of water vapor just above (or below) the plate surface. In addition T_s is the surface temperature.

8. The thickness of the plate (z-direction) is unaffected by changes in molecular moisture content.
9. Surface tension, stress tensor components, and capillary forces are negligible (i.e., zero).

With the preceding physical model and assumptions, it is possible to derive the equations governing the behavior of the flat plate depicted in Figure 3 from the laws of physics. In the context of the present work the "laws of physics" is considered to be synonymous with the conservation axioms and entropy inequality per the continuum mechanics hypothesis. In formulating the equations governing the evaporative cooling of composite plates, it is necessary to consider two fundamental laws, namely conservation of energy and conservation of mass.

FIRST AXIOM OF THERMODYNAMICS

For many physical phenomena the heat equation is sufficient for performing heat transfer analysis. Its applicability, however, is generally restricted to modeling physical occurrences where heat is transferred in the absence of other forms of energy transport (i.e., mass diffusion) [5]. For phenomena involving coupled heat and mass transfer, the standard heat equation can be modified to account for the transfer of energy by diffusion.

The first axiom of thermodynamics which is also known as the first law of thermodynamics or conservation of energy principle may be used to derive the governing energy equation. This axiom applied to a continuum is the assertion that the time rate of change in internal and kinetic energies is equal to the rates of heat addition and work of the applied forces. This is expressed mathematically over an arbitrary region as [6,7]

$$\frac{D}{Dt}\int_R \rho\left(\varepsilon+\frac{1}{2}\mathbf{v}^2\right)dR=\oint_S\left(\mathbf{v}\cdot\overline{\overline{\boldsymbol{\sigma}}}\cdot\mathbf{n}-\mathbf{q}\cdot\mathbf{n}\right)dS+\int_R\rho\left(\mathbf{v}\cdot\mathbf{b}+r\right)dR \tag{2.2}$$

In Eq. (2.2) the quantities ρ, ε, $\boldsymbol{v}$, $\overline{\overline{\boldsymbol{\sigma}}}$, $\boldsymbol{q}$, $\boldsymbol{b}$, r, and $\boldsymbol{n}$ are defined as density, internal energy, velocity, stress, heat flux, body force, heat supply density and unit normal vector. Bolded quantities denote vectors. Double arrows above bolded quantities signify second order tensors. Furthermore, in Eq. (2.2), R and S refer to the arbitrary region (volume) and the closed surface of that region respectively. The derivative operator per Eq. (2.2) represents the *substantial derivative* or *material derivative* operator that is commonly used in the Lagrangian frame of reference.

Applying Leibniz's rule [6] allows Eq. (2.2) to be rewritten in the Eulerian frame of reference as

$$\begin{aligned}\int_R\left\{\frac{\partial}{\partial t}\left[\rho\left(\varepsilon+\frac{1}{2}\mathbf{v}^2\right)\right]\right\}dR+\oint_S\left\{\left[\rho\left(\varepsilon+\frac{1}{2}\mathbf{v}^2\right)\right]\mathbf{v}\cdot\mathbf{n}\right\}dS=\oint_S\left(\mathbf{v}\cdot\overline{\overline{\boldsymbol{\sigma}}}\cdot\mathbf{n}-\mathbf{q}\cdot\mathbf{n}\right)dS\\+\int_R\rho\left(\mathbf{v}\cdot\mathbf{b}+r\right)dR\end{aligned} \tag{2.3}$$

Gauss's divergence theorem [6] is next used to write Eq. (2.3) as a single integral statement over the arbitrary region R.

$$\int_R \left\{ \frac{\partial}{\partial t}\left[\rho\left(\varepsilon + \frac{1}{2}\mathbf{v}^2 \right) \right] + \nabla \cdot \left[\rho\left(\varepsilon + \frac{1}{2}\mathbf{v}^2 \right)\mathbf{v} \right] - \nabla \cdot \left(\mathbf{v} \cdot \bar{\bar{\boldsymbol{\sigma}}} - \mathbf{q} \right) - \rho\left(\mathbf{v} \cdot \mathbf{b} + r \right) \right\} dR = 0 \quad (2.4)$$

Requiring body forces, heat supply density and all stress tensor components to be zero allows the above equation to be simplified considerably. The result is

$$\int_R \left\{ \frac{\partial}{\partial t}\left[\rho\left(\varepsilon + \frac{1}{2}\mathbf{v}^2 \right) \right] + \nabla \cdot \left[\rho\left(\varepsilon + \frac{1}{2}\mathbf{v}^2 \right)\mathbf{v} \right] + \nabla \cdot \mathbf{q} \right\} dR = 0 \quad (2.5)$$

Order of magnitude analysis allows for one final simplification of Eq. (2.5). If the mass average velocity of the material is small (i.e., $\|\mathbf{v}\| \ll 1$), the kinetic energy terms are negligible as they are determined from the square of the velocity. These terms are therefore higher order and may be neglected in the analysis. Under this condition, Eq. (2.5) simplifies to

$$\int_R \left[\frac{\partial(\rho\varepsilon)}{\partial t} + \nabla \cdot (\rho\varepsilon\mathbf{v}) + \nabla \cdot \mathbf{q} \right] dR = 0 \quad (2.6)$$

Since the region chosen in Eq. (2.2) was arbitrary it follows that the integrand of Eq. (2.6) must be zero. Thus the differential energy equation becomes

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \nabla \cdot (\rho\varepsilon\mathbf{v}) + \nabla \cdot \mathbf{q} = 0 \quad (2.7)$$

which is a common form of the energy equation in the literature [5,8,9].

AXIOM OF CONSERVATION OF MASS

The conservation of mass principle may be utilized to derive the governing differential equation describing mass transport. Conservation of mass states that the time rate of change of mass is constant. This principle may be applied to the total mass

or the individual masses of species in a mixture [10]. The conservation of mass principle applied to molecular water imbedded in an epoxy composite in an arbitrary region implies that [6]

$$\frac{D}{Dt}\int_R \rho_w dR = 0 \tag{2.8}$$

It should be noted that Eq. (2.8) assumes that there is no generation of molecular water from chemical reactions. This is reasonable for the present analysis. Applying Leibniz's rule allows Eq. (2.8) to be rewritten as

$$\int_R \frac{\partial \rho_w}{\partial t} dR + \oint_S \rho_w \mathbf{v}\cdot\mathbf{n} dS = 0 \tag{2.9}$$

Gauss's divergence theorem is next used to write Eq. (2.9) as a single integral statement over the region.

$$\int_R \left[\frac{\partial \rho_w}{\partial t} + \nabla\cdot\left(\rho_w \mathbf{v}_w\right) \right] dR = 0 \tag{2.10}$$

Since the region chosen in Eq. (2.8) is arbitrary it follows that the integrand of Eq. (2.10) must be zero. This implies that

$$\frac{\partial \rho_w}{\partial t} + \nabla\cdot\left(\rho_w \mathbf{v}_w\right) = 0 \tag{2.11}$$

CONSTITUTIVE RELATIONSHIPS

In their present forms Eqs. (2.7) and (2.11) are indeterminate as there are more unknowns than equations. For this reason it is necessary to introduce constitutive

relationships to allow these equations to be solved. The first constitutive relationship that will be introduced relates internal energy to enthalpy, pressure and density as [11]

$$\varepsilon = \hat{h} - \frac{p}{\rho} \tag{2.12}$$

where the quantities $\hat{h}$, and p represent the enthalpy and pressure of the mixture respectively.

The material studied in this analysis is treated as a binary mixture of molecular water imbedded in a solid epoxy composite plate. The product $\rho\hat{h}\mathbf{v}$ may therefore be expressed as a summation of contributions of the individual species of the mixture [10]. This is written as

$$\rho\hat{h}\mathbf{v} = \sum_{i=1}^{n} \rho_i \hat{h}_i \mathbf{v}_i \tag{2.13}$$

Eq. (2.13) is applicable to a mixture composed of n species. The quantity ρ_i in the above equation is the *partial density* of a particular species [10]. Partial density is defined as the mass of constituent i in a small volume of a mixture divided by the volume of that mixture. The quantities $\hat{h}_i$ and $\mathbf{v}_i$ are the enthalpy and absolute velocity of the i^{th} species of the mixture. In the present study n=2 and Eq. (2.13) reduces to

$$\rho\hat{h}\mathbf{v} = \rho_e \hat{h}_e \mathbf{v}_e + \rho_w \hat{h}_w \mathbf{v}_w \tag{2.14}$$

where the subscripts e and w refer to the epoxy matrix and the imbedded molecular water respectively. Since the epoxy is stationary, Eq. (2.14) further reduces to

$$\rho\hat{h}\mathbf{v} = \rho_w \hat{h}_w \mathbf{v}_\mathbf{w} \tag{2.15}$$

In the literature [10,11], the *mass average velocity*, **v**, of a binary system is defined by the following equation

$$\rho \mathbf{v} = \rho_e \mathbf{v}_\mathbf{e} + \rho_w \mathbf{v}_\mathbf{w} \tag{2.16}$$

Since the epoxy composite matrix is stationary Eq. (2.16) reduces to

$$\rho \mathbf{v} = \rho_w \mathbf{v}_\mathbf{w} = \mathbf{n}_\mathbf{w}'' \tag{2.17}$$

Eq. (2.17) illustrates that in the current study the mass average velocity multiplied by the density, $\rho\mathbf{v}$, is equivalent to the mass flux of the molecular water, $\mathbf{n}_\mathbf{w}''$, in the epoxy. The *mass flux of the molecular water relative to the average mass velocity* of the binary mixture is denoted by $\mathbf{j}_w$. It is defined as [10,11]

$$\mathbf{j}_\mathbf{w} = \rho_w \left(\mathbf{v}_\mathbf{w} - \mathbf{v} \right) \tag{2.18}$$

Eqs. (2.17) and (2.18) may be combined to relate $\mathbf{n}_\mathbf{w}''$ to $\mathbf{j}_w$ as

$$\mathbf{n}_\mathbf{w}'' = \frac{1}{1 - m_w} \mathbf{j}_\mathbf{w} \tag{2.19}$$

where m_w is the mass fraction of water in the mixture. It is the ratio of the partial density of molecular water to the total density of the mixture. It is expressed as

$$m_w = \frac{\rho_w}{\rho} = \frac{\rho_w}{\rho_e + \rho_w} \tag{2.20}$$

It is next important to introduce a constitutive relationship for $\mathbf{j}_w$. Under the condition of material symmetry (i.e., isotropic material) coupled with the assumption that water molecules diffuse in epoxy composites due to species concentration gradients, Fick's Law of Diffusion applies. Fick's Law is expressed as [10,11]

$$\mathbf{j_w} = -\rho D \nabla m_w \tag{2.21}$$

Fick's Law of Diffusion is a large assumption as mass diffusion may result from the combination of many factors including temperature and pressure gradients, as well as external forces [5,9]. Assuming Fick's Law to be adequate allows for the following assertion

$$\rho_w \mathbf{v_w} = \frac{1}{1-m_w}\mathbf{j_w} = -\frac{\rho D}{1-m_w}\nabla m_w \tag{2.22}$$

The next constitutive relationship that will be introduced is Fourier's Law of Heat Conduction [5] which relates the components of the heat flux vector, **q**, to temperature, *T*. This set of equations is applicable to isotropic materials and is written as

$$\mathbf{q} = -k\nabla T \tag{2.23}$$

Enthalpy may also be related to the temperature by

$$\hat{h} = c_P T \tag{2.24}$$

where c_p is the effective specific heat of the mixture of epoxy / molecular moisture.

ENERGY EQUATION

By utilizing the constitutive relationships developed in the preceding section, the energy equation may be expressed in terms of just two dependent variables, namely the temperature, *T*, and the partial density of the molecular water, ρ_w. Inserting Eq. (2.12) into the energy equation, Eq. (2.7), and neglecting the effects of pressure variations results in

$$\frac{\partial\left(\rho\hat{h}\right)}{\partial t}+\nabla\cdot\left(\rho\hat{h}\mathbf{v}\right)+\nabla\cdot\mathbf{q}=0 \tag{2.25}$$

Applying Eq. (2.15) allows the above to be rewritten as

$$\frac{\partial\left(\rho\hat{h}\right)}{\partial t}+\nabla\cdot\left(\rho_w\hat{h}_w\mathbf{v}_\mathbf{w}\right)+\nabla\cdot\mathbf{q}=0 \tag{2.26}$$

Substituting the results of Eqs. (2.17) and (2.19) into Eq. (2.26) results in

$$\frac{\partial\left(\rho\hat{h}\right)}{\partial t}+\nabla\cdot\left(\hat{h}_w\frac{1}{1-m_w}\mathbf{j}_\mathbf{w}\right)+\nabla\cdot\mathbf{q}=0 \tag{2.27}$$

Finally, by utilizing the constitutive relationships per Eqs. (2.21), (2.23) and (2.24) allows the energy equation to be expressed as

$$\frac{\partial}{\partial t}\left(\rho c_P T\right)=\nabla\cdot\left(k\nabla T\right)+\nabla\cdot\left(c_{P,w}T\frac{\rho D}{1-m_w}\nabla m_w\right) \tag{2.28}$$

where $c_{P,w}$ is the specific heat of water.

If heat transfer in one or more directions can be considered negligible then the energy equation simplifies considerably. For the case of the flat plate depicted in Figure 3, the thickness of the component is very small in comparison to the other two dimensions (i.e., $L \ll L_P$). Under these circumstances Eq. (2.28) reduces to the following one dimensional form

$$\frac{\partial}{\partial t}\left(\rho c_P T\right)=\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+\frac{\partial}{\partial z}\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right) \tag{2.29}$$

Equations similar to Eq. 2.29 have been presented and used in the literature [5,12].

Eq. (2.29) represents the governing nonlinear energy equation for the present study of coupled heat and mass transfer. It should be noted that this equation possesses two dependent variables, namely T and ρ_w. As a result the transport of energy is coupled to the mass transport of the molecular water. In the absence of mass transport, the governing three-dimensional form of the energy equation reduces to

$$\frac{\partial}{\partial t}\left(\rho c_P T\right) = \nabla \cdot \left(k \nabla T\right) \tag{2.30}$$

which is the classical heat diffusion equation [11].

MASS TRANSPORT EQUATION

As was mentioned previously the energy equation, Eq. (2.28), on its own is indeterminate as it contains two dependent variables, namely temperature, T, and partial density of molecular water, ρ_w. This is expected as this equation was derived to include the effects of mass transfer. It is for this reason that the conservation of mass principle is also needed for the present analysis.

The mass transport equation is produced by using the result of Eq. (2.17). This allows Eq. (2.11) to be written in terms of the mass flux of the water. This results in

$$\frac{\partial \rho_w}{\partial t} + \nabla \cdot \mathbf{n_w}'' = 0 \tag{2.31}$$

Using the result of Eq. (2.19) allows Eq. (2.31) to be written as

$$\frac{\partial \rho_w}{\partial t} + \nabla \cdot \left(\frac{1}{1 - m_w} \mathbf{j_w} \right) = 0 \tag{2.32}$$

The constitutive relationship employed to allow for the solving of Eq. (2.32) is Fick's Law of Diffusion which was presented in Eq. (2.21). Inserting this relationship into Eq. (2.32) results in

$$\frac{\partial \rho_w}{\partial t} = \nabla \cdot \left(\frac{\rho D}{1 - m_w} \nabla m_w \right) \tag{2.33}$$

which reduces to the following one–dimensional form for the case of a thin flat plate

$$\frac{\partial \rho_w}{\partial t} = \frac{\partial}{\partial z} \left(\frac{\rho D}{1 - m_w} \frac{\partial m_w}{\partial z} \right) \tag{2.34}$$

which is the classical mass transport equation [11].

CHAPTER III

FINITE ELEMENT MODEL OF GOVERNING EQUATIONS

The energy and mass transport equations, Eqs. (2.29) and (2.34), developed in Chapter II are nonlinear and coupled. The material properties (k, c_P and D) of these equations are functions of z, T and ρ_w and thus produce nonlinearity. Equation (2.29) contains the product of T and the derivative of ρ_w which further contributes to the nonlinear nature of the equations. Coupling is obvious as the energy equation contains derivatives of ρ_w. It is further demonstrated later in this chapter that coupling is also exhibited in the boundary conditions of Eqs. (2.29) and (2.34).

Unlike special cases of the classical heat and mass transfer equations, Eqs. (2.29) and (2.34) do not posses known analytical solutions. It is therefore necessary to seek approximate solutions of these equations that will yield sensible results. Several approximation methods for solving differential equations are available including *Finite Differences*, *Weighted Residuals*, *Finite Elements*, etc [13,14]. In the present work the governing equations are approximated using a *weak form Galerkin finite element* formulation [14].

WEAK FORM OF ENERGY EQUATION

The first step in creating the finite element model is to develop the *weak forms* of Eqs. (2.29) and (2.34). The explicit form of an equation accompanied by that equation's

boundary conditions (essential and natural) is often denoted as the *strong form* of an equation. The *weak form* of an equation (which is equivalent to the *strong form*) on the other hand is a single equation containing the natural boundary terms as well as a relaxation in the differentiability requirements of the dependent variable(s). As a result, the *weak form* is a suitable building block of a finite element model [13].

The weak form of Eq. (2.29) is constructed by first moving all terms of the equation to the left hand side. Since the approximate solution to be introduced will be incapable of satisfying the differential equation exactly, it follows that the equation will equal a residual value. The residual R_h may be written as

$$R_h = \frac{\partial}{\partial t}\left(\rho c_P T\right) - \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) - \frac{\partial}{\partial z}\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right) \tag{3.1}$$

Eq. (3.1) is next multiplied by a weighting function w and integrated over the physical domain. The integral of this weighted residual is required to be zero. This is expressed mathematically as

$$\int_{\Omega}\left[w\frac{\partial}{\partial t}\left(\rho c_P T\right) - w\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) - w\frac{\partial}{\partial z}\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right]d\Omega = 0 \tag{3.2}$$

where Ω represents the volume of the body which is in this case a thin flat plate.

The physical domain is next discretized into a set of finite elements. An illustration of this process is presented in Figure 5. In this figure L is the thickness of the plate, e is the integer assigned to the e^{th} element, NN is the number of nodes and NE is the number of elements in the finite element model. Figure 5 also illustrates the manner

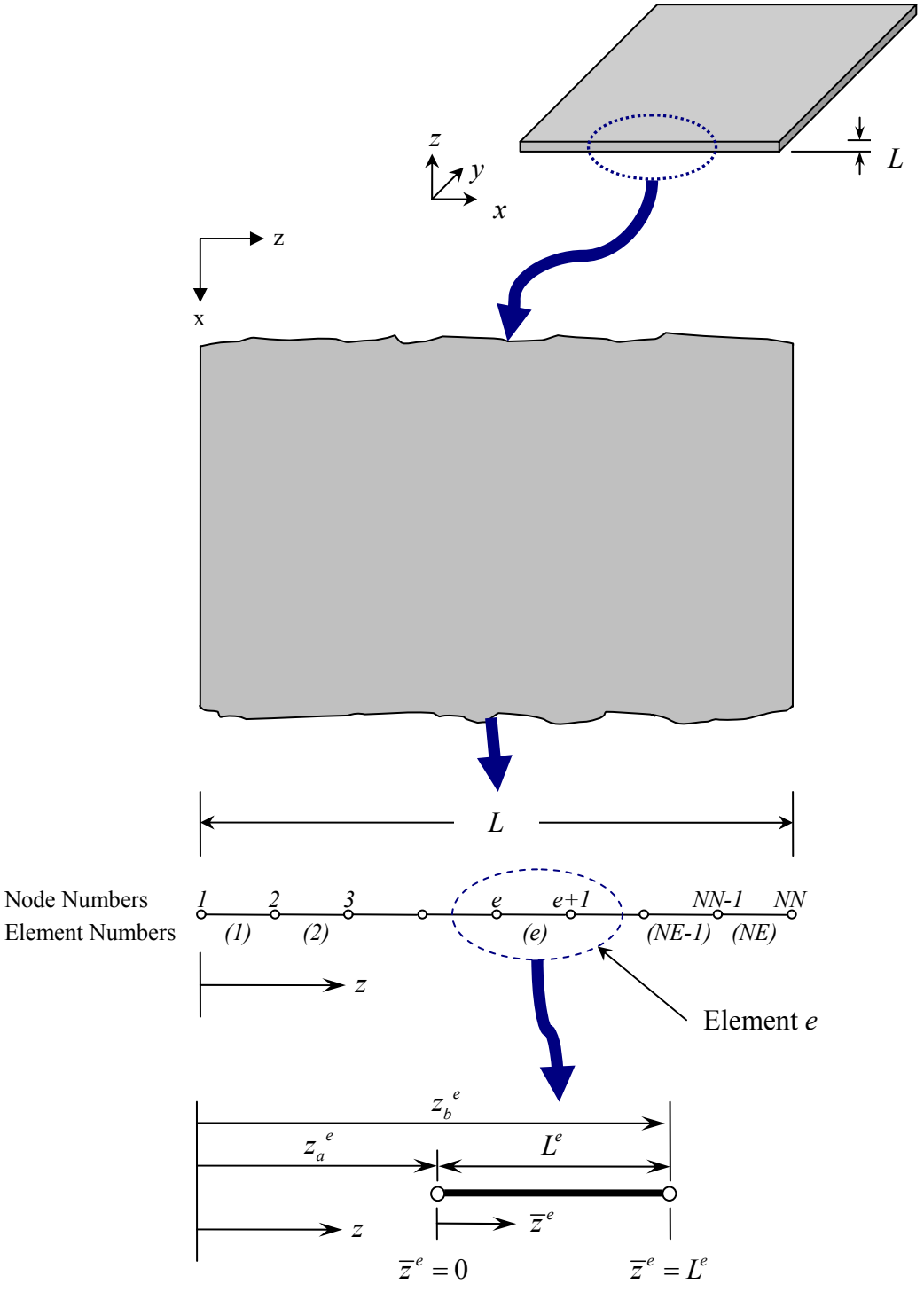

Figure 5. Thin flat plate discretized into a set of one dimensional finite elements.

in which the local coordinate $\bar{z}^e$ per element *e* is related to the global coordinate *z*. In addition, the quantity L^e is the length of element *e*.

With the domain discretized as depicted in Figure 5, Eq. (3.2) may be written for the e^{th} element of the model as

$$\int_{z_a^e}^{z_b^e}\left[w\frac{\partial}{\partial t}\left(\rho c_P T\right) - w\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) - w\frac{\partial}{\partial z}\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right] dz = 0 \tag{3.3}$$

Eq. (3.3) is thus applicable to each element of the model. The limits of integration $z_a^{\,e}$ and $z_b^{\,e}$ are the boundaries of the e^{th} element of the model in the *z* direction as is shown in Figure 5.

The next step in creating the weak form of the energy equation is to perform integration by parts on the second and third terms of the integrand of Eq. (3.3). This operation relaxes the differentiability requirements of the approximate solution (from second order to first) while producing the natural boundary terms associated with the equation.

$$\begin{aligned}\int_{z_a^e}^{z_b^e}\left[w\frac{\partial}{\partial t}\left(\rho c_P T\right) + \frac{\partial w}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + \frac{\partial w}{\partial z}\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right] dz \\ -\left[w\left(k\frac{\partial T}{\partial z}\right)\right]_{z_a^e}^{z_b^e} - \left[w\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right]_{z_a^e}^{z_b^e} = 0\end{aligned} \tag{3.4}$$

Eq. (3.4) may be further reduced to

$$\begin{aligned}\int_{z_a^e}^{z_b^e}\left[w\frac{\partial}{\partial t}\left(\rho c_P T\right) + \frac{\partial w}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + \frac{\partial w}{\partial z}\left(c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right] dz \\ - w\left(z_a^{\,e}\right)\hat{Q}_1^{\,e} - w\left(z_b^{\,e}\right)\hat{Q}_2^{\,e} = 0\end{aligned} \tag{3.5}$$

where

$$\hat{Q}_1^{\,e} = \left(-k\frac{\partial T}{\partial z} - c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z} \right)\Bigg|_{z_a^{\,e}} \qquad \hat{Q}_2^{\,e} = \left(k\frac{\partial T}{\partial z} + c_{P,w}T\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z} \right)\Bigg|_{z_b^{\,e}} \tag{3.6}$$

The quantities $\hat{Q}_1^{\,e}$ and $\hat{Q}_2^{\,e}$ per Eq. (3.6) are the secondary variables [13] associated with the energy equation. They are flux quantities representing the energy convecting from the boundaries of element *e*. Equation (3.5) is therefore the weak form of Eq. (2.29).

WEAK FORM OF MASS TRASPORT EQUATION

The development of the weak form of the mass transport equation follows the same steps used to develop the weak form of the energy equation. Since Eq. (2.34) cannot be satisfied exactly, it will instead be equal to some residual value R_m. This is expressed as

$$R_m = -\frac{\partial}{\partial z}\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z} \right) + \frac{\partial \rho_w}{\partial t} \tag{3.7}$$

Multiplying Eq. (3.7) by the weighting function *w* and requiring that the integral of the weighted residual be zero results in

$$\int_\Omega \left[-w\frac{\partial}{\partial z}\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z} \right) + w\frac{\partial \rho_w}{\partial t} \right] d\Omega = 0 \tag{3.8}$$

Discretizing the domain into the same set of finite element used to approximate the energy equation results in

$$\int_{z_a^{\,e}}^{z_b^{\,e}} \left[-w\frac{\partial}{\partial z}\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z} \right) + w\frac{\partial \rho_w}{\partial t} \right] dz = 0 \tag{3.9}$$

Integration by parts is next performed on the first term of the integrand of Eq. (3.9). This operation results in

$$\int_{z_a^e}^{z_b^e}\left[\frac{\partial w}{\partial z}\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)+w\frac{\partial \rho_w}{\partial t}\right]dz-\left[w\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right]_{z_a^e}^{z_b^e}=0 \tag{3.10}$$

which may be further reduced to

$$\int_{z_a^e}^{z_b^e}\left[\frac{\partial w}{\partial z}\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)+w\frac{\partial \rho_w}{\partial t}\right]dz-w\left(z_a{}^e\right)\hat{Q}_3{}^e-w\left(z_b{}^e\right)\hat{Q}_4{}^e=0 \tag{3.11}$$

where

$$\hat{Q}_3{}^e=\left.\left(-\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right|_{z_a{}^e} \qquad \hat{Q}_4{}^e=\left.\left(\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right)\right|_{z_b{}^e} \tag{3.12}$$

Equation (3.11) is therefore the weak form of the nonlinear mass transport equation, Eq. (2.34). It should be noted that the quantities $\hat{Q}_3{}^e$ and $\hat{Q}_4{}^e$ per Eq. (3.11) are the secondary variables of the mass transport equation. They represent the mass diffusing from the boundaries of element *e*.

It should be obvious that the developed weak forms, Eqs. (3.5) and (3.11), share common attributes. Both possess primary variables and flux terms. Because of these similarities a general schematic is presented via Figure 6 that is applicable for both Eqs. (3.5) and (3.11). This figure depicts an enlarged version of the e^{th} element previously presented in Figure 5. In Figure 6 *u* is the primary dependent variable and the quantities $\hat{Q}_a{}^e$ and $\hat{Q}_b{}^e$ are the secondary variables of a particular equation. In the case of Eq. (3.5) $u=T$, $\hat{Q}_a{}^e=\hat{Q}_1{}^e$ and $\hat{Q}_b{}^e=\hat{Q}_2{}^e$. For Eq. (3.11) $u=\rho_w$, $\hat{Q}_a{}^e=\hat{Q}_3{}^e$ and $\hat{Q}_b{}^e=\hat{Q}_4{}^e$. It

should be noted that the node numbers per Figure 3 are the local node numbers of element *e*.

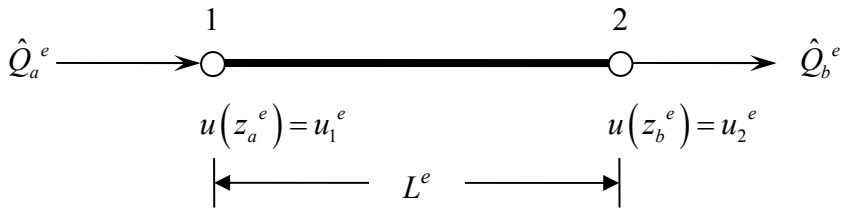

Figure 6. One dimensional linear finite element with primary and secondary variables.

FINITE ELEMENT MODEL OF ENERGY EQUATION

The finite element model of the energy equation is formulated directly from Eq. (3.5). As opposed to some approximation techniques, the finite element model is developed such that the unknown parameters are the dependent variables of the system at discrete locations (i.e., the nodes) [13,15]. In the present analysis, the variables are the temperature, *T*, and the partial density of molecular water, ρ_w. The solution introduced via the finite element method for each element is assumed to be of the following form [15]

$$T\left(z,t\right)=\sum_{j=1}^{n}T_j^{\,e}\left(t\right)\psi_j^{\,e}\left(z\right) \tag{3.13}$$

$$\rho_w\left(z,t\right)=\sum_{j=1}^{n}\rho_{w_j}^{\,e}\left(t\right)\psi_j^{\,e}\left(z\right) \tag{3.14}$$

where n denotes the number of nodes of a given finite element. The superscript e per Eqs. (3.13) and (3.14) signifies the element number of the finite element model. The quantities ψ_j^e are often called shape functions or interpolation functions.

In order to create a system of n linearly independent equations for each element, a set of weighting functions, w, are chosen to be equal to the shape functions (Galerkin approach [13]).

$$w = \psi_i^e(z) \tag{3.15}$$

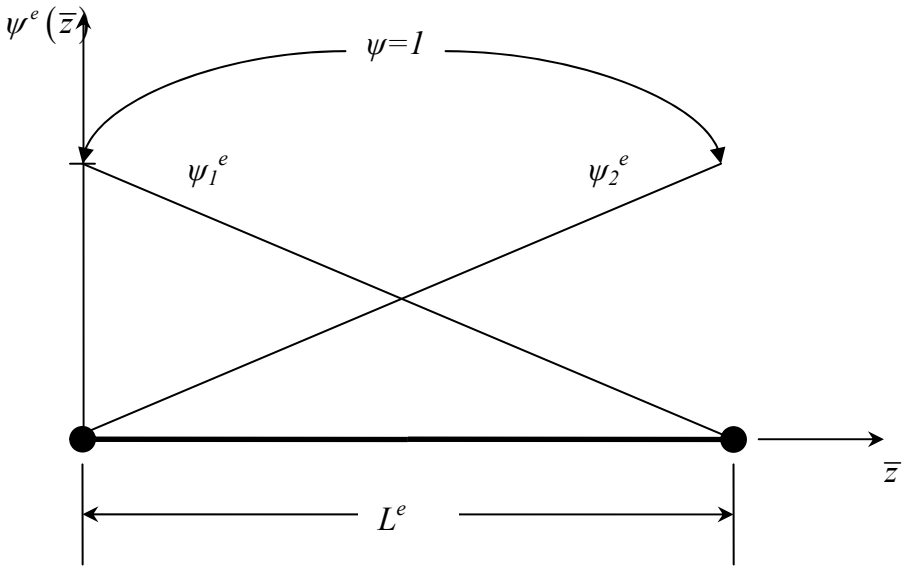

Figure 7. Shape functions of linear one–dimensional finite element.

The Lagrange shape functions for the one–dimensional linear finite element expressed in local coordinates are given as [15]

$$\psi_1^e(\bar{z}) = 1 - \frac{\bar{z}}{L^e} \tag{3.16}$$

$$\psi_2^{\,e}\left(\bar{z}\right)=\frac{\bar{z}}{L^e} \tag{3.17}$$

where L^e is the length of an element and is given by

$$L^e=z_b^{\,e}-z_a^{\,e} \tag{3.18}$$

The interpolation functions per Eqs. (3.16) and (3.17) are depicted graphically in local coordinates in Figure 7.

Prior to the creation of the finite element model, it should first be noted from Eq. (2.29) that the time derivative term may be rewritten via the product rule of differential calculus as

$$\frac{\partial}{\partial t}\left(\rho c_P T\right)=\frac{\partial}{\partial t}\left[\left(\rho_e c_{P,e}+\rho_w c_{P,w}\right)T\right]=\left(\rho_e c_{P,e}+\rho_w c_{P,w}\right)\frac{\partial T}{\partial t}+c_{P,w}T\frac{\partial \rho_w}{\partial t} \tag{3.19}$$

The above result combined with Eqs. (3.13), (3.14) and (3.15) allows Eq. (3.5) to be approximated for an arbitrary finite element as

$$\begin{aligned}
&\int_0^{L^e}\left\{\psi_i\left[\left(\rho_e c_{P,e}+\left(\sum_{j=1}^{n}\rho_{w_j}^{\,e}\psi_j\right)c_{P,w}\right)\left(\sum_{j=1}^{n}\frac{\partial T_j^{\,e}}{\partial t}\psi_j\right)+c_{P,w}\left(\sum_{j=1}^{n}T_j^{\,e}\psi_j\right)\left(\sum_{j=1}^{n}\frac{\partial \rho_{w_j}^{\;e}}{\partial t}\psi_j\right)\right]\right\}d\bar{z}\\
&+\int_0^{L^e}\left\{\frac{\partial \psi_i}{\partial \bar{z}}c_{P,w}\left(\sum_{j=1}^{n}T_j^{\,e}\psi_j\right)\frac{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}^{\,e}\psi_j\right)\right]D^e}{1-\frac{\left(\sum_{j=1}^{n}\rho_{w_j}^{\,e}\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}^{\,e}\psi_j\right)}}\frac{\partial}{\partial \bar{z}}\left[\frac{\left(\sum_{j=1}^{n}\rho_{w_j}^{\,e}\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}^{\,e}\psi_j\right)}\right]\right\}d\bar{z}\\
&+\int_0^{L^e}\left[\frac{\partial \psi_i}{\partial \bar{z}}k^e\left(\sum_{j=1}^{n}T_j^{\,e}\frac{\partial \psi_j}{\partial \bar{z}}\right)\right]d\bar{z}-\psi_i\left(0\right)\hat{Q}_1-\psi_i\left(L^e\right)\hat{Q}_2=0
\end{aligned} \tag{3.20}$$

The quotient rule of differential calculus may next be employed to simplify Eq. (3.20). Through the application of this rule it follows that

$$\frac{\partial}{\partial \bar{z}}\left[\frac{\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)}\right]=\frac{\rho_e\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\dfrac{\partial\psi_j}{\partial\bar{z}}\right)}{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)\right]^2} \tag{3.21}$$

Eq. (3.21) allows Eq. (3.20) to be rewritten as

$$\begin{aligned}
&\int_0^{L^e}\left\{\psi_i\left[\left(\rho_e c_{P,e}+\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)c_{P,w}\right)\left(\sum_{j=1}^{n}\frac{\partial T_j{}^{e}}{\partial t}\psi_j\right)+c_{P,w}\left(\sum_{j=1}^{n}T_j{}^{e}\psi_j\right)\left(\sum_{j=1}^{n}\frac{\partial \rho_{w_j}{}^{e}}{\partial t}\psi_j\right)\right]\right\}d\bar{z}\\
&+\int_0^{L^e}\left\{\frac{\partial\psi_i}{\partial\bar{z}}c_{P,w}\left(\sum_{j=1}^{n}T_j{}^{e}\psi_j\right)\frac{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)\right]D^e}{1-\dfrac{\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)}}\frac{\rho_e\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\dfrac{\partial\psi_j}{\partial\bar{z}}\right)}{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^{e}\psi_j\right)\right]^2}\right\}d\bar{z}\\
&+\int_0^{L^e}\left[\frac{\partial\psi_i}{\partial\bar{z}}k^e\left(\sum_{j=1}^{n}T_j{}^{e}\frac{\partial\psi_j}{\partial\bar{z}}\right)\right]d\bar{z}-\psi_i(0)\hat{Q}_1-\psi_i\left(L^e\right)\hat{Q}_2=0
\end{aligned} \tag{3.22}$$

A careful examination of Eq. (3.22) allows for drastic simplification. Carrying out the necessary algebraic steps allows Eq. (3.22) to be written as

$$\int_0^{L^e}\left[\psi_i\left(\rho_e c_{P,e}+\left(\sum_{j=1}^{n}{\rho_{w_j}}^e\psi_j\right)c_{P,w}\right)\left(\sum_{j=1}^{n}\frac{\partial {T_j}^e}{\partial t}\psi_j\right)\right]d\bar{z}$$
$$+\int_0^{L^e}\left[c_{P,w}\psi_i\left(\sum_{j=1}^{n}{T_j}^e\psi_j\right)\left(\sum_{j=1}^{n}\frac{\partial {\rho_{w_j}}^e}{\partial t}\psi_j\right)+c_{P,w}D^e\frac{\partial\psi_i}{\partial\bar{z}}\left(\sum_{j=1}^{n}{T_j}^e\psi_j\right)\left(\sum_{j=1}^{n}{\rho_{w_j}}^e\frac{\partial\psi_j}{\partial\bar{z}}\right)\right]d\bar{z} \quad (3.23)$$
$$+\int_0^{L^e}\left[\frac{\partial\psi_i}{\partial\bar{z}}k^e\left(\sum_{j=1}^{n}{T_j}^e\frac{\partial\psi_j}{\partial\bar{z}}\right)\right]d\bar{z}-\psi_i(0)\hat{Q}_1-\psi_i\left(L^e\right)\hat{Q}_2=0$$

It should be apparent that Eq. (3.20) is a set of *n* nonlinear equations that approximate the temperature field across a single element. These equations may be written in matrix form as

$$\left[{C_h}^e\right]\left\{\dot{T}^e\right\}+\left[{K_h}^e\right]\left\{T^e\right\}=\left\{{f_h}^e\right\} \quad (3.24)$$

where $\left\{\dot{T}^e\right\}$ is the time derivative of the nodal temperature vector. The matrices $\left[{C_h}^e\right]$, $\left[{K_h}^e\right]$ and column vector $\left\{{f_h}^e\right\}$ may be expressed in index notation as

$${C_{h_{i,j}}}^e=\int_0^{L^e}\psi_i\psi_j\left[\rho_e c_{P,e}+\left(\sum_{m=1}^{n}{\rho_{w_m}}^e\psi_m\right)c_{P,w}\right]d\bar{z} \quad (3.25)$$

$${K_{h_{i,j}}}^e=\int_0^{L^e}\left[k^e\frac{\partial\psi_i}{\partial\bar{z}}\frac{\partial\psi_j}{\partial\bar{z}}+c_{P,w}\psi_i\psi_j\left(\sum_{m=1}^{n}\frac{\partial {\rho_{w_m}}^e}{\partial t}\psi_m\right)+c_{P,w}D^e\left(\sum_{m=1}^{n}{\rho_{w_m}}^e\psi_m\right)\frac{\partial\psi_i}{\partial\bar{z}}\psi_j\right]d\bar{z} \quad (3.26)$$

$${f_{h_i}}^e=\psi_i(0)\hat{Q}_1+\psi_i\left(L^e\right)\hat{Q}_2 \quad (3.27)$$

Eqs. (3.25) and (3.26) may also be expressed in matrix form by introducing the $n\times1$ column vectors $\left\{\psi^e\right\}$, $\left\{B^e\right\}$, $\left\{{\rho_w}^e\right\}$ and $\left\{\dot{\rho}^e\right\}$. By definition the j^{th} row of $\left\{\psi^e\right\}$ contains the j^{th} interpolation function or ψ_j^e. Likewise, the j^{th} row of $\left\{B^e\right\}$, $\left\{{\rho_w}^e\right\}$ and

$\left\{\dot{\rho}_w^{\,e}\right\}$ contain $\frac{d\psi_j^{\,e}}{d\bar{z}}$, $\rho_j^{\,e}$ and $\dot{\rho}_j^{\,e}$ respectively. Using these definitions allows the matrices of Eq. (3.24) to be written as

$$\left[C_h^{\,e}\right]=\int_0^{L^e}\left\{\psi^e\right\}\left\{\psi^e\right\}^T\left(\rho_e c_{P,e}+c_{P,w}\left\{\rho_w^{\,e}\right\}^T\left\{\psi^e\right\}\right)d\bar{z} \tag{3.28}$$

$$\begin{aligned}\left[K_h^{\,e}\right]=&\int_0^{L^e}\left[k^e\left\{B^e\right\}\left\{B^e\right\}^T+c_{P,w}\left(\left\{\dot{\rho}_w^{\,e}\right\}^T\left\{\psi^e\right\}\right)\left\{\psi^e\right\}\left\{\psi^e\right\}^T\right]d\bar{z}\\&+\int_0^{L^e}\left[c_{P,w}D^e\left(\left\{\rho_w^{\,e}\right\}^T\left\{B\right\}\right)\left\{B^e\right\}\left\{\psi^e\right\}^T\right]d\bar{z}\end{aligned} \tag{3.29}$$

The matrix form of Eqs. (3.25), (3.26) and (3.27) may next be created for a particular element type (i.e., linear, quadratic, cubic, etc.). Selecting the linear element (n=2) depicted in Figure 7 with shape functions defined in Eqs. (3.16) and (3.17) allows the previously defined column vectors $\left\{\psi^e\right\}$, $\left\{B^e\right\}$, $\left\{\rho_w^{\,e}\right\}$ and $\left\{\dot{\rho}_w^{\,e}\right\}$ to be written as

$$\left\{\psi^e\right\}=\begin{pmatrix}1-\dfrac{\bar{z}}{L^e}\\[2ex]\dfrac{\bar{z}}{L^e}\end{pmatrix}\qquad\left\{B^e\right\}=\begin{pmatrix}-\dfrac{1}{L^e}\\[2ex]\dfrac{1}{L^e}\end{pmatrix}\qquad\left\{\rho_w^{\,e}\right\}=\begin{pmatrix}\rho_{w_1}^{\;\;e}\\[1ex]\rho_{w_2}^{\;\;e}\end{pmatrix}\qquad\left\{\dot{\rho}_w^{\,e}\right\}=\begin{pmatrix}\dot{\rho}_{w_1}^{\;\;e}\\[1ex]\dot{\rho}_{w_2}^{\;\;e}\end{pmatrix} \tag{3.30}$$

Inserting the vectors of Eq. (3.30) into Eqs. (3.28) and (3.29) and requiring that the material properties be constant for each element results in

$$\left[C_h^{\,e}\right]=\frac{\rho_e^{\,e}c_{P,e}^{\;\;e}L^e}{6}\begin{pmatrix}2&1\\1&2\end{pmatrix}+\frac{c_{P,w}^{\;\;e}L^e}{12}\begin{pmatrix}3\rho_{w_1}^{\;\;e}+\rho_{w_2}^{\;\;e}&\rho_{w_1}^{\;\;e}+\rho_{w_2}^{\;\;e}\\\rho_{w_1}^{\;\;e}+\rho_{w_2}^{\;\;e}&\rho_{w_1}^{\;\;e}+3\rho_{w_2}^{\;\;e}\end{pmatrix} \tag{3.31}$$

$$\left[K_h^{\,e}\right]=\frac{k^e}{L^e}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}+\frac{{c_{P,w}}^e L^e}{12}\begin{pmatrix}3\dot{\rho}_{w_1}{}^e+\dot{\rho}_{w_2}{}^e & \dot{\rho}_{w_1}{}^e+\dot{\rho}_{w_2}{}^e\\ \dot{\rho}_{w_1}{}^e+\dot{\rho}_{w_2}{}^e & \dot{\rho}_{w_1}{}^e+3\dot{\rho}_{w_2}{}^e\end{pmatrix}+\frac{{c_{P,w}}^e D^e\left({\rho_{w_2}}^e-{\rho_{w_1}}^e\right)}{2L^e}\begin{pmatrix}-1 & -1\\ 1 & 1\end{pmatrix} \tag{3.32}$$

$$\left\{f_h^{\,e}\right\}=\begin{pmatrix}\hat{Q}_1\\ \hat{Q}_2\end{pmatrix} \tag{3.33}$$

FINITE ELEMENT MODEL OF MASS TRANSPORT EQUATION

Like the energy equation the finite element model of the mass transport equation is formed directly from, Eq. (2.34), the governing differential equation of mass transfer. The methodology utilized to form the finite element model is akin to the process employed in the previous section for the energy equation. The assumed form of the solution is given by Eq. (3.14). In addition, the weighting functions are also specified by Eq. (3.15). It is important to note that *n* possesses the same numerical value for both the energy and mass transfer finite element models. This insures that the same elements are used for both models (i.e., linear, quadratic, cubic, etc.). These specifications allow the weak form given by Eq. (3.11) to be written in local coordinates for an arbitrary element as

$$\int_0^{L^e}\left\{\frac{\partial\psi_i}{\partial\bar{z}}\frac{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)\right]D^e}{1-\frac{\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)}}\frac{\partial}{\partial\bar{z}}\left[\frac{\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)}\right]+\psi_i\left(\sum_{j=1}^{n}\frac{\partial\rho_{w_j}{}^e}{\partial t}\psi_j\right)\right\}d\bar{z} \quad (3.34)$$
$$-w_i(0)\hat{Q}_3-w_i\left(L^e\right)\hat{Q}_4=0$$

Eq. (3.34) may be simplified via the quotient rule by using the result of Eq. (3.21). Inserting this result into Eq. (3.34) allows the finite element model of element *e* to be expressed as

$$\int_0^{L^e}\left\{\frac{\partial\psi_i}{\partial\bar{z}}\frac{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)\right]D^e}{1-\frac{\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)}{\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)}}\frac{\rho_e\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\frac{\partial\psi_j}{\partial\bar{z}}\right)}{\left[\rho_e+\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\psi_j\right)\right]^2}+\psi_i\left(\sum_{j=1}^{n}\frac{\partial\rho_{w_j}{}^e}{\partial t}\psi_j\right)\right\}d\bar{z} \quad (3.35)$$
$$-w_i(0)\hat{Q}_3-w_i\left(L^e\right)\hat{Q}_4=0$$

which simplifies to

$$\int_0^{L^e}\left[D^e\frac{\partial\psi_i}{\partial\bar{z}}\left(\sum_{j=1}^{n}\rho_{w_j}{}^e\frac{\partial\psi_j}{\partial\bar{z}}\right)+\psi_i\left(\sum_{j=1}^{n}\frac{\partial\rho_{w_j}{}^e}{\partial t}\psi_j\right)\right]d\bar{z}-w_i(0)\hat{Q}_3-w_i\left(L^e\right)\hat{Q}_4=0 \quad (3.36)$$

As was the case with Eq. (3.20), Eq. (3.36) is also a set of *n* nonlinear equations that approximate the partial density of molecular water across a single element as a function of position and time. These equations may be written in matrix form as

$$\left[C_m{}^e\right]\left\{\dot{\rho}_w{}^e\right\}+\left[K_m{}^e\right]\left\{\rho_w{}^e\right\}=\left\{f_m{}^e\right\} \tag{3.37}$$

These matrices may be expressed in index notation as

$$C_{m_{i,j}}{}^e=\int_0^{L^e}\psi_i{}^e\psi_j{}^e d\bar{z} \tag{3.38}$$

$$K_{m_{i,j}}{}^e=\int_0^{L^e}D^e\frac{\partial\psi_i{}^e}{\partial\bar{z}}\frac{\partial\psi_j{}^e}{\partial\bar{z}}d\bar{z} \tag{3.39}$$

$$f_{m_i}{}^e=\psi_i{}^e\left(0\right)\hat{Q}_3{}^e+\psi_i{}^e\left(L^e\right)\hat{Q}_4{}^e \tag{3.40}$$

Eqs. (3.38) and (3.39) may also be written in matrix form by utilizing the previously defined column vectors $\left\{\psi^e\right\}$ and $\left\{B^e\right\}$.

$$\left[C_m{}^e\right]=\int_0^{L^e}\left\{\psi^e\right\}\left\{\psi^e\right\}^T d\bar{z} \tag{3.41}$$

$$\left[K_m{}^e\right]=\int_0^{L^e}D^e\left\{B^e\right\}\left\{B^e\right\}^T d\bar{z} \tag{3.42}$$

The matrix form of Eqs. (3.25), (3.26) and (3.27) may next be created for a particular element type (i.e., linear, quadratic, cubic, etc.). To be consistent with the energy equation, the linear element (*n*=2) is also selected for the mass transport finite element model. Utilizing the column vectors defined by Eq. (3.30) allows $\left[C_m{}^e\right]$, $\left[K_m{}^e\right]$ and $\left\{f_m{}^e\right\}$ to be expressed as

$$\left[C_m{}^e\right] = \frac{L^e}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{3.43}$$

$$\left[K_m{}^e\right] = \frac{D^e}{L^e}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \tag{3.44}$$

$$\left\{f_m{}^e\right\} = \begin{pmatrix} \hat{Q}_3 \\ \hat{Q}_4 \end{pmatrix} \tag{3.45}$$

where D^e is considered to be constant for element e.

ELEMENT ASSEMBLY

The governing equations of energy and mass transfer have been approximated for each element of the finite element models. It is next necessary to assemble the approximate energy and mass transfer equations of the individual elements into two sets of global equations for the entire plate; one set approximating the energy equation and the other approximating the mass transport equation. Each set of equations is assembled by requiring continuity of the primary variables (i.e., temperature, T, and partial density of molecular water, ρ_w) and by balance of the secondary variables (i.e., heat and mass flux) [15]. Enforcing these conditions results in the following sets of nonlinear algebraic equations

$$\left[C_h\right]\left\{\dot{T}\right\} + \left[K_h\right]\left\{T\right\} = \left\{f_h\right\} \tag{3.46}$$

$$\left[C_m\right]\left\{\dot{\rho}_w\right\} + \left[K_m\right]\left\{\rho_w\right\} = \left\{f_m\right\} \tag{3.47}$$

If the global node numbering is in a consecutive order from the bottom of the plate to the top (i.e., follows the numbering scheme presented in Figure 5), then the matrices of Eqs. (3.46) and (3.47) will be of the form

$$[C_h]=\begin{pmatrix} c_{h_{1,1}}^{(1)} & c_{h_{1,2}}^{(1)} & 0 & \cdots & \cdots & \cdots & 0 \\ c_{h_{2,1}}^{(1)} & c_{h_{2,2}}^{(1)}+c_{h_{1,1}}^{(2)} & c_{h_{1,2}}^{(2)} & 0 & \cdots & \cdots & \vdots \\ 0 & c_{h_{2,1}}^{(2)} & c_{h_{2,2}}^{(2)}+c_{h_{1,1}}^{(3)} & c_{h_{1,2}}^{(3)} & 0 & \ddots & \vdots \\ \vdots & 0 & c_{h_{2,1}}^{(3)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & c_{h_{1,2}}^{(NE-1)} & 0 \\ \vdots & \vdots & \ddots & \ddots & c_{h_{2,1}}^{(NE-1)} & c_{h_{2,2}}^{(NE-1)}+c_{h_{1,1}}^{(NE)} & c_{h_{1,2}}^{(NE)} \\ 0 & \cdots & \cdots & \cdots & 0 & c_{h_{2,1}}^{(NE)} & c_{h_{2,2}}^{(NE)} \end{pmatrix} \tag{3.48}$$

$$[K_h]=\begin{pmatrix} k_{h_{1,1}}^{(1)} & k_{h_{1,2}}^{(1)} & 0 & \cdots & \cdots & \cdots & 0 \\ k_{h_{2,1}}^{(1)} & k_{h_{2,2}}^{(1)}+k_{h_{1,1}}^{(2)} & k_{h_{1,2}}^{(2)} & 0 & \cdots & \cdots & \vdots \\ 0 & k_{h_{2,1}}^{(2)} & k_{h_{2,2}}^{(2)}+k_{h_{1,1}}^{(3)} & k_{h_{1,2}}^{(3)} & 0 & \ddots & \vdots \\ \vdots & 0 & k_{h_{2,1}}^{(3)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & k_{h_{1,2}}^{(NE-1)} & 0 \\ \vdots & \vdots & \ddots & \ddots & k_{h_{2,1}}^{(NE-1)} & k_{h_{2,2}}^{(NE-1)}+k_{h_{1,1}}^{(NE)} & k_{h_{1,2}}^{(NE)} \\ 0 & \cdots & \cdots & \cdots & 0 & k_{h_{2,1}}^{(NE)} & k_{h_{2,2}}^{(NE)} \end{pmatrix} \tag{3.49}$$

$$[C_m]=\begin{pmatrix} c_{m_{1,1}}^{(1)} & c_{m_{1,2}}^{(1)} & 0 & \cdots & \cdots & \cdots & 0 \\ c_{m_{2,1}}^{(1)} & c_{m_{2,2}}^{(1)}+c_{m_{1,1}}^{(2)} & c_{m_{1,2}}^{(2)} & 0 & \cdots & \cdots & \vdots \\ 0 & c_{m_{2,1}}^{(2)} & c_{m_{2,2}}^{(2)}+c_{m_{1,1}}^{(3)} & c_{m_{1,2}}^{(3)} & 0 & \ddots & \vdots \\ \vdots & 0 & c_{m_{2,1}}^{(3)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & c_{m_{1,2}}^{(NE-1)} & 0 \\ \vdots & \vdots & \ddots & \ddots & c_{m_{2,1}}^{(NE-1)} & c_{m_{2,2}}^{(NE-1)}+c_{m_{1,1}}^{(NE)} & c_{m_{1,2}}^{(NE)} \\ 0 & \cdots & \cdots & \cdots & 0 & c_{m_{2,1}}^{(NE)} & c_{m_{2,2}}^{(NE)} \end{pmatrix} \tag{3.50}$$

$$[K_m]=\begin{pmatrix} k_{m_{1,1}}^{(1)} & k_{m_{1,2}}^{(1)} & 0 & \cdots & \cdots & \cdots & 0 \\ k_{m_{2,1}}^{(1)} & k_{m_{2,2}}^{(1)}+k_{m_{1,1}}^{(2)} & k_{m_{1,2}}^{(2)} & 0 & \cdots & \cdots & \vdots \\ 0 & k_{m_{2,1}}^{(2)} & k_{m_{2,2}}^{(2)}+k_{m_{1,1}}^{(3)} & k_{m_{1,2}}^{(3)} & 0 & \ddots & \vdots \\ \vdots & 0 & k_{m_{2,1}}^{(3)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & k_{m_{1,2}}^{(NE-1)} & 0 \\ \vdots & \vdots & \ddots & \ddots & k_{m_{2,1}}^{(NE-1)} & k_{m_{2,2}}^{(NE-1)}+k_{m_{1,1}}^{(NE)} & k_{m_{1,2}}^{(NE)} \\ 0 & \cdots & \cdots & \cdots & 0 & k_{m_{2,1}}^{(NE)} & k_{m_{2,2}}^{(NE)} \end{pmatrix} \tag{3.51}$$

$$\{f_h\}=\begin{pmatrix} \hat{Q}_1^{(1)} \\ \hat{Q}_2^{(1)}+\hat{Q}_1^{(2)} \\ \hat{Q}_2^{(2)}+\hat{Q}_1^{(3)} \\ \vdots \\ \vdots \\ \hat{Q}_2^{(NE-1)}+\hat{Q}_1^{(NE)} \\ \hat{Q}_2^{(NE)} \end{pmatrix} \tag{3.52}$$

$$\{f_m\}=\begin{pmatrix} \hat{Q}_3^{(1)} \\ \hat{Q}_4^{(1)}+\hat{Q}_3^{(2)} \\ \hat{Q}_4^{(2)}+\hat{Q}_3^{(3)} \\ \vdots \\ \vdots \\ \hat{Q}_4^{(NE-1)}+\hat{Q}_3^{(NE)} \\ \hat{Q}_4^{(NE)} \end{pmatrix} \tag{3.53}$$

BOUNDARY CONDITIONS

The next step in creating the finite element model is to evaluate the boundary conditions of Eqs. (3.52) and (3.53). It should be noted that all boundary conditions involve secondary variables. Boundary conditions associated with secondary variables

are often called natural boundary conditions and represent flux values. The secondary variables associated with a one dimensional linear finite element ($\hat{Q}_a$ and $\hat{Q}_b$) are depicted in Figure 6.

Since there are no sources or sinks within the plate providing heat or mass flux it follows that

$$\left.\begin{aligned}\hat{Q}_2^{(i)} + \hat{Q}_1^{(i+1)} &= 0\\ \hat{Q}_4^{(i)} + \hat{Q}_3^{(i+1)} &= 0\end{aligned}\right\} \qquad 1 \le i \le NE-1 \tag{3.54}$$

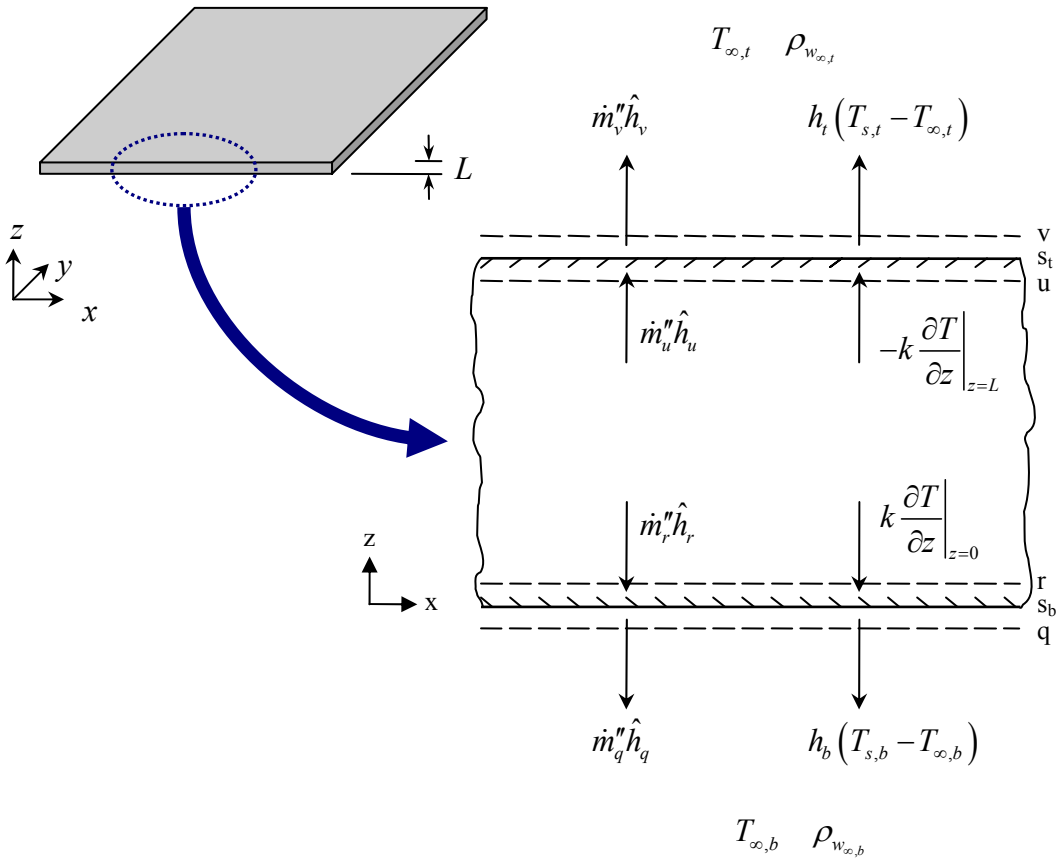

Figure 8. Energy flux through top and bottom surfaces of a flat plate.

As a result Eqs. (3.52) and (3.53) simplify to

$$\{f_h\} = \begin{pmatrix} \hat{Q}_1^{(1)} \\ 0 \\ \vdots \\ 0 \\ \hat{Q}_2^{(NE)} \end{pmatrix} \qquad \{f_m\} = \begin{pmatrix} \hat{Q}_3^{(1)} \\ 0 \\ \vdots \\ 0 \\ \hat{Q}_4^{(NE)} \end{pmatrix} \tag{3.55}$$

In order to evaluate the quantities $\hat{Q}_1^{(1)}$, $\hat{Q}_2^{(NE)}$, $\hat{Q}_3^{(1)}$ and $\hat{Q}_4^{(NE)}$ it is necessary to perform energy and mass flux balances on the top and bottom surfaces of the flat plate. Figure 8 depicts the energy flux through control volumes that surround the top and bottom surfaces of the flat plate. In this figure *q* and *r* denote the surfaces of the control volume surrounding the bottom surface, s_b, of the flat plate. Likewise, *u* and *v* signify the control volume surfaces that encompass the top surface, s_t, of the flat plate. The distance between *q* and *r* is taken to be small (i.e., nearly zero). This is also the case for surfaces *u* and *v*.

The quantities $T_{s,t}$ and $T_{s,b}$ are the temperature values at the top and bottom surfaces of the plate. Likewise $T_{\infty,t}$ and $T_{\infty,b}$ are free stream air temperatures above and below the flat plate. In addition $\dot{m}_q''$, $\dot{m}_r''$, $\dot{m}_u''$ and $\dot{m}_v''$ are scalar quantities that represent the mass fluxes through the *q*, *r*, *u* and *v* control surfaces respectively. The quantities h_t and h_b are the convection coefficients of air above and below the plate. The conservation of energy principle applied to the control volume encompassing the bottom surface of Figure 8 leads to the following equation [12]:

$$k\frac{\partial T}{\partial z}\bigg|_{z=0} + \dot{m}_r''\hat{h}_r = \dot{m}_q''\hat{h}_q + h_b\left(T_b - T_{\infty,b}\right) \tag{3.56}$$

It should be noted that Eq. (3.56) assumes there is no net storage of moisture in the control volume.

The mass flux $\dot{m}_r''$ may next be related to the *z* component of the mass flux of moisture $\mathbf{n}_w''$ by

$$\dot{m}_r'' = -n_w''{}_z\Big|_{z=0} = \frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\Bigg|_{z=0} \tag{3.57}$$

and

$$\dot{m}_q'' = h_{m,b}\left(\rho_{w_q} - \rho_{w_{\infty,b}}\right) \tag{3.58}$$

where $h_{m,b}$ is the convective mass transfer coefficient below the plate. The quantities ρ_{w_q} and $\rho_{w_{\infty,b}}$ are the partial densities of water vapor at the surface *q* and in the free stream below the plate. Utilizing the relationships developed above and setting $\hat{h}_r = c_{p,w}T_{s,b}$ allows Eq. (3.56) to be written as

$$\left[k\frac{\partial T}{\partial z} + c_{p,w}T_{s,b}\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right]\Bigg|_{z=0} = h_{m,b}\hat{h}_q\left(\rho_{w_q} - \rho_{w_{\infty,b}}\right) + h_{s,b}\left(T_b - T_{\infty,b}\right) \tag{3.59}$$

A comparison of Eq. (3.6) with Eq. (3.59) allows boundary term $\hat{Q}_1^{(1)}$ to be written as

$$\hat{Q}_1^{(1)} = -h_{m,b}\hat{h}_q\left(\rho_{w_q} - \rho_{w_{\infty,b}}\right) - h_{s,b}\left(T_1 - T_{\infty,b}\right) \tag{3.60}$$

$\hat{Q}_2^{(NE)}$ may next be evaluated using the same process that was employed to determine $\hat{Q}_1^{(1)}$. An energy balance through the control surfaces *u* and *v* results in the following equation

$$-k\left.\frac{\partial T}{\partial z}\right|_{z=L}+\dot{m}_u''\hat{h}_u=\dot{m}_v''\hat{h}_v+h_t\left(T_t-T_{\infty,t}\right) \tag{3.61}$$

The mass flux quantities are given by

$$\dot{m}_u''=n_w''{}_z\Big|_{z=L}=-\left.\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right|_{z=L} \tag{3.62}$$

$$\dot{m}_v''=h_{m,t}\left(\rho_{w_v}-\rho_{w_{\infty,t}}\right) \tag{3.63}$$

Setting $\hat{h}_u=c_{p,w}T_{s,t}$ and using the above relations allows (3.61) to be written as

$$\left[-k\frac{\partial T}{\partial z}-c_{p,w}T_{s,t}\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right]\Bigg|_{z=L}=h_{m,t}\hat{h}_v\left(\rho_{w_v}-\rho_{w_{\infty,t}}\right)+h_t\left(T_t-T_{\infty,t}\right) \tag{3.64}$$

Comparing Eq. (3.6) with Eq. (3.64) allows boundary term $\hat{Q}_2^{(NE)}$ to be written as

$$\hat{Q}_2^{(NE)}=-h_{m,t}\hat{h}_v\left(\rho_{w_v}-\rho_{w_{\infty,t}}\right)-h_t\left(T_{NN}-T_{\infty,t}\right) \tag{3.65}$$

To determine $\hat{Q}_3^{(1)}$ and $\hat{Q}_4^{(NE)}$ it is necessary to perform a balance of mass fluxes similar to the energy balance performed previously. Figure 9 depicts the mass flux through control volumes bounding the top and bottom surfaces of the flat plate. The notation of Figure 9 is identical to the notation used in Figure 8. Thus Figure 9 leads to the following equations describing the mass transfer from the top and bottom surfaces

$$\left[\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right]\Bigg|_{z=0}=h_{m,b}\left(\rho_{w_q}-\rho_{w_{\infty,b}}\right) \tag{3.66}$$

$$\left[-\frac{\rho D}{1-m_w}\frac{\partial m_w}{\partial z}\right]\Bigg|_{z=L}=h_{m,t}\left(\rho_{w_v}-\rho_{w_{\infty,t}}\right) \tag{3.67}$$

By comparing Eqs. (3.66) and (3.67) with Eq. (3.12) is follows that the boundary terms $\hat{Q}_3^{(1)}$ and $\hat{Q}_4^{(NE)}$ may be expressed as

$$\hat{Q}_3^{(1)} = -h_{m,b}\left(\rho_{w_q} - \rho_{w_{\infty,b}}\right) \tag{3.68}$$

$$\hat{Q}_4^{(NE)} = -h_{m,t}\left(\rho_{w_v} - \rho_{w_{\infty,t}}\right) \tag{3.69}$$

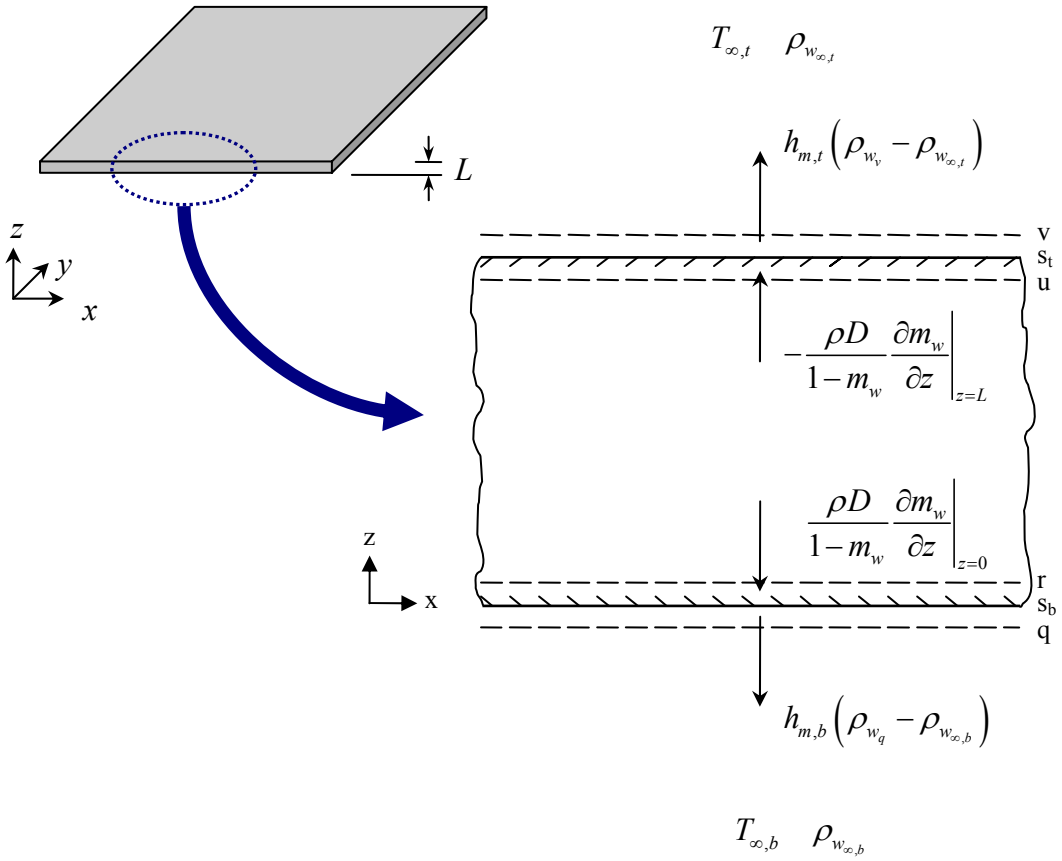

Figure 9. Mass flux through top and bottom surfaces of a flat plate.

Thus the column vectors $\{f_h\}$ and $\{f_m\}$ per Eq. (3.55) may be expressed as

$$\{f_h\} = \begin{pmatrix} -h_{m,b}\hat{h}_q\left(\rho_{w_q} - \rho_{w_{\infty,b}}\right) - h_b\left(T_1 - T_{\infty,b}\right) \\ 0 \\ \vdots \\ 0 \\ -h_{m,t}\hat{h}_v\left(\rho_{w_v} - \rho_{w_{\infty,t}}\right) - h_t\left(T_{NN} - T_{\infty,t}\right) \end{pmatrix} \tag{3.70}$$

$$\{f_m\} = \begin{pmatrix} -h_{m,b}\left(\rho_{w_q} - \rho_{w_{\infty,b}}\right) \\ 0 \\ \vdots \\ 0 \\ -h_{m,t}\left(\rho_{w_v} - \rho_{w_{\infty,t}}\right) \end{pmatrix} \tag{3.71}$$

It is necessary to address one final matter in regards to the developed boundary conditions. In Figures 8 and 9 the temperature is expected to behave as a continuous function. In other words there are no discontinuities in the temperature field and

$$T_r = T_q = T_{s,b} = T_1 \tag{3.72}$$

$$T_u = T_v = T_{s,t} = T_{NN} \tag{3.73}$$

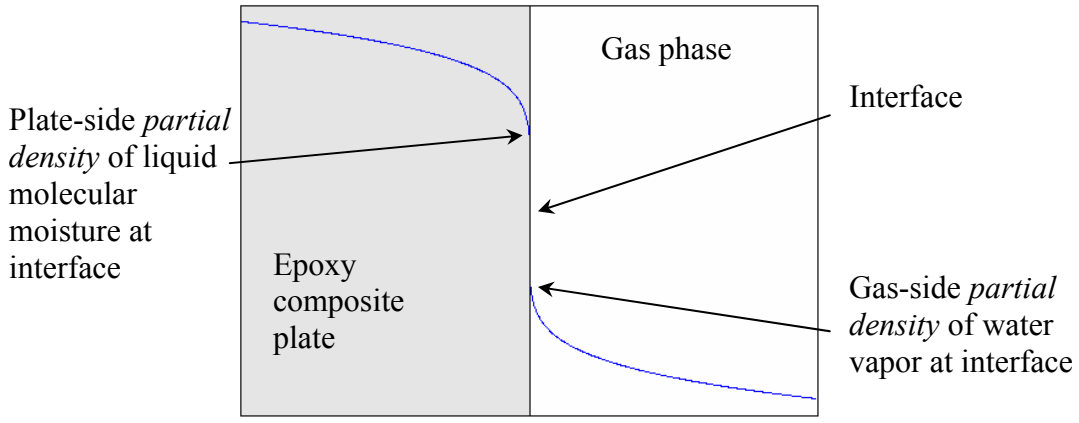

Figure 10. Partial density profile of water, ρ_w, in the vicinity of the plate surface [5].

Eqs. (3.72) and (3.73) simply assert that the temperature field is continuous between the surfaces of the flat plate and the atmosphere. Unlike temperature, however, partial densities are not necessarily continuous functions across gas-liquid or gas-solid interfaces [5,9]. Instead there exists a jump in the partial density of moisture at both interfaces of the flat plate. Figure 10 depicts a general plot of partial density in the vicinity of an interface of a flat plat. This figure is presented to illustrate the fact that a discontinuity exists and not as a means of determining ρ_w across the interface. Unlike the results of Eqs. (3.72) and (3.73) it should therefore be noted that

$$\rho_{w_r} \neq \rho_{w_q} \tag{3.74}$$

$$\rho_{w_u} \neq \rho_{w_v} \tag{3.75}$$

Since mass transfer is driven on the fluid side by evaporation it follows from thermodynamic phase equilibrium [2] that the partial densities ρ_{w_q} and ρ_{w_v} are related to the densities of saturated water vapor. These quantities are therefore approximated via Assumption 7 as

$$\rho_{w_q} = \rho_{sat}\left(T_1\right) m_w = \rho_{sat}\left(T_1\right) \frac{\rho_1}{\rho_e + \rho_1} \tag{3.76}$$

$$\rho_{w_v} = \rho_{sat}\left(T_{NN}\right) m_w = \rho_{sat}\left(T_{NN}\right) \frac{\rho_{NN}}{\rho_e + \rho_{NN}} \tag{3.77}$$

where ρ_{sat} is the saturated density of water vapor at a given temperature. This quantity may be expressed as a function of temperature via curve fitting the thermodynamic data. From Eqs. (3.76) and (3.77) it is evident that the saturation density of the water vapor is a function of the surface temperature of the plate.

The enthalpy values $\hat{h}_q$ and $\hat{h}_v$ of the superheated water vapor may next be written as

$$\hat{h}_q = c_{P,w}T_1 + h_{fg} + c_{p,s}\left(T_{\infty,b} - T_1\right) \tag{3.78}$$

$$\hat{h}_v = c_{P,w}T_{NN} + h_{fg} + c_{p,s}\left(T_{\infty,t} - T_{NN}\right) \tag{3.79}$$

where h_{fg} is the latent heat or heat of vaporization of water and $c_{p,s}$ is the specific heat of water vapor. Using the results of Eqs. (3.78) and (3.79) allows the boundary conditions to be written as

$$\{f_h\} = \begin{pmatrix} -h_{m,b}\left[c_{P,w}T_1 + h_{fg} + c_{p,s}\left(T_{\infty,b} - T_1\right)\right]\left[\rho_{sat}\left(T_1\right)\dfrac{\rho_1}{\rho_e + \rho_1} - \rho_{\infty,b}\right] - h_b\left(T_1 - T_{\infty,b}\right) \\ 0 \\ \vdots \\ 0 \\ -h_{m,t}\left[c_{P,w}T_{NN} + h_{fg} + c_{p,s}\left(T_{\infty,t} - T_{NN}\right)\right]\left[\rho_{sat}\left(T_{NN}\right)\dfrac{\rho_{NN}}{\rho_e + \rho_{NN}} - \rho_{\infty,t}\right] - h_t\left(T_{NN} - T_{\infty,t}\right) \end{pmatrix} \tag{3.80}$$

$$\{f_m\} = \begin{pmatrix} -h_{m,b}\left[\rho_{sat}\left(T_1\right)\dfrac{\rho_1}{\rho_e + \rho_1} - \rho_{\infty,b}\right] \\ 0 \\ \vdots \\ 0 \\ -h_{m,t}\left[\rho_{sat}\left(T_{NN}\right)\dfrac{\rho_{NN}}{\rho_e + \rho_{NN}} - \rho_{\infty,t}\right] \end{pmatrix} \tag{3.81}$$

It should be noted that Eqs. (3.80) and (3.81) contain the quantities T_1, T_{NN}, ρ_1 and ρ_{NN}. It is therefore necessary to rearrange the global matrix equations, Eqs. (3.46) and (3.47), such that the coefficients of T_1, T_{NN}, ρ_1 and ρ_{NN} per Eqs. (3.80) and (3.81) are

added into the global $[K_h]$ and $[K_m]$ matrices. Performing the necessary rearrangement results in

$$\{f_h\} = \begin{pmatrix} -h_{m,b}\left[h_{fg} + c_{p,s}T_{\infty,b}\right]\left[\rho_{sat}(T_1)\dfrac{\rho_1}{\rho_e+\rho_1} - \rho_{\infty,b}\right] + h_b T_{\infty,b} \\ 0 \\ \vdots \\ 0 \\ -h_{m,t}\left[h_{fg} + c_{p,s}T_{\infty,t}\right]\left[\rho_{sat}(T_{NN})\dfrac{\rho_{NN}}{\rho_e+\rho_{NN}} - \rho_{\infty,t}\right] + h_t T_{\infty,t} \end{pmatrix} \tag{3.82}$$

$$\{f_m\} = \begin{pmatrix} h_{m,b}\rho_{\infty,b} \\ 0 \\ \vdots \\ 0 \\ h_{m,t}\rho_{\infty,t} \end{pmatrix} \tag{3.83}$$

$$K_{h_{1,1}} = K_{h_{1,1}} + h_{m,b}\left(c_{P,w} - c_{p,s}\right)\left[\rho_{sat}(T_1)\frac{\rho_1}{\rho_e+\rho_1} - \rho_{\infty,b}\right] + h_b \tag{3.84}$$

$$K_{h_{NN,NN}} = K_{h_{NN,NN}} + h_{m,t}\left(c_{P,w} - c_{p,s}\right)\left[\rho_{sat}(T_{NN})\frac{\rho_{NN}}{\rho_e+\rho_{NN}} - \rho_{\infty,t}\right] + h_t \tag{3.85}$$

$$K_{m_{1,1}} = K_{m_{1,1}} + \frac{h_{m,b}\rho_{sat}(T_1)}{\rho_e+\rho_1} \tag{3.86}$$

$$K_{m_{NN,NN}} = K_{m_{NN,NN}} + \frac{h_{m,t}\rho_{sat}(T_{NN})}{\rho_e+\rho_{NN}} \tag{3.87}$$

where $K_{h_{1,1}}$ and $K_{h_{NN,NN}}$ are the new *1,1* and *NN,NN* elements of $[K_h]$; and $K_{m_{1,1}}$ and $K_{m_{NN,NN}}$ are the new *1,1* and *NN,NN* elements of $[K_m]$.

CONVECTIVE HEAT TRANSFER COEFFICIENTS

The convective heat transfer coefficients h_b and h_t may be determined by analytical methods or through empirical formulas. For laminar, turbulent and mixed flows over a flat plate, convective heat transfer coefficients are readily available in the literature [5,10,11]. For laminar parallel flow, h is derived from the Blasius solution of the Faulkner-Skan Equation [8]. For mixed and turbulent flows, empirical formulas are utilized [10,11]. To determine the value of h for parallel flows it is convenient to the define the following quantities [10,11]

$$Re_x \equiv \frac{U_\infty x}{\nu_a} \tag{3.88}$$

$$Pr \equiv \frac{\nu_a \rho_a c_{P,a}}{k_a} \tag{3.89}$$

where Re_x and Pr are the local Reynold and Prandtl numbers at location x along the flat plate. U_∞ is the free stream velocity of the air flowing over the plate. The quantities ν, ρ, c_P and k are kinematic viscosity, density, specific heat and thermal conductivity respectively. The subscript a conveys to the reader that the preceding properties are that of the air. As these properties are functions of temperature it is important that they be evaluated at the film temperature T_f which is approximated as [11]

$$T_f = \frac{1}{2}\left(T_s + T_\infty\right) \tag{3.90}$$

It is next useful to define the local Nusselt number as

$$Nu_x \equiv \frac{hx}{k_a} \tag{3.91}$$

The Reynold and Nusselt numbers may also be expressed as average values along the *x* direction of the plate as [11]

$$\overline{Re} \equiv \frac{U_\infty L_P}{\nu_a} \tag{3.92}$$

$$\overline{Nu} \equiv \frac{\bar{h} L_P}{k_a} \tag{3.93}$$

where L_P is the length of the flat plate as per Figure 3. The average and local Nusselt numbers may be determined analytically for laminar flow over a flat plate as [11]

$$Nu_x = 0.332 Re_x^{1/2} Pr^{1/3} \qquad Pr \geq 0.6 \tag{3.94}$$

$$\overline{Nu}_L = 0.664 Re_L^{1/2} Pr^{1/3} \qquad Pr \geq 0.6 \tag{3.95}$$

Since *Pr* for air is always greater than 0.7 in the present study, Eqs. (3.94) and (3.95) may be utilized for determining *h*. For turbulent flow the local Nusselt number is calculated from the following empirical equation [11]

$$Nu_x = 0.0296 Re_x^{4/5} Pr^{1/3} \qquad 0.6 \leq Pr \leq 60 \tag{3.96}$$

Transition between laminar and turbulent flows is assumed to occur when Re_x is greater than the critical Reynolds number $Re_{x,c}$. For mixed flows where the flow is part laminar and turbulent the following empirical relationship may be used to approximate the average Nusselt number [11]

$$\overline{Nu}_L = \left(0.037 Re_L^{4/5} - 871\right) Pr^{1/3} \qquad \begin{array}{l} 0.6 \leq Pr \leq 60 \\ Re_{x,c} \cong 5 \times 10^5 \end{array} \tag{3.97}$$

Values for h are also available in the literature for gas flows across a surface due to impinging jets. In the case of a single round nozzle discharging a fluid against a flat surface the following Reynolds and average Nusselt numbers are defined as [11]

$$\overline{Nu} = \frac{\bar{h} d_h}{k_a} \tag{3.98}$$

$$Re = \frac{V d_h}{\nu_a} \tag{3.99}$$

where d_h is the inner diameter of the nozzle from which the gas stream is flowing. The average Nusselt number $\overline{Nu}$ in the vicinity of the stagnation point of the flow against the flat plat is determined empirically by the formula [11]

$$\overline{Nu} = Pr^{0.42} G F_1 \tag{3.100}$$

where

$$F_1 = 2Re^{1/2}\left(1 + 0.005Re^{0.55}\right)^{1/2} \tag{3.101}$$

$$G = \frac{d_h}{r}\frac{1 - 1.1 d_h/r}{1 + 0.1\left(H/d_h - 6\right) d_h/r} \tag{3.102}$$

The above relations allow the average convection heat transfer coefficient to be expressed as

$$\bar{h} = \frac{2k_a Re^{1/2} Pr^{0.42}\left(1 - 1.1 d_h/r\right)\left(1 + 0.005Re^{0.55}\right)^{1/2}}{r\left(1 + 0.1\left(H/d_h - 6\right) d_h/r\right)} \tag{3.103}$$

Equation (3.103) may therefore be used to estimate the average convection heat transfer coefficient $\bar{h}$ in the vicinity of the stagnation point of the flow. The quantity r per Eq. (3.102) is the radial distance from the stagnation point to an arbitrary distance on the

surface of the plate over which $\bar{h}$ is averaged. The quantity H is the distance from the exit plane of the nozzle to the surface of the flat plate. It is important to note that Eq. (3.100) is only valid in the following ranges [11]

$$\begin{aligned} &2,000 \le Re \le 400,000 \\ &2 \le H/d_h \le 12 \\ &2.5 \le r/d_h \le 7.5 \end{aligned} \tag{3.104}$$

It is also necessary to present solutions for the average convective heat transfer coefficient, $\bar{h}$, for free convection of a vertical flat plate. Free convection occurs from buoyancy driven flow arising from temperature differences. For free convection the Nusselt number is the same as for parallel flows as was presented in Eq. (3.93). The Rayleigh number, Ra_L, is next defined as [11]

$$Ra_L = \frac{g\rho_a c_{P,a} \left| T_s - T_\infty \right| L_P^{\,3}}{\nu_a k_a T_f} \tag{3.105}$$

where g is the gravitational constant. The Nusselt number may next be calculated as [11]

$$Nu_L = 0.68 + \frac{0.670 Ra_L^{\,1/4}}{\left[1 + \left(0.492/Pr\right)^{9/16}\right]^{4/9}} \qquad Ra_L < 10^9 \tag{3.106}$$

The average convective heat transfer coefficient, $\bar{h}$, for free convection may therefore be determined by Eqs. (3.105), (3.106) and (3.93).

CONVECTIVE MASS TRANSFER COEFFICIENTS

It is next necessary to determine the convective mass transfer coefficients $h_{m,b}$ and $h_{m,t}$. To do so it is important to introduce some additional non–dimensional quantities. The Schmidt number *Sc* is defined as [11]

$$Sc \equiv \frac{\nu_a}{D_a} \tag{3.107}$$

For parallel flow over a flat plate local and average Sherwood numbers are defined as [11]

$$Sh_x \equiv \frac{h_m x}{D_a} \tag{3.108}$$

$$\overline{Sh} \equiv \frac{\overline{h_m} L_P}{D_a} \tag{3.109}$$

where the quantity D_a is the diffusivity of water vapor in air. In the case of a single impinging jet, the average Schmidt number is expressed as [11]

$$\overline{Sh} \equiv \frac{\overline{h_m} d_h}{D_a} \tag{3.110}$$

It has been observed that there are strong analogies between heat and mass transfer. The governing equations and boundary conditions are often of a similar nature. As a result it has also been shown that there often exists a similarity between the convective transfer coefficients. This correspondence has been experimentally verified and is of the form [5,10,11]

$$\frac{Nu}{Pr^n} = \frac{Sh}{Sc^n} \tag{3.111}$$

where the terms of Eqs. (3.111) may be local or average quantities. Thus if one convective transfer coefficient is known, the other may be determined from Eq. (3.111).

It is important to note that the above analogy only holds for low rates of mass transfer [5,10]. This is because the transfer coefficients are dependent upon the flow field across the plate. One of the boundary conditions of the Blasius solution is that the fluid flow orthogonal to the plate (z direction) is zero at the plate surface. In other words, if evaporation occurs at a rate that significantly affects the flow field over the plate, the Blasius solution is no longer valid. It also follows that the empirical formulas for turbulent parallel flow as well as for flows from impinging jets will also be invalid under these conditions. It is therefore assumed that mass transfer from the plate imparts a negligible variation in the velocity field of the air.

Inserting Eqs. (3.91) and (3.108) into Eq. (3.111) results in

$$\frac{h}{k_a Pr^n} = \frac{h_m}{D_a Sc^n} \tag{3.112}$$

which may be rewritten as

$$\frac{h}{h_m} = \frac{k_a}{D_a Le^n} = \rho c_P Le^{1-n} \tag{3.113}$$

where the Lewis Number, Le, is defined as [11]

$$Le \equiv \frac{Sc}{Pr} = \frac{\alpha_a}{D_a} \tag{3.114}$$

and the thermal diffusivity, α_a, is

$$\alpha_a = \frac{k_a}{\rho_a c_{P,a}} \tag{3.115}$$

The convective mass transfer coefficient is therefore determined to be of the form

$$h_m = \frac{h}{\rho c_P \left(\frac{k}{\rho c_P D} \right)^{1-n}} \tag{3.116}$$

For parallel flow over a flat plate it has been shown that $n = {}^{1}\!/_{3}$ [11]. For the case of flow from impinging jets it has been suggested that $n=0.42$ [11]. Therefore, once the convective heat transfer coefficient, h, has been calculated, h_m may then be determined for parallel flow as

$$h_m = \frac{h}{\rho c_P \left(\frac{k}{\rho c_P D} \right)^{2/3}} \tag{3.117}$$

For the case of an impinging jet the convective mass transfer coefficient is

$$h_m = \frac{h}{\rho c_P \left(\frac{k}{\rho c_P D} \right)^{0.58}} \tag{3.118}$$

It should be noted that all material properties in Eqs. (3.117) and (3.118) are properties of air and are evaluated at the film temperature T_f.

MATERIAL PROPERTIES

Many of the material properties and coefficients introduced in this work are not constant. In fact some of these quantities vary significantly with variations in temperature and/or pressure. In the present study it is convenient to represent these quantities as functions of temperature only (i.e., isobaric conditions).

The following equations have been generated via polynomial curve fits of the published data [11] for properties of air at atmospheric pressure.

$$\rho_a = -7.9282\left(10^{-14}\right)T_f^{\,5} + 9.5330\left(10^{-11}\right)T_f^{\,4} - 5.0164\left(10^{-8}\right)T_f^{\,3} + 1.68472\left(10^{-5}\right)T_f^{\,2} - 0.0047T_f + 1.2770 \tag{3.119}$$

$$c_{P,a} = -8.8889\left(10^{-14}\right)T_f^{\,6} + 9.9528\left(10^{-11}\right)T_f^{\,5} - 4.2892\left(10^{-8}\right)T_f^{\,4} + 8.2319\left(10^{-6}\right)T_f^{\,3} - 0.0002T_f^{\,2} + 0.0153T_f + 1006.5729 \tag{3.120}$$

$$\nu_a = -4.1026\left(10^{-19}\right)T_f^{\,5} + 3.1179\left(10^{-16}\right)T_f^{\,4} - 1.3269\left(10^{-13}\right)T_f^{\,3} + 1.2036\left(10^{-10}\right)T_f^{\,2} + 8.8727\left(10^{-8}\right)T_f + 1.3414\left(10^{-5}\right) \tag{3.121}$$

$$k_a = 1.8461\left(10^{-15}\right)\mathrm{T_f}^{5} - 1.4637\left(10^{-12}\right)\mathrm{T_f}^{4} + 3.6606\left(10^{-10}\right)\mathrm{T_f}^{3} - 5.7872\left(10^{-8}\right)\mathrm{T_f}^{2} + 7.93825\left(10^{-5}\right)\mathrm{T_f} + 0.0242 \tag{3.122}$$

$$Pr_a = 1.3778\left(10^{-15}\right)T_f^{\,6} - 1.2822\left(10^{-12}\right)T_f^{\,5} + 4.3350\left(10^{-10}\right)T_f^{\,4} - 6.31081\left(10^{-8}\right)T_f^{\,3} + 3.8276\left(10^{-6}\right)T_f^{\,2} - 2.3383\left(10^{-4}\right)T_f + 0.7117 \tag{3.123}$$

It should be noted that Eqs. (3.119) through (3.123) are valid between -23°C and 300°C. Furthermore, the film temperature T_f in these equations must be in units of Celsius (not Kelvin). The diffusion coefficient of water vapor in air may also be calculated from the literature [16] as

$$D_a = 1.87\left(10^{-10}\right)\left(T_f + 273\right)^{2.072} \tag{3.124}$$

The latent heat of vaporization may be determined by the following polynomial fit of the published data [2]

$$h_{fg} = 2.750\left(10^{-10}\right)T_f^{\,6} - 2.9275\left(10^{-7}\right)T_f^{\,5} + 7.67128\left(10^{-5}\right)T_f^{\,4} - 0.0211T_f^{\,3} + 0.9009T_f^{\,2} - 2372.6057T_f + 2501365.4834 \tag{3.125}$$

The thermal conductivity and specific heat of each element are calculated for each element of the finite element model as

$$k^e = \frac{\rho_e k_e + \rho_w{}^e k_w}{\rho^e} \tag{3.126}$$

$$c_P{}^e = \frac{\rho_e c_{P,e} + \rho_w{}^e c_{P,w}}{\rho^e} \tag{3.127}$$

TIME STEP APPROXIMATIONS

It has been previously established that the matrix form of the energy and mass transfer equations of the finite element models are transient, nonlinear and coupled. These characteristics present distinct challenges to the solution process. These challenges, however, can be overcome and the equations solved. This is accomplished by first addressing the transient nature of Eqs. (3.46) and (3.47). An examination of these equations reveals that the finite element models of the energy and mass transfer equations are of the form

$$[C]\{\dot{u}\} + [K]\{u\} = \{f\} \tag{3.128}$$

with initial conditions

$$\{u\}\big|_{t=0} = \{u_0\} \tag{3.129}$$

The column vector $\{u_0\}$ per Eq. (3.129) represents the known dependent variables at the nodes at *t=0* s. These dependent variables are the initial temperatures and the partial densities of molecular water.

To solve Eq. (3.128) with the above initial conditions it is convenient to introduce a time approximation scheme. The α–family of time approximation [14,15] allows time derivatives of dependent variables $\{\dot{u}\}$ in a set of matrix equations to be approximated between two successive time steps as

$$\frac{\{u\}_{s+1}-\{u\}_s}{\Delta t_{s+1}}=(1-\alpha)\{\dot{u}\}_s+\alpha\{\dot{u}\}_{s+1} \tag{3.130}$$

where $0\leq\alpha\leq1$. Quantities per Eq. (3.130) containing a subscript s are to be evaluated a time $t=t_s$. It should also be noted that Δt_{s+1} is the s^{th} time step and

$$\Delta t_{s+1}=t_{s+1}-t_s \tag{3.131}$$

where t_{s+1} and t_s are the time values at time step (s+1) and s respectively. It is next necessary to consider Eq. (3.128) at two successive time steps [14]

$$[C]_s\{\dot{u}\}_s=\{f\}_s-[K]_s\{u\}_s \tag{3.132}$$

$$[C]_{s+1}\{\dot{u}\}_{s+1}=\{f\}_{s+1}-[K]_{s+1}\{u\}_{s+1} \tag{3.133}$$

It is important to note that Eqs. (3.132) and (3.133) allow for variations in $[C]$, $[K]$ and $\{f\}$ between time steps. Solving Eq. (3.133) for $\{\dot{u}\}_{s+1}$ results in

$$\{\dot{u}\}_{s+1}=[C]_{s+1}^{-1}\left(\{f\}_{s+1}-[K]_{s+1}\{u\}_{s+1}\right) \tag{3.134}$$

It is next necessary for Eq. (3.130) to be multiplied by $[C]_s$. Performing this operation and rearranging terms results in

$$\Delta t_{s+1}(1-\alpha)[C]_s\{\dot{u}\}_s+\Delta t_{s+1}\alpha[C]_s\{\dot{u}\}_{s+1}=[C]_s\left(\{u\}_{s+1}-\{u\}_s\right) \tag{3.135}$$

The ultimate objective of Eq. (3.135) is to be able to eliminate the quantities $\{\dot{u}\}_s$ and $\{\dot{u}\}_{s+1}$. This is achieved through the use of Eqs. (3.132) and (3.134). Substituting the right hand sides of the equations for $[C]_s\{\dot{u}\}_s$ and $\{\dot{u}\}_{s+1}$ into Eq. (3.135) results in

$$\begin{aligned}&\Delta t_{s+1}\left[(1-\alpha)\left(\{f\}_s-[K]_s\{u\}_s\right)+\alpha[C]_s[C]_{s+1}^{-1}\left(\{f\}_{s+1}-[K]_{s+1}\{u\}_{s+1}\right)\right]\\&=[C]_s\left[\{u\}_{s+1}-\{u\}_s\right]\end{aligned} \tag{3.136}$$

Performing the necessary linear algebra steps allows Eq. (3.136) to be rewritten in a more concise and convenient form as

$$\begin{aligned}&\left([C]_s+\Delta t_{s+1}\alpha[C]_s[C]_{s+1}^{-1}[K]_{s+1}\right)\{u\}_{s+1}=\left[[C]_s-\Delta t_{s+1}(1-\alpha)[K]_s\right]\{u\}_s\\&+\Delta t_{s+1}\left[(1-\alpha)\{f\}_s+\alpha[C]_s[C]_{s+1}^{-1}\{f\}_{s+1}\right]\end{aligned} \tag{3.137}$$

Eq. (3.137) represents the time approximation of Eq. (3.128) using the α-family of time approximation. It is a very general equation as no restrictions were placed upon the elements of $[C]$, $[K]$, and $\{f\}$ and thus they may vary between time steps.

It should be noted that solving Eq. (3.137) can prove to be an onerous process for large systems of equations as it requires the inversion of several large matrices. There are some additional conditions that may lead to a simplification of this set of matrix equations. In the event that

$$[C]=[C]_s=[C]_{s+1} \tag{3.138}$$

it should be intuitive that

$$[I]=[C]_s[C]_{s+1}^{-1} \tag{3.139}$$

where $[\,I\,]$ is the *NN*×*NN* identity matrix. Under these conditions Eq. (3.137) simplifies to [15]

$$\begin{aligned}\left(\left[C\right]+\Delta t_{s+1}\alpha\left[K\right]_{s+1}\right)\left\{u\right\}_{s+1} &= \left[\left[C\right]-\Delta t_{s+1}\left(1-\alpha\right)\left[K\right]_{s}\right]\left\{u\right\}_{s} \\ &\quad +\Delta t_{s+1}\left[\left(1-\alpha\right)\left\{f\right\}_{s}+\alpha\left\{f\right\}_{s+1}\right]\end{aligned} \tag{3.140}$$

whose solution only requires the inversion of $\left(\left[C\right]+\Delta t_{s+1}\alpha\left[K\right]_{s+1}\right)$. In the event that $[C]$ is not constant between time steps, Eq. (3.140) may be generalized by defining $[C]$ through the α-family of approximation as

$$\left[C\right]\cong\left(1-\alpha\right)\left[C\right]_{s}+\alpha\left[C\right]_{s+1} \tag{3.141}$$

Inserting Eq. (3.141) into Eq. (3.140) results in

$$\begin{aligned}&\left[\left(1-\alpha\right)\left[C\right]_{s}+\alpha\left[C\right]_{s+1}+\Delta t_{s+1}\alpha\left[K\right]_{s+1}\right]\left\{u\right\}_{s+1}= \\ &\qquad\left[\left(1-\alpha\right)\left[C\right]_{s}+\alpha\left[C\right]_{s+1}-\Delta t_{s+1}\left(1-\alpha\right)\left[K\right]_{s}\right]\left\{u\right\}_{s} \\ &\qquad+\Delta t_{s+1}\left[\left(1-\alpha\right)\left\{f\right\}_{s}+\alpha\left\{f\right\}_{s+1}\right]\end{aligned} \tag{3.142}$$

Although Eq. (3.137) is sufficient for performing approximate time step solutions of Eq. (3.128), it is more convenient and time efficient to use Eq. (3.140) for the mass transport equation and Eq. (3.142) for the energy equation. Thus it is these equations that are utilized for the finite element model. Also, the *Crank-Nicolson* numerical integration scheme is utilized by letting α=½ [15] and the energy and mass transfer equations are written explicitly as

$$\begin{aligned}&\left[\left(1-\alpha\right)\left[C_{h}\right]_{s}+\alpha\left[C_{h}\right]_{s+1}+\Delta t_{s+1}\alpha\left[K_{h}\right]_{s+1}\right]\left\{T\right\}_{s+1}= \\ &\qquad\left[\left(1-\alpha\right)\left[C_{h}\right]_{s}+\alpha\left[C_{h}\right]_{s+1}-\Delta t_{s+1}\left(1-\alpha\right)\left[K_{h}\right]_{s}\right]\left\{T\right\}_{s} \\ &\qquad+\Delta t_{s+1}\left[\left(1-\alpha\right)\left\{f_{h}\right\}_{s}+\alpha\left\{f_{h}\right\}_{s+1}\right]\end{aligned} \tag{3.143}$$

$$\left(\left[C_m\right]+\Delta t_{s+1}\alpha\left[K_m\right]_{s+1}\right)\left\{\rho_w\right\}_{s+1}=\left[\left[C_m\right]-\Delta t_{s+1}\left(1-\alpha\right)\left[K_m\right]_s\right]\left\{\rho_w\right\}_s + \Delta t_{s+1}\left[\left(1-\alpha\right)\left\{f_m\right\}_s+\alpha\left\{f_m\right\}_{s+1}\right] \tag{3.144}$$

ALGORITHM FOR SOLVING THE NONLINEAR EQUATIONS

It has been shown that the transient nature of the energy and mass transfer equations necessitates that they be solved approximately at successive time steps via Eqs. (3.143) and (3.144). As these equations are nonlinear and coupled, they cannot be solved by conventional Gaussian elimination. Instead the solution procedure necessitates the use of iteration techniques in order to calculate $\left\{T\right\}_{s+1}$ and $\left\{\rho_w\right\}_{s+1}$ [14]. The magnitude of the task of solving Eqs. (3.143) and (3.144) is such that it is impracticable without the aid of the digital computer. It is therefore necessary to create an algorithm that may be implemented in the form of a computer program that allows for the solving of these nonlinear coupled equations. The necessary algorithm is developed and summarized in this section. Appendix B contains a computer program created in Matlab that utilizes the algorithm to solve Eqs. (3.143) and (3.144).

In order to create a general computer program for solving Eqs. (3.143) and (3.144) it is first necessary to specify the parameters of the system. These parameters include the initial conditions (i.e., partial density of water, $\left\{\rho_w\right\}_0$, and temperature, $\left\{T\right\}_0$, throughout the plate at *t=0*), the physical dimensions of the plate and the material properties of the system (i.e., plate and free steam air properties).

In addition it is necessary to specify the conditions of the finite element model, namely the number of elements, the simulation time, and the time step to be utilized. As the solution procedure is iterative it is also crucial to specify an *error* criteria. The purpose of this criteria will soon become apparent. For a complete list of parameters needed refer to Appendices A and B.

With the above parameters specified it is possible to create the individual nodes and elements of the model. Since the steepest gradients (i.e., molecular moisture concentration and temperature gradients) are expected near the boundaries of the plate it is not appropriate for the elements to be of equal lengths in the FE model. The following loop is therefore convenient, as it automates the process of creating the nodes of the FE model.

$$\begin{aligned} &z = zeros\left(NN,1\right);\\ &\text{for}\quad i = 1:1:NN\\ &\qquad z\left(i,1\right) = \frac{L}{2}\left\{1-\cos\left[\frac{\pi\left(i-1\right)}{NN-1}\right]\right\};\\ &\text{end} \end{aligned} \tag{3.145}$$

The first line of Eq. (3.145) zeros out an $NN\times 1$ column vector z whose elements will correspond to nodal positions. The *for loop* that follows assigns a spatial coordinate to each node. It should be noted that Eq. (3.145) assigns spatial coordinates such that the spacing between nodes is not constant. Instead the nodes near the plate boundaries are much closer together than are the nodes near the center of the plate. With the nodal positions specified, the element lengths and connectivity may also be formed via simple

additional *for loops*. If the number of nodes in the model were specified as 20, then the non-dimensional nodal coordinates would be given by Figure 11.

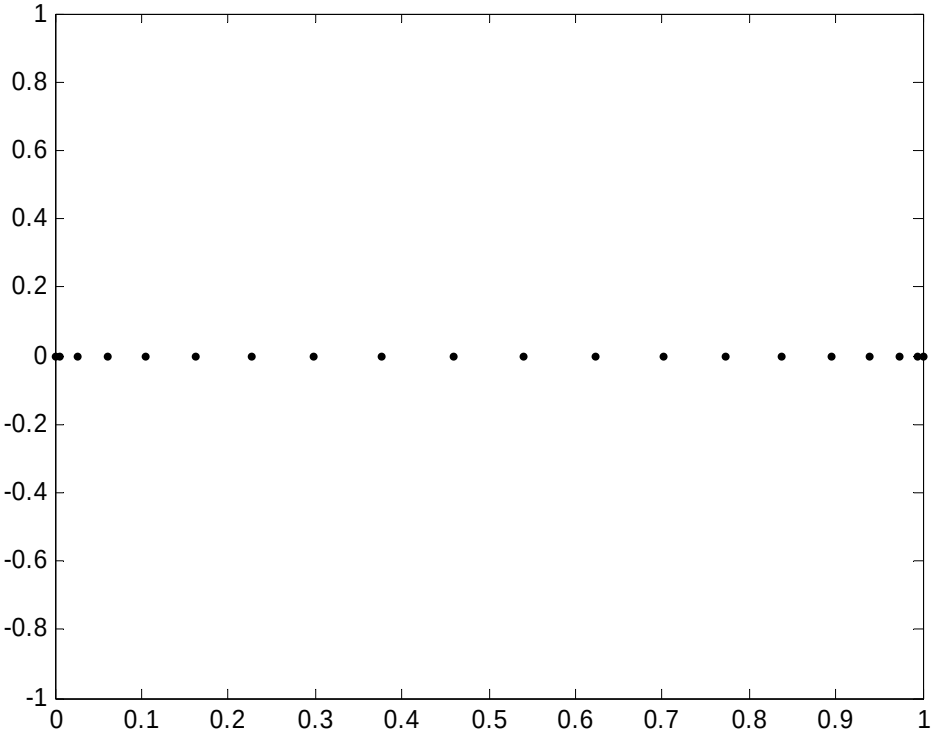

Figure 11. Non-dimensional nodal plot (*NN=20*).

Once the nodes, elements and connectivity of the elements have been well defined it is possible to begin the iterative solution procedure. The initial conditions at t=0 s are utilized to enter the first time step by requiring that

$$\{T\}_{s+1} = \{T\}_0 \tag{3.146}$$

$$\{\rho_w\}_{s+1} = \{\rho_w\}_0 \tag{3.147}$$

Utilizing Eqs. (3.146) and (3.147) it is possible to proceed to the first time step $t=t_s$. In this case $t_s=0$. Upon entering this time step, the computer creates the following column vectors

$$\{T\}_s = \{T\}_{s+1} \tag{3.148}$$

$$\{\rho_w\}_s = \{\rho_w\}_{s+1} \tag{3.149}$$

where the terms $\{T\}_s$ and $\{\rho_w\}_s$ are known from the initial conditions (i.e., not guessed). With Eqs. (3.148) and (3.149) it is possible to calculate the convective heat and mass transfer coefficients, the thermodynamic properties of air at the film temperatures, and the material properties. The local matrices $\left[C_h{}^e\right]_s$, $\left[K_h{}^e\right]_s$, $\left[C_m{}^e\right]_s$, and $\left[K_m{}^e\right]_s$ may next be formed. These may then be assembled to form the global matrices $[C_h]_s$, $[K_h]_s$, $[C_m]_s$, $[K_m]_s$ as well as the forcing vectors $\{f_h\}_s$ and $\{f_m\}_s$. It is necessary to next assume values of $\{T\}_{s+1}$ and $\{\rho_w\}_{s+1}$ before advancing into the next time step. It is convenient (but not necessary) to guess that these new vectors will be

$$\{T\}_{s+1} = 1.01\{T\}_s \tag{3.150}$$

$$\{\rho_w\}_{s+1} = 1.01\{\rho_w\}_s \tag{3.151}$$

It is now possible to begin the next time step $t=t_{s+1}$ where $t_{s+1}=t_s+\Delta t_{s+1}$. The quantities of $\{T\}_{s+1}$ and $\{\rho_w\}_{s+1}$ per Eqs. (3.150) and (3.151) are utilized to calculate the convective heat and mass transfer coefficients, the thermodynamic properties at the film temperatures, and the material properties. These values are then used to form the local

matrices $\left[C_h^{\,e}\right]_{s+1}$, $\left[K_h^{\,e}\right]_{s+1}$, $\left[C_m^{\,e}\right]_{s+1}$, and $\left[K_m^{\,e}\right]_{s+1}$. These local element matrices are then assembled into the global matrices and vectors $\left[C_h\right]_{s+1}$, $\left[K_h\right]_{s+1}$, $\left\{f_h\right\}_{s+1}$, $\left[C_m\right]_{s+1}$, $\left[K_m\right]_{s+1}$ and $\left\{f_m\right\}_{s+1}$.

With the global matrices defined and assembled at $t=t_s$ and $t=t_{s+1}$ it is next possible to solve Eqs. (3.143) and (3.144) for the quantities $\left\{T\right\}_{s+1}$ and $\left\{\rho_w\right\}_{s+1}$. Since it was necessary to know these quantities *a priori* in order to form the $t=t_{s+1}$ matrices, it is not expected that the results obtained from solving Eqs. (3.143) and (3.144) will be the true solution of these equations. Thus iteration is necessary.

The iterative process proceeds in the following manner. The *error* between the calculated and assumed values of $\left\{T\right\}_{s+1}$ and $\left\{\rho\right\}_{s+1}$ is compared to an allowable error, ε, between iterations. This *error* is defined mathematically as [14]

$$error = \sqrt{\frac{\sum_{i=1}^{NN}\left(u_i^{(r)} - u_i^{(r-1)}\right)^2}{\sum_{i=1}^{NN}\left(u_i^{(r)}\right)^2}} \tag{3.152}$$

where r refers to the iteration number and u is the dependent variable (i.e., temperature or partial density of molecular water). Thus Eq. (3.152) must be evaluated for the temperature and the partial density of molecular water. In the event that $error>\varepsilon$, it is necessary to perform another iteration by beginning time step $t=t_{s+1}$ anew and utilizing the most recently solved quantities $\left\{T\right\}_{s+1}$ and $\left\{\rho_w\right\}_{s+1}$ as the new guess values. This process is repeated until either both *error* values are less than ε or the number of iterations exceeds a limiting number. Once convergence has been reached (i.e., $error<\varepsilon$)

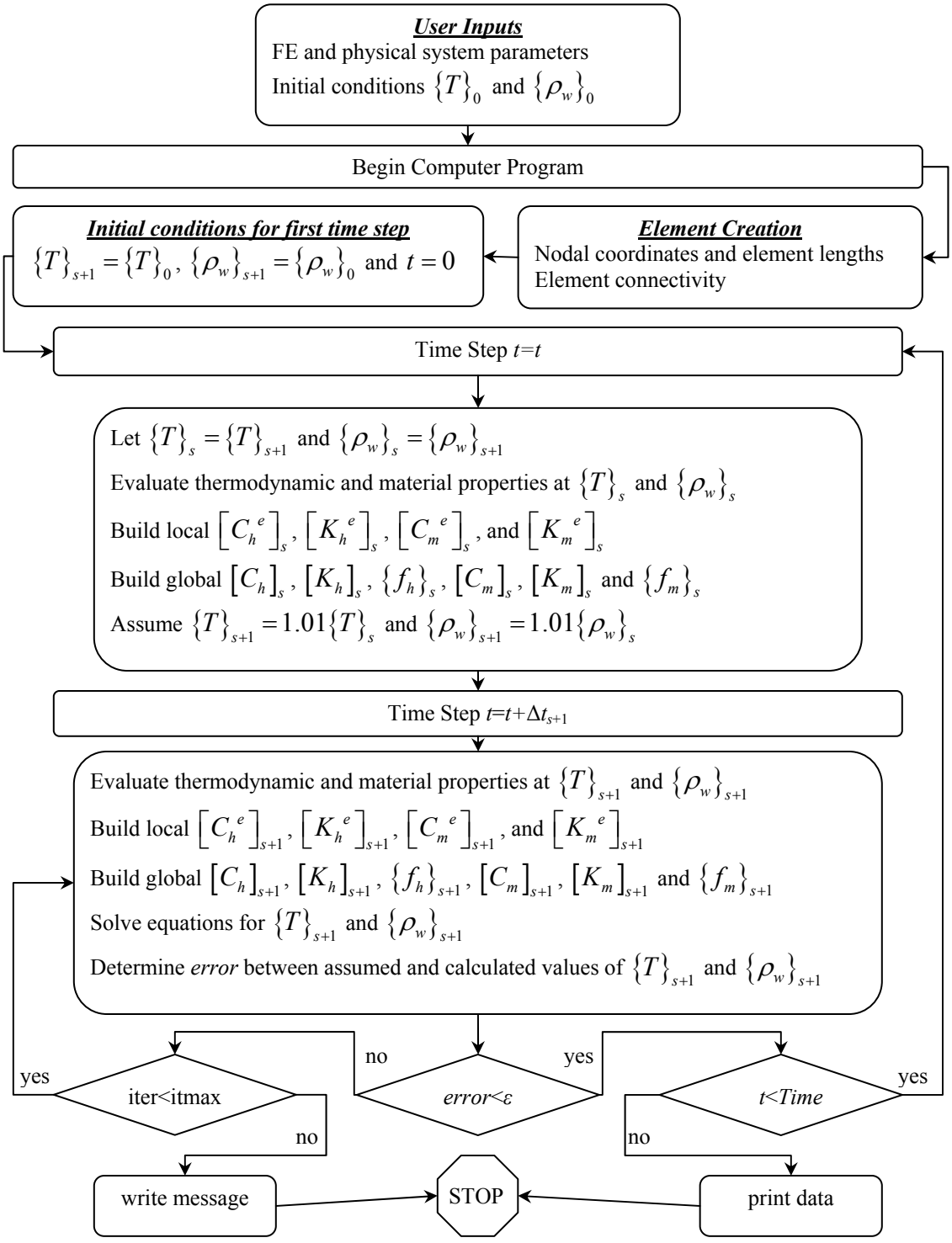

Figure 12. Flow chart for FE model of nonlinear coupled heat and mass transfer analysis.

it is possible to progress to the next time step by returning to $t=t_s$. For more complete details on the time step process used to solve Eqs. (3.143) and (3.144) refer to Figure 12 and Appendix B.

CHAPTER IV

RESULTS AND DISCUSSION

In this chapter the algorithm presented in Chapter III is utilized to determine surface temperature profiles vs. time of a flat plate for various initial conditions, and diffusion coefficients. To assess the effects of surface moisture evaporation, the surface temperature profiles of moist plates are compared with the surface temperature profiles of dry plates. The programs utilized for this analysis are contained in Appendices A and B; where Appendix A performs dry plate analysis and Appendix B performs analyses associated with moisture bearing plates (i.e., coupled energy and mass transfer).

Since it is desirable to compare numerical results with values obtained through experiment, only results for impinging flow are presented in this chapter. Eq. (3.118) is therefore used to determine the convective mass transfer coefficients in the analysis.

SYSTEM PARAMETERS PER NUMERICAL RESULTS

The initial and boundary conditions for the analyses and results of this chapter are as follows:

1. The plate possesses an initial temperature of 25°C.
2. The plate possesses an initial molecular moisture content at t=0 s. The results in this chapter are for initial moisture contents of 0%, 5%, 10%, 15%, 20%, 25% and 30%.

3. The plate is oriented vertically and at the surface z=0 the plate is exposed to stagnant air. Heat and mass transfer from this surface occur due to buoyancy effects (i.e., free convection).
4. At the surface z=L the plate is exposed to a single stream of air flowing orthogonal to the plate from a round impinging nozzle.

The results of this chapter rely in part on diffusion coefficient data contained in the literature for moisture diffusion in epoxy composite plate structures. Results are also generated for arbitrary diffusion coefficients. In the literature it has been shown that the diffusion coefficient for molecular water in epoxy composites is temperature dependent and obeys the following relationship [17,18]

$$D = D_0 e^{-E/k_b(T+273)} \tag{4.1}$$

A curve fit of the data collected by Browning [17] for diffusion coefficients produces values for the constants per Eq. (4.1). These are tabulated in Table 1.

Table 1. Curve fit of diffusion coefficient data [17].

Material Properties		
E/k_b	9,177.9	[K]
D_0	0.5269	[m^2/s]

The figures contained in this chapter are generated using the E/k_b value per Table 1 [17]. To allow for the comparison of different diffusion coefficient functions, however, different D_0 values are utilized to assess the effect of the plate's diffusion coefficient on the surface temperature at z=L of a flat plate. The values utilized for D_0

are contained in Table 2. Plots of the diffusion coefficient vs. temperature are also provided via Figure 13.

Table 2. Diffusion constants utilized for numerical results.

Diffusion constants		
D_0	0.5269	[m^2/s]
D_0	52.69	[m^2/s]
D_0	5,269	[m^2/s]

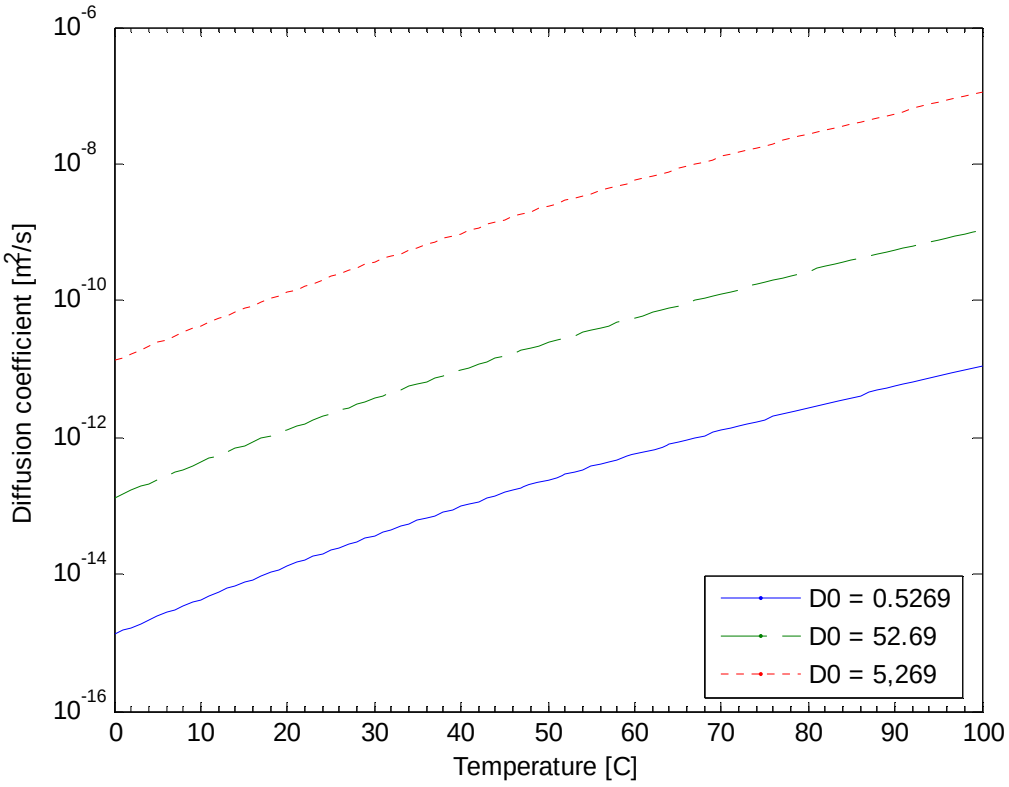

Figure 13. Diffusion coefficients vs. temperature.

Table 3 contains the parameters used to create the finite element models for the analysis. All results of this chapter are generated using the values contained in this table as they insure convergence of the numerical results. Since the surface temperature vs. time plots are nearly linear on a logarithmic time scale, a constant logarithmic time step is employed. As a result, the $(s+1)^{th}$ time step is calculated as

$$\Delta t_{s+1} = \left(t_F + 1\right)^{(s+1)/STEPS} - \left(t_F + 1\right)^{s/STEPS} \tag{4.2}$$

Table 3. Parameters associated with the finite element models.

Finite Element Parameters		
Number of Elements (*NE*)	100	
Number of Nodes (*NN*)	101	
Error Criteria (ε)	10^{-6}	[kg/m^3 or °C]
Simulation Time (t_F)	900	[s]
Number of Time Steps (*STEPS*)	10,000	

Table 4. Dimensions and properties of plate and nozzle.

Dimensions / Properties		
Plate Thickness (L)	1.0	[cm]
Distance from Nozzle to Plate (H)	5.1	[cm]
Radial Distance for Averaging (r)	6.0	[cm]
Inner Nozzle Diameter (d_h)	0.95	[cm]
Density (ρ_e)	1,130	[kg/m^3]
Specific heat ($c_{P,e}$)	1,000	[J/kg°C]
Thermal conductivity (k_e)	0.16	[W/m°C]

The material properties, plate dimensions and impinging nozzle dimensions used in the analysis are listed in Table 4. These values are constant for all the analysis presented in this chapter. The thermodynamic properties of liquid water used in the analysis are contained in Table 5. These values are also constant in the analysis and results.

Table 5. Properties of liquid water and water vapor [11].

Dimensions / Properties		
Specific heat – liquid ($c_{P,w}$)	4,181	[J/kg°C]
Specific heat – vapor ($c_{P,s}$)	1,900	[J/kg°C]
Thermal conductivity – liquid (k_w)	0.606	[W/m°C]

In addition to the quantities contained in the preceding tables, the velocity of the air leaving the nozzle is maintained at 9.94 m/s in the analysis. This produces a convective heat transfer coefficient h approximately equal to 60 W/m^2°C. The moisture content in the air surrounding the plate is also specified as zero. All other material and thermodynamic properties not specified in this section are calculated as a function of temperature via the curve fits presented in Chapter III.

DRY FLAT PLATE

In this section the surface temperature results (at $z=L$) of a dry plate (0% initial moisture content or $M_0=0$) with the previously specified material properties, initial conditions and boundary conditions are presented via Figure 14. From this figure it is

evident that the surface temperature increases rapidly in the early stages ($0\,\text{s} < t < 100\,\text{s}$) of the plate's exposure to hot air. After this point the temperature continues to increase, but at a much slower rate which is shown to decay with time. This is expected as heat transfer at the surface depends on the difference in plate surface and free stream temperatures. The results of Figure 14 are used in subsequent sections to assess the effect of moisture content, M_0, and diffusion coefficients (more specifically D_0) on the surface temperature of moisture bearing epoxy plates.

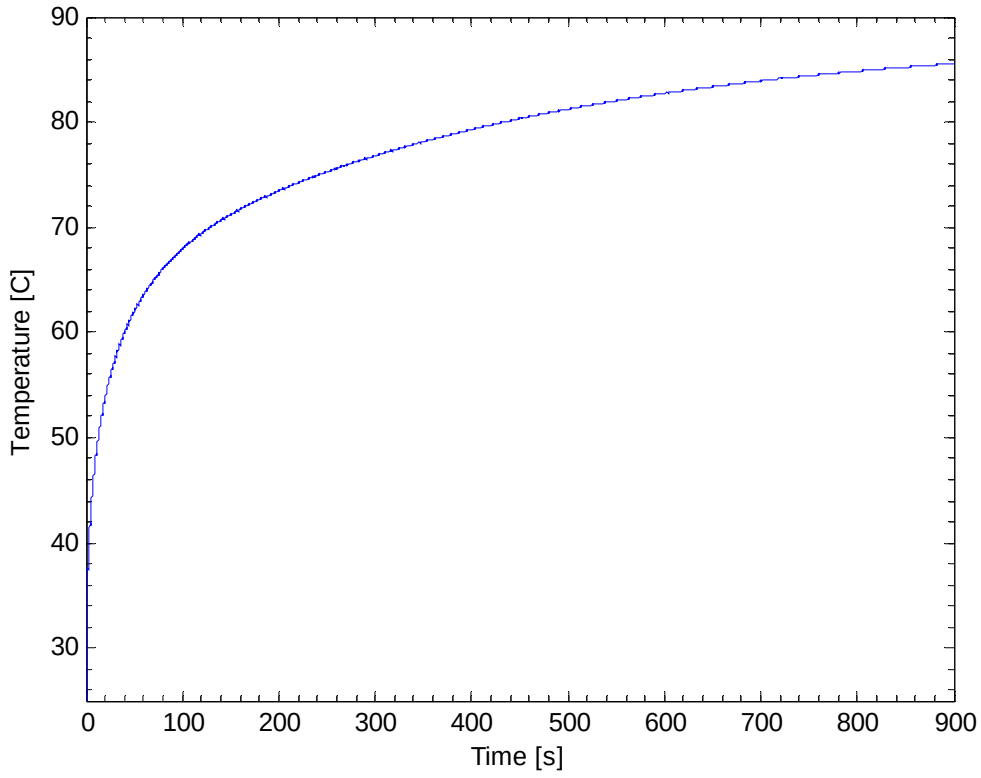

Figure 14. Surface temperature (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=0%).

FLAT PLATE WITH 5 % INITIAL MOISTURE CONTENT

In this section the temperature and energy transfer results (at $z=L$) are presented for plates where M_0=5% and all other conditions are as specified in the beginning of the chapter. In Figure 15 surface temperature vs. time profiles are provided for plates with three different D_0 values as per Table 2. The temperature vs. time profile of a dry plate is also presented in this figure.

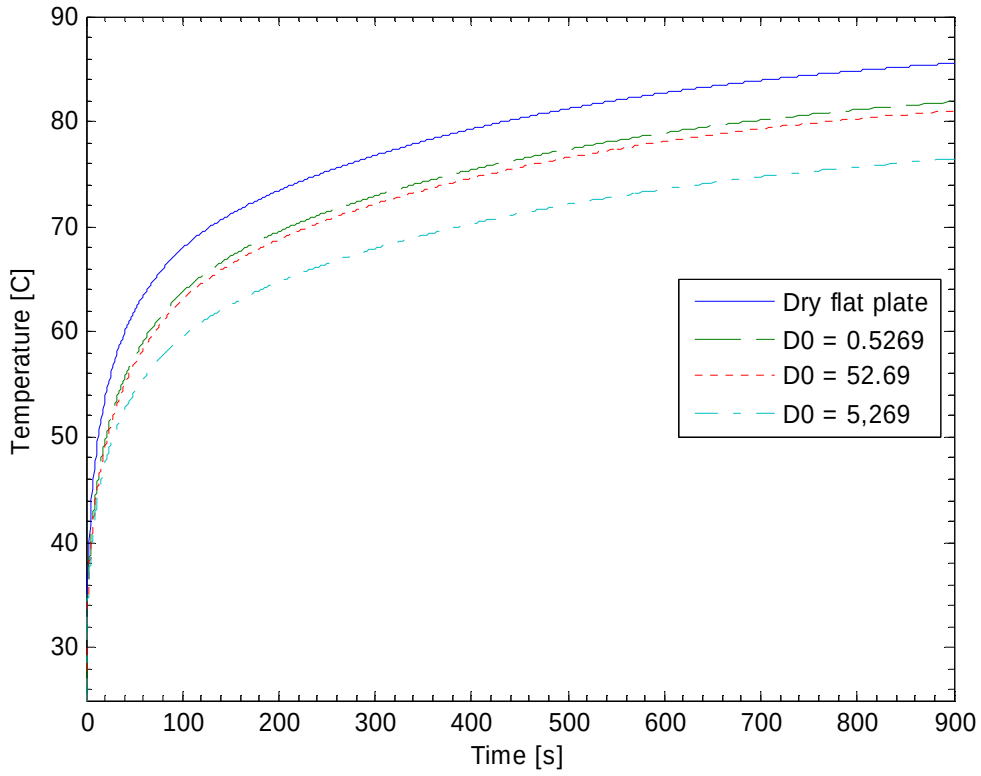

Figure 15. Surface temperature ($z=L$) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=5%).

From Figure 15 it is apparent that an increase in D_0 results in a reduction of surface temperature. This is because the resistance of molecular moisture diffusion in

the plate decreases with increasing D_0. Thus, as D_0 increases moisture is able to diffuse more rapidly through the plate to the surface where it is able to evaporate.

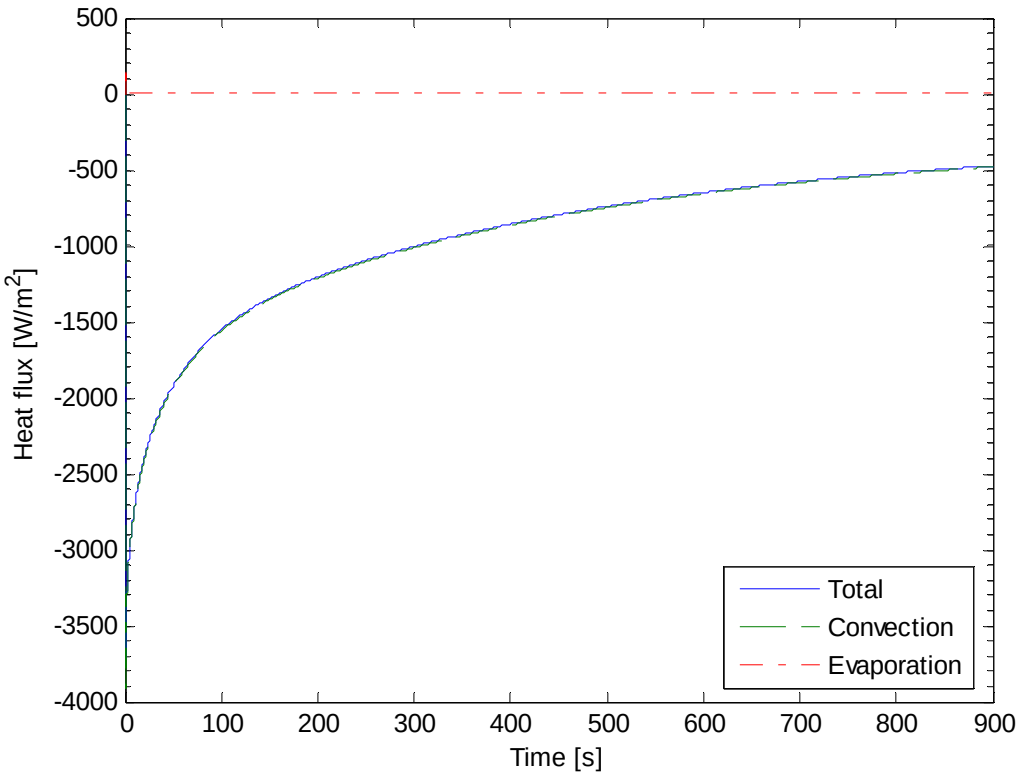

Figure 16. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s, M_0=5%).

It should be noted from the energy equation, Eq. (2.29), that an increase in D_0 (and thus evaporation) is not the only factor affecting the surface temperature profile. A careful examination of this equation makes it apparent that an increase in the specific heat, c_P, of the plate will result in a reduction of the surface temperature vs. time profile. Since the specific heat of moisture is greater than the specific heat of a dry plate, moist

plates will posses lower temperature values over time than will dry plates. This is simply because a moist plate possesses a larger specific heat value than does a dry plate.

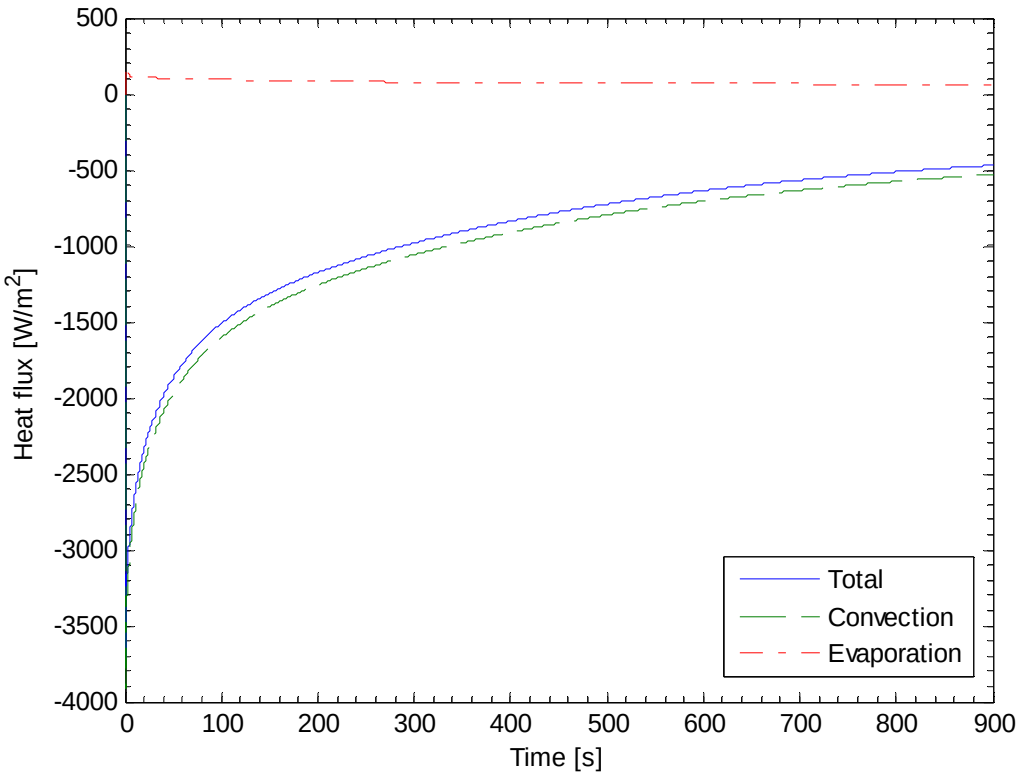

Figure 17. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s, M_0=5%).

To assess the effect of evaporation on the surface temperature profiles of Figure 15 it is convenient to examine the contributions of the rate of energy entering or leaving the plate. Figures 16, 17 and 18 provide plots that allow for this assessment. Each figure contains three heat flux or rate of energy transfer curves. The *Evaporation* curve is a plot of the rate of energy flux from the surface due to evaporation. This quantity is positive indicating that heat due to evaporation is leaving the plate and entering the

environment. The *Convection* curve depicts the heat flux entering the plate from the environment due to the difference between the plate surface and air temperatures. This quantity is negative indicating that heat is entering the plate. The *Total* curve represents the total heat flux entering the plate due to both evaporation and convection. These three curves allow the effects of evaporation to be compared with the effects of convection on the overall surface temperature of a plate.

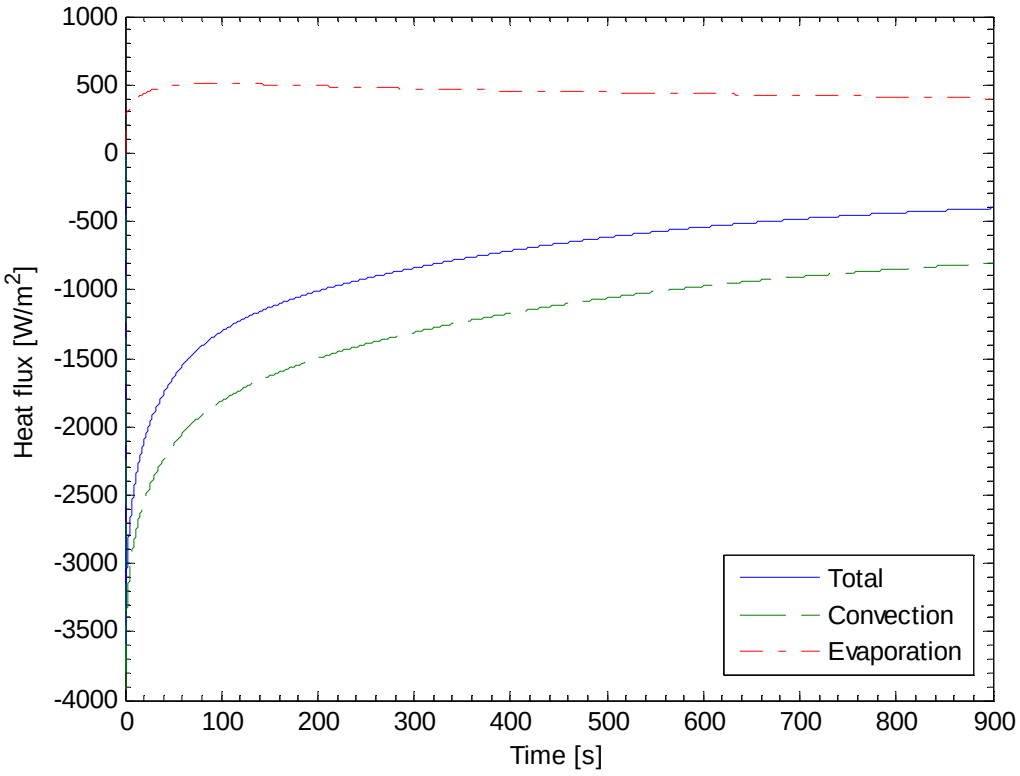

Figure 18. Heat flux contributions at plate surface ($z=L$) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s, M_0=5%).

It should be noted from Figure 16 that for D_0=0.5269 m^2/s, heat transfer due to evaporation is negligible. This specimen however is observed to possess a lower

temperature profile than a dry plate. This distinction is thus attributed to differences in specific heats rather than the effects of evaporation. From Figure 17 it is evident that the D_0=52.69 m^2/s specimen produces a very small amount of evaporation. As a result the temperature profile is very similar to the D_0=0.5269 m^2/s specimen. For the D_0=5,269 m^2/s specimen, however, the effect of evaporation is much more pronounced as is evident in Figure 18. As a result, this specimen is shown to have a much lower temperature profile than the other two (D_0=0.5269 m^2/s and D_0=52.69 m^2/s).

FLAT PLATE WITH 10 % INITIAL MOISTURE CONTENT

In this section the temperature and energy transfer results (at $z=L$) are presented for plates where M_0=10% and all other conditions are as specified in the beginning of the chapter. In Figure 19 surface temperature vs. time profiles are provided for plates with three different D_0 values as per Table 2. The temperature vs. time profile of a dry plate is also presented in this figure.

As was the case in the previous section, it is apparent from Figure 19 that an increase in D_0 results in a reduction of surface temperature. This is again attributed to the fact that the resistance of molecular moisture diffusion in the plate decreases with increasing D_0. Thus, as D_0 increases moisture is able to diffuse more rapidly through the plate to the surface where it is able to evaporate.

It should again be noted from the energy equation, that an increase in D_0 (and thus evaporation) is not the only factor affecting the surface temperature profile. The change in specific heat of the plate due to the presence of molecular moisture also plays

an important role on the surface temperature profile. This effect is enhanced as M_0 increases.

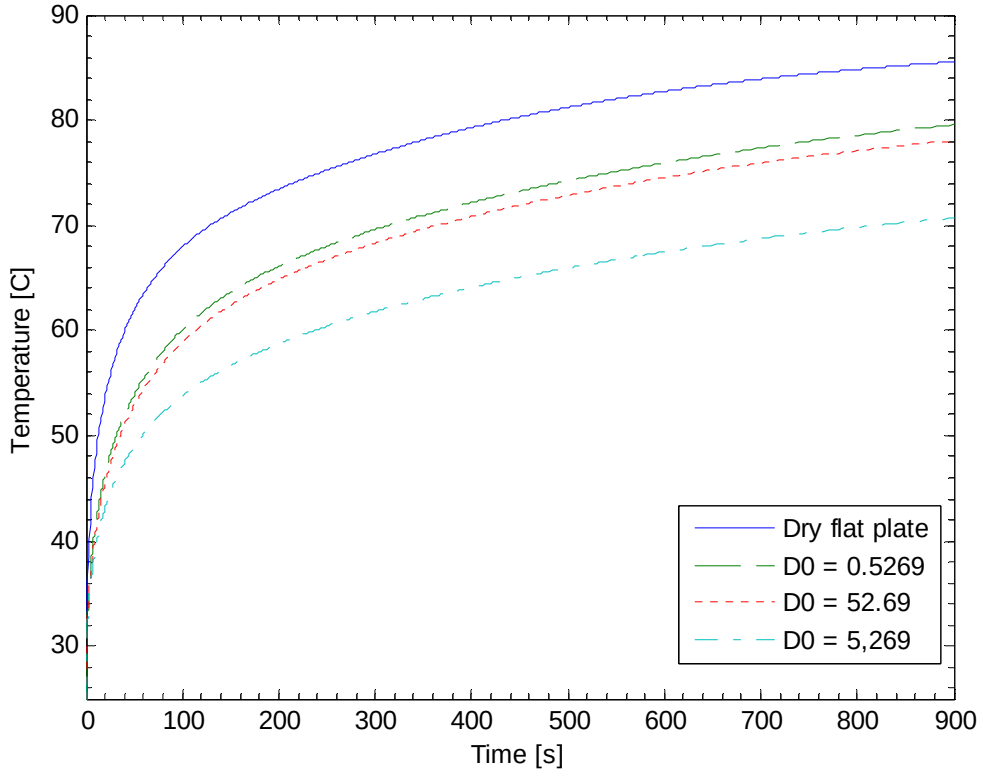

Figure 19. Surface temperature (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=10%).

To assess the effect of evaporation on the surface temperature profiles of Figure 19 it is once again convenient to examine the contributions of the rate of energy entering or leaving the plate. Figures 20, 21 and 22 provide plots that allow for this assessment.

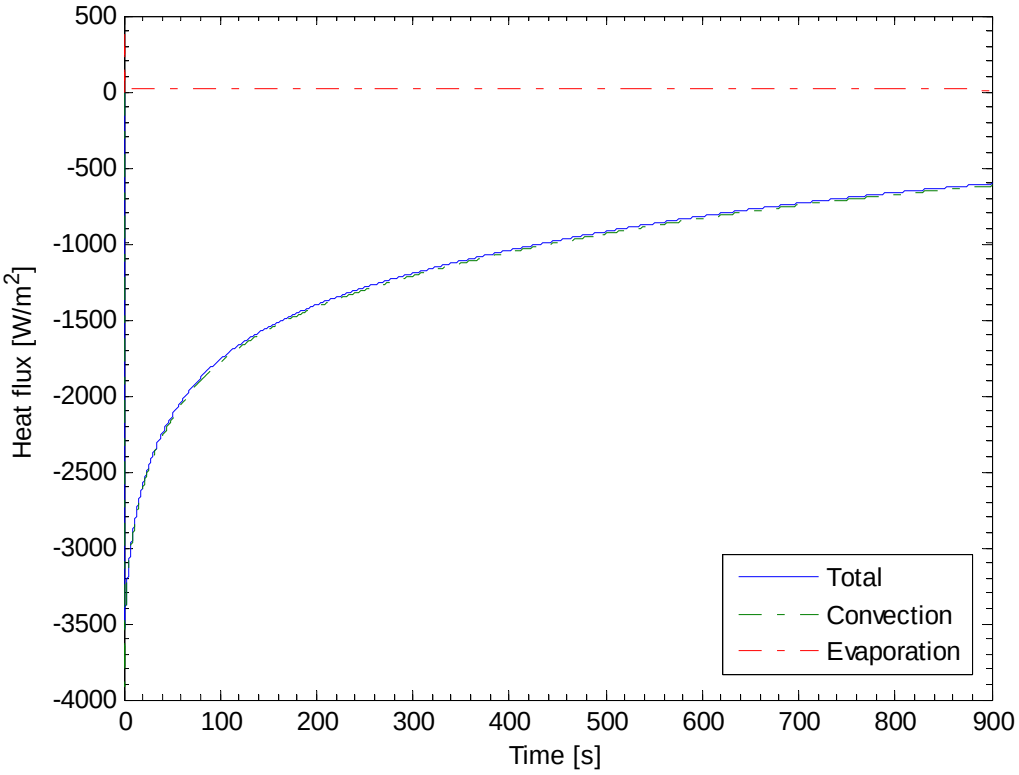

Figure 20. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s, M_0=10%).

It should be noted from Figure 20 that for D_0=0.5269 m^2/s, heat transfer due to evaporation is negligible. This specimen however is observed to possess a lower temperature profile than a dry plate. This distinction is again attributed to differences in specific heats rather than the effects of evaporation. From Figure 21 it is evident that the D_0=52.69 m^2/s specimen produces a very small amount of evaporation. As a result the temperature profile is very similar to the D_0=0.5269 m^2/s specimen. For the D_0=5,269 m^2/s specimen, however, the effect of evaporation is much more pronounced as is

evident in Figure 22. As a result, this specimen is shown to have a much lower temperature profile than the other two (D_0=0.5269 m^2/s and D_0=52.69 m^2/s).

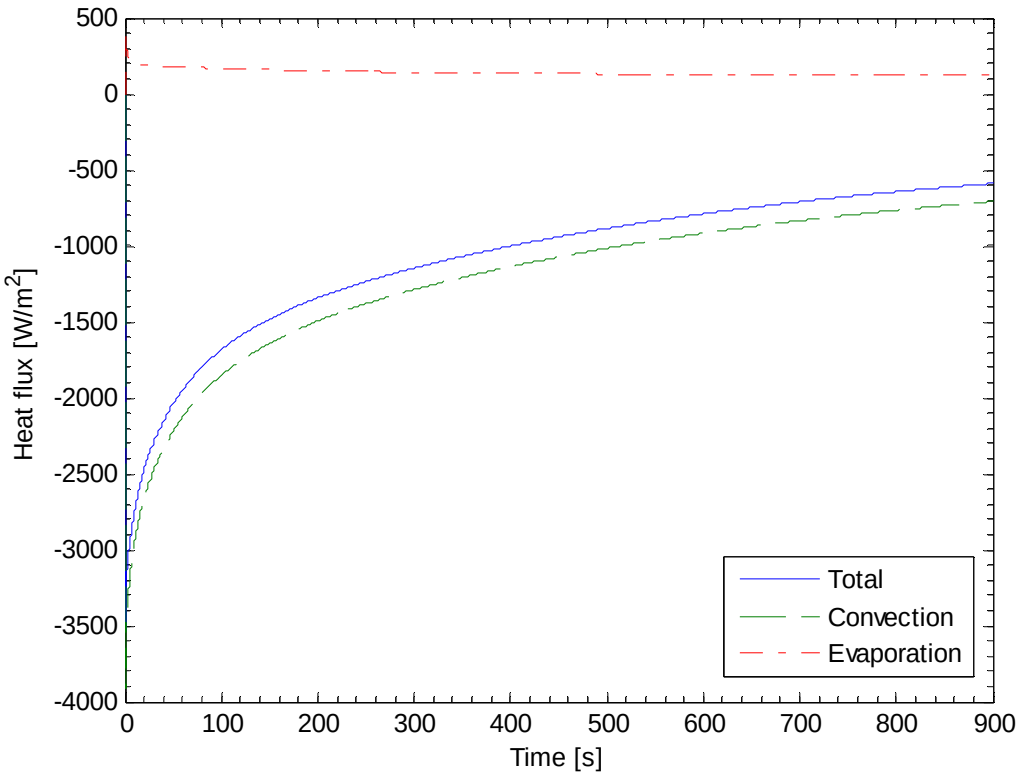

Figure 21. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s, M_0=10%).

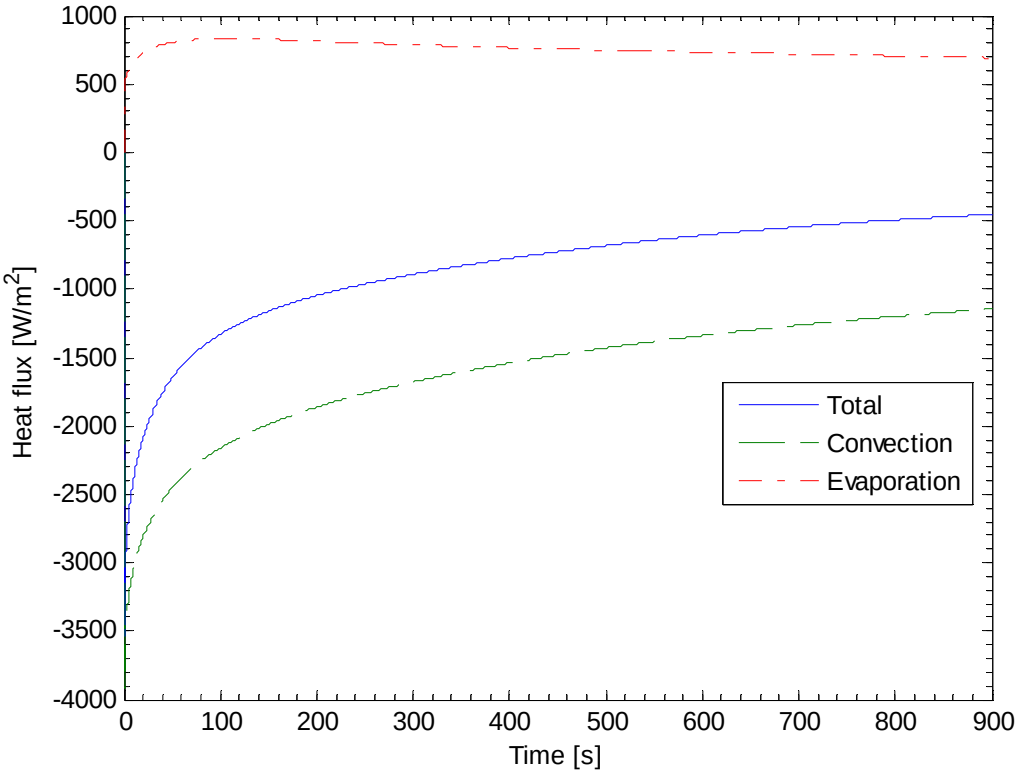

Figure 22. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s, M_0=10%).

FLAT PLATE WITH 15 % INITIAL MOISTURE CONTENT

In this section the temperature and energy transfer results (at z=L) are presented for plates where M_0=15% and all other conditions are as specified in the beginning of the chapter. In Figure 23 surface temperature vs. time profiles are provided for plates with three different D_0 values as per Table 2. The temperature vs. time profile of a dry plate is also presented in this figure.

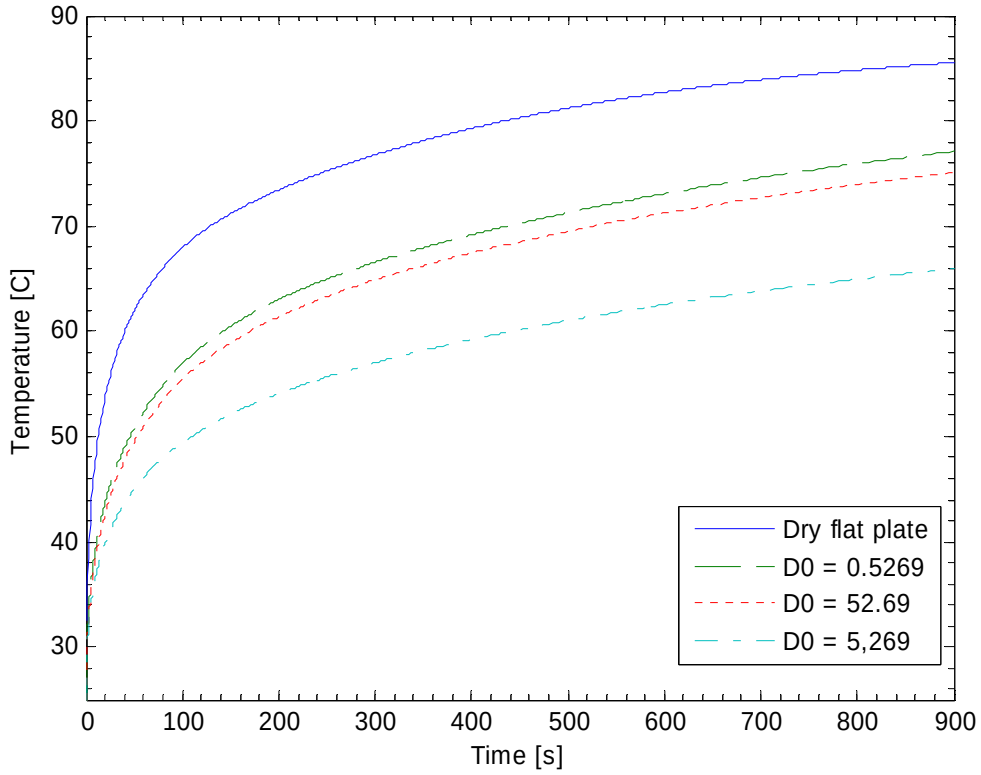

Figure 23. Surface temperature (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=15%).

As was the case in the previous sections, it is apparent from Figure 23 that an increase in D_0 results in a reduction of surface temperature. This is again attributed to the fact that the resistance of molecular moisture diffusion in the plate decreases with increasing D_0. Thus, as D_0 increases moisture is able to diffuse more rapidly through the plate to the surface where it is able to evaporate.

It should again be noted from the energy equation, that an increase in D_0 (increased evaporation rate) is not the only factor affecting the surface temperature profile. The change in specific heat of the plate due to the presence of molecular

moisture also plays an important role on the surface temperature profile. This effect is enhanced as M_0 increases.

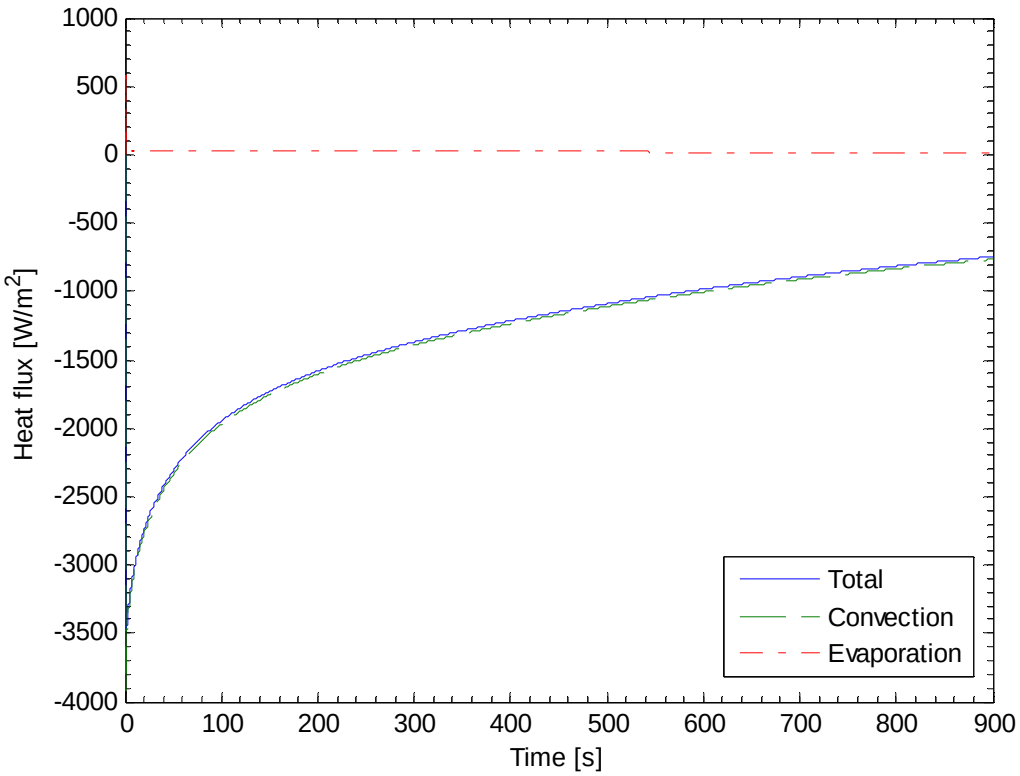

Figure 24. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s, M_0=15%).

To assess the effect of evaporation on the surface temperature profiles of Figure 23 it is once again convenient to examine the contributions of the rate of energy entering or leaving the plate. Figures 24, 25 and 26 provide plots that allow for this assessment.

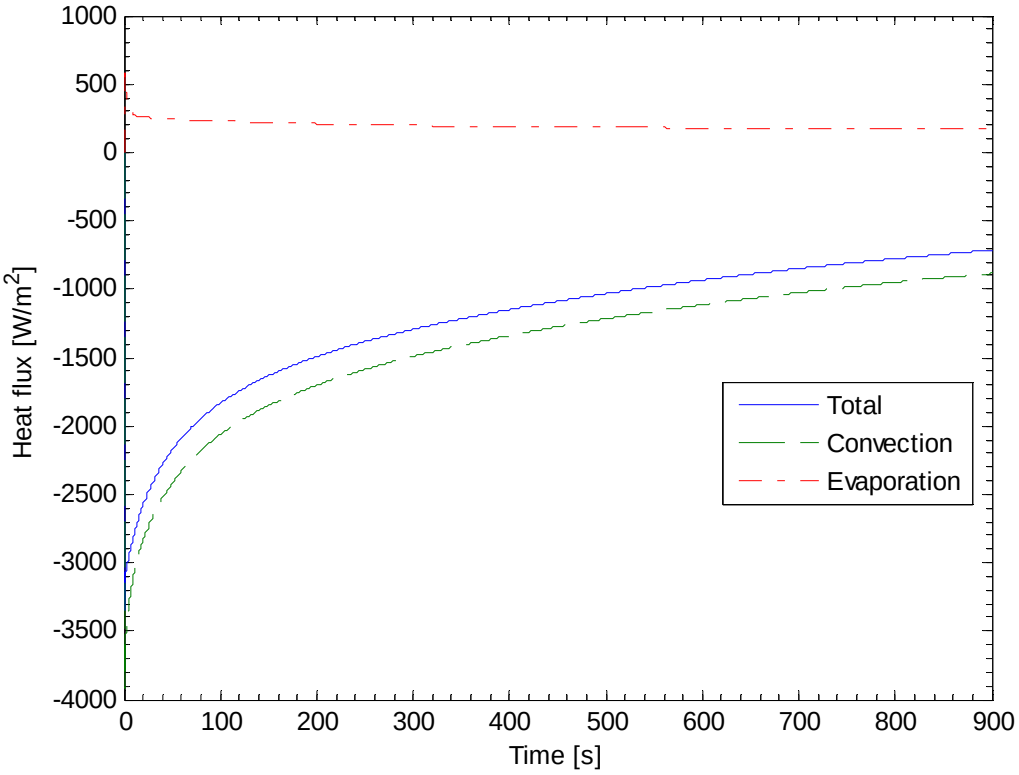

Figure 25. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s, M_0=15%).

It should be noted from Figure 24 that for D_0=0.5269 m^2/s, heat transfer due to evaporation is once again negligible. This specimen however is observed to possess a lower temperature profile than a dry plate. This distinction is again attributed to differences in specific heats rather than the effects of evaporation. From Figure 25 it is evident that the D_0=52.69 m^2/s specimen produces a very small amount of evaporation. As a result the temperature profile is once again very similar to the D_0=0.5269 m^2/s specimen. For the D_0=5,269 m^2/s specimen, however, the effect of evaporation is much more pronounced as is evident in Figure 26. As a result, this specimen is again shown to

have a much lower temperature profile than the other two (D_0=0.5269 m^2/s and D_0=52.69 m^2/s).

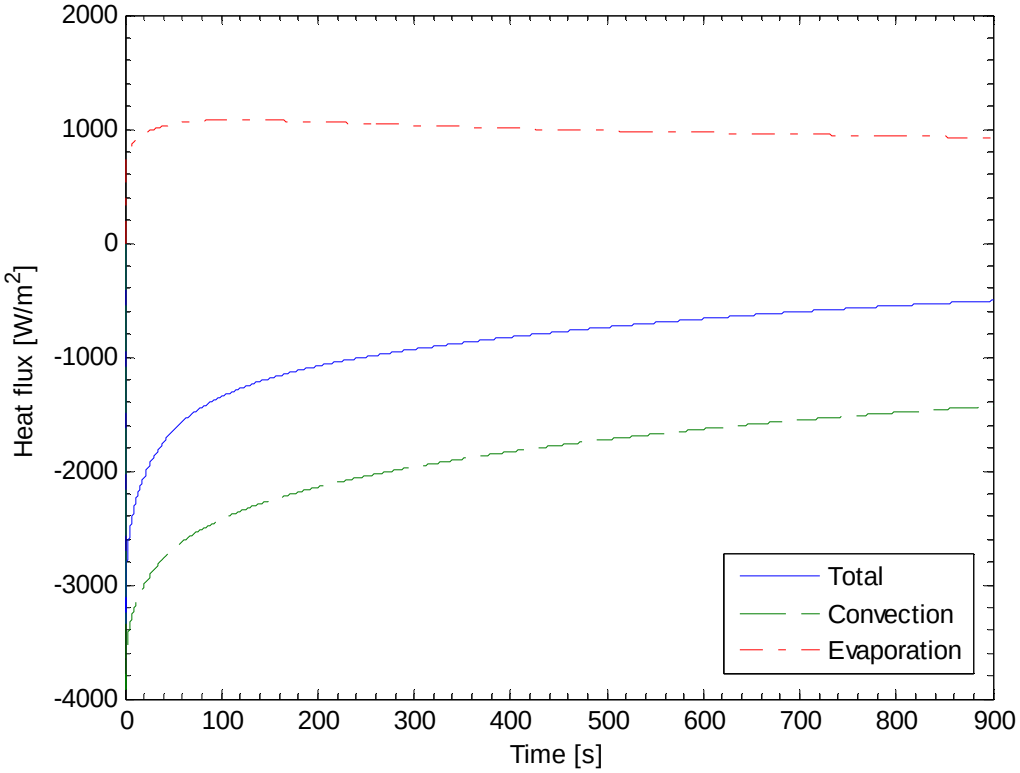

Figure 26. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s, M_0=15%).

FLAT PLATE WITH 20 % INITIAL MOISTURE CONTENT

In this section the temperature and energy transfer results (at z=L) are presented for plates where M_0=20% and all other conditions are as specified in the beginning of the chapter. In Figure 27 surface temperature vs. time profiles are provided for plates with

three different D_0 values as per Table 2. The temperature vs. time profile of a dry plate is also presented in this figure.

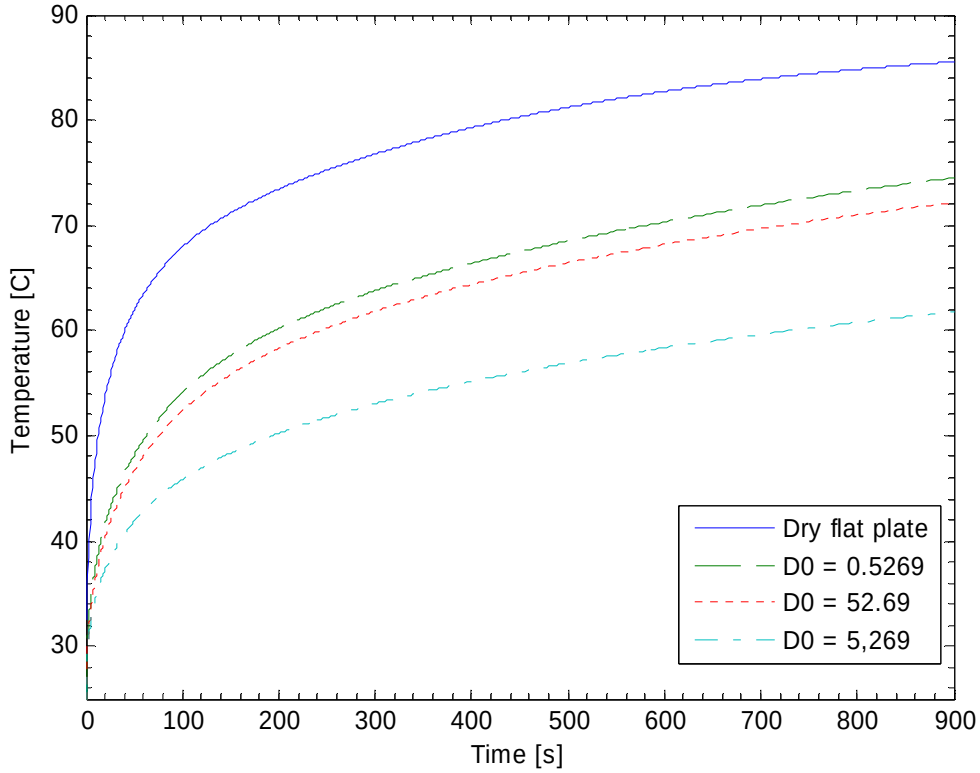

Figure 27. Surface temperature (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=20%).

As was the case in the previous sections, it is apparent from Figure 27 that an increase in D_0 results in a reduction of surface temperature. This is again attributed to the fact that the resistance of molecular moisture diffusion in the plate decreases with increasing D_0. Thus, as D_0 increases moisture is able to diffuse more rapidly through the plate to the surface where it is able to evaporate.

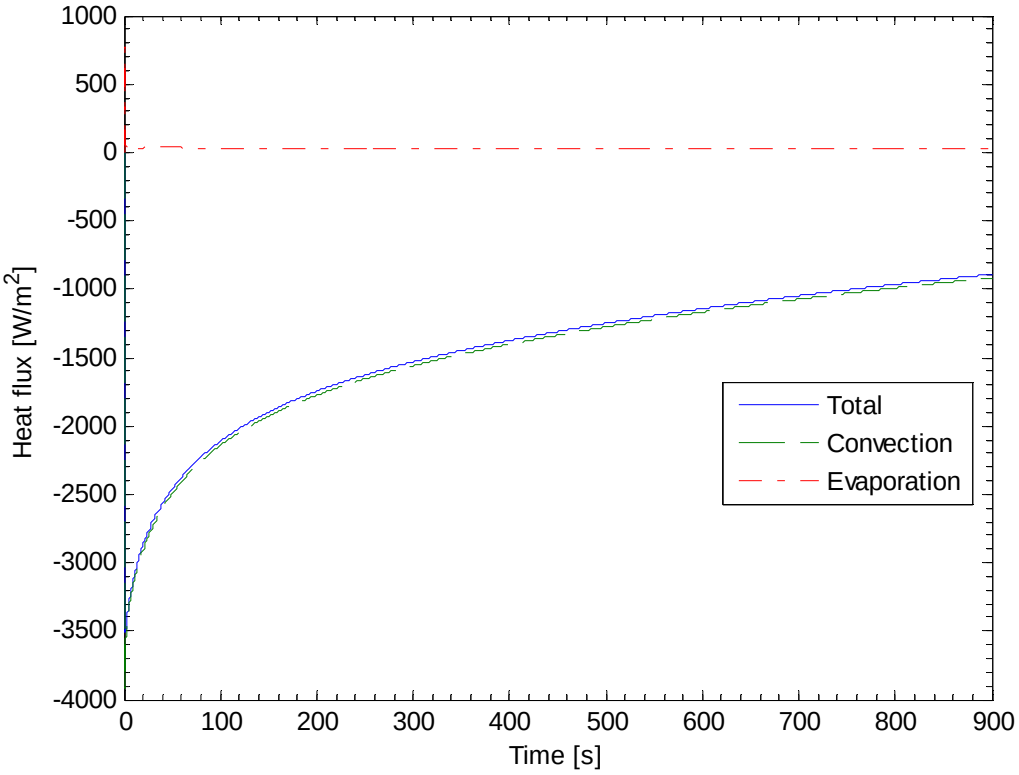

Figure 28. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s, M_0=20%).

It should again be noted from the energy equation, that an increase in D_0 (increased evaporation rate) is not the only factor affecting the surface temperature profile. The change in specific heat of the plate due to the presence of molecular moisture also plays an important role on the surface temperature profile. This effect is enhanced as M_0 increases.

To assess the effect of evaporation on the surface temperature profiles of Figure 27 it is once again convenient to examine the contributions of the rate of energy entering or leaving the plate. Figures 28, 29 and 30 provide plots that allow for this assessment.

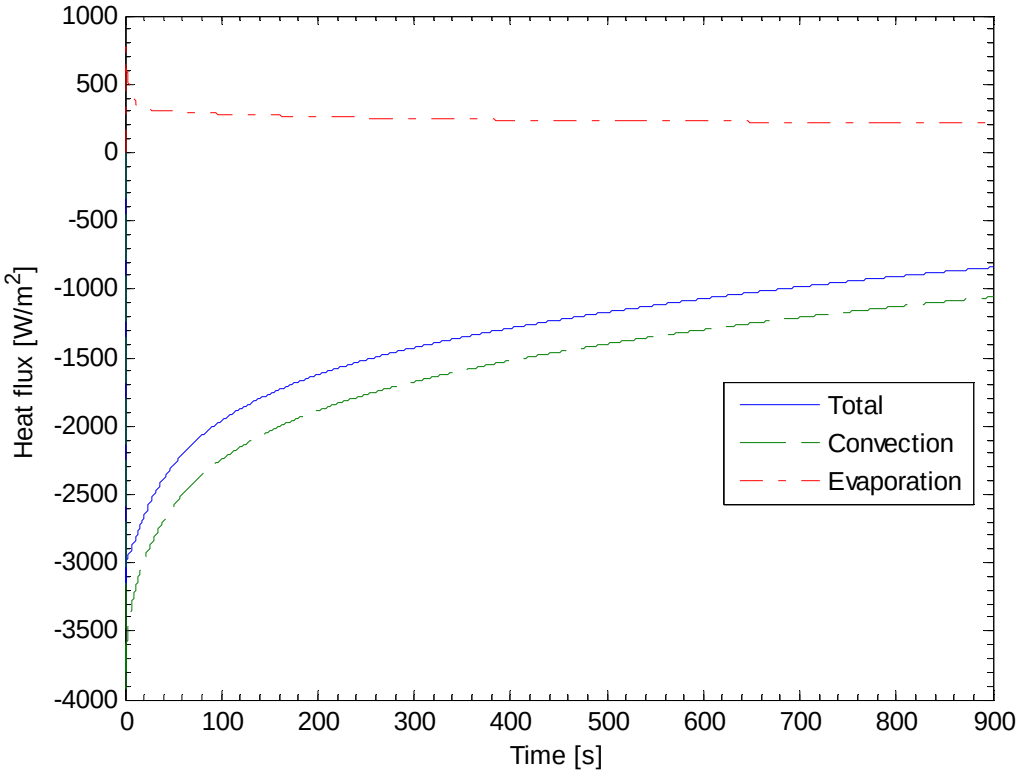

Figure 29. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s, M_0=20%).

It should be noted from Figure 28 that for D_0=0.5269 m^2/s, heat transfer due to evaporation is once again negligible. This specimen however is observed to possess a lower temperature profile than a dry plate. This distinction is again attributed to differences in specific heats rather than the effects of evaporation. From Figure 29 it is evident that the D_0=52.69 m^2/s specimen produces a slightly more significant amount of evaporation. As a result the temperature profile is slightly less than the D_0=52.69 m^2/s specimen. For the D_0=5,269 m^2/s specimen, however, the effect of evaporation is much more pronounced as is evident in Figure 30. As a result, this specimen is again shown to

have a much lower temperature profile than the other two (D_0=0.5269 m^2/s and D_0=52.69 m^2/s).

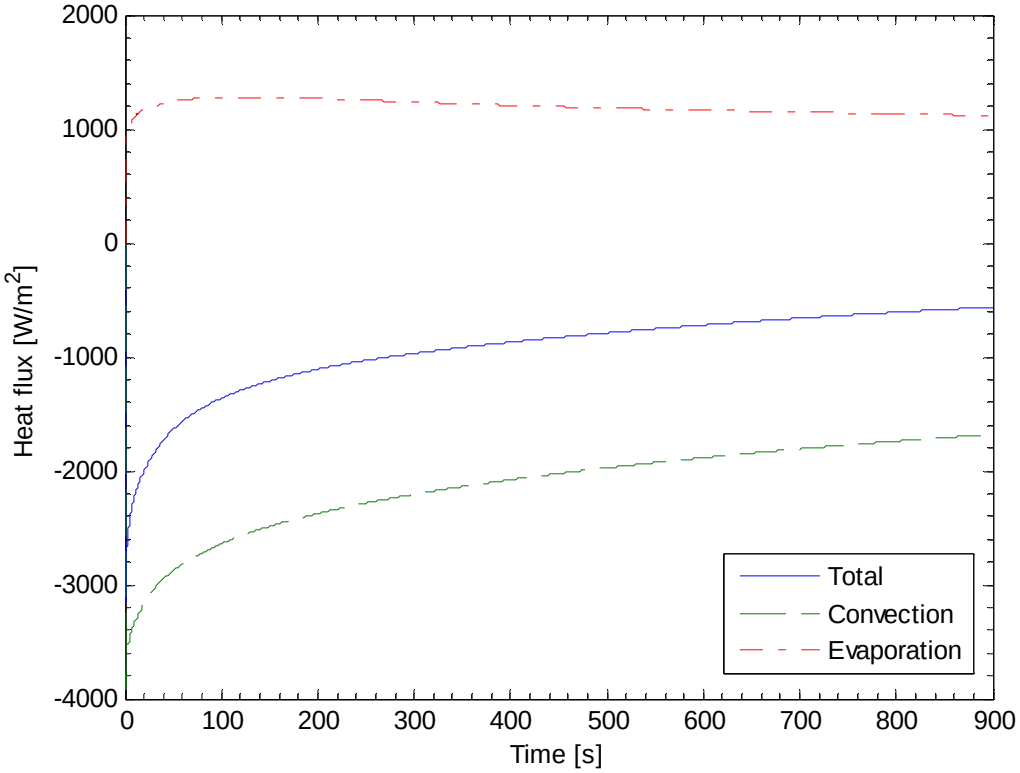

Figure 30. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s, M_0=20%).

FLAT PLATE WITH 25 % INITIAL MOISTURE CONTENT

In this section the temperature and energy transfer results (at z=L) are presented for plates where M_0=25% and all other conditions are as specified in the beginning of the chapter. In Figure 31 surface temperature vs. time profiles are provided for plates with

three different D_0 values as per Table 2. The temperature vs. time profile of a dry plate is also presented in this figure.

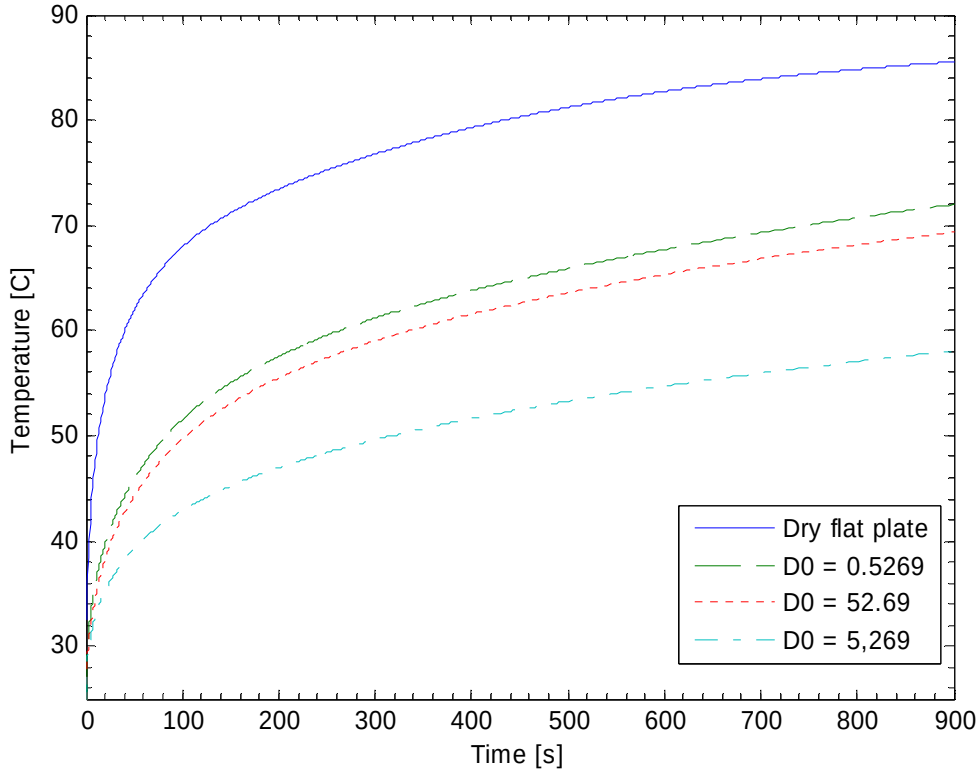

Figure 31. Surface temperature (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=25%).

As was the case in the previous sections, it is apparent from Figure 31 that an increase in D_0 results in a reduction of surface temperature. This is again attributed to the fact that the resistance of molecular moisture diffusion in the plate decreases with increasing D_0. Thus, as D_0 increases moisture is able to diffuse more rapidly through the plate to the surface where it is able to evaporate.

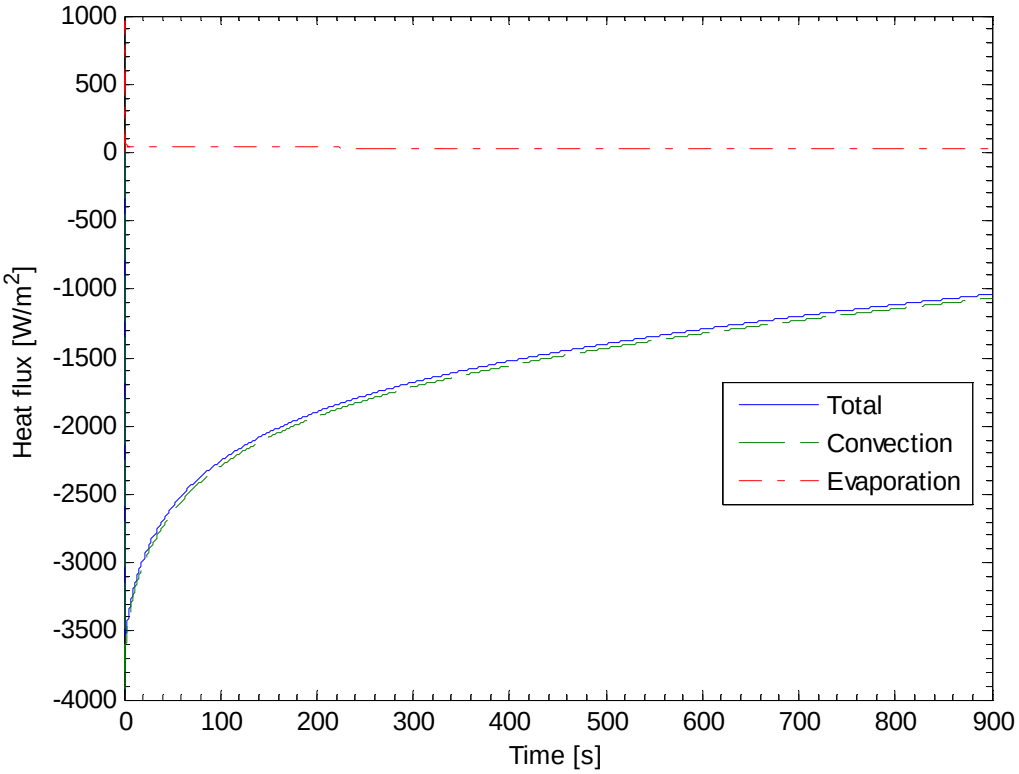

Figure 32. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s, M_0=25%).

It should again be noted from the energy equation, that an increase in D_0 (increased evaporation rate) is not the only factor affecting the surface temperature profile. The change in specific heat of the plate due to the presence of molecular moisture also plays an important role on the surface temperature profile. This effect is enhanced as M_0 increases.

To assess the effect of evaporation on the surface temperature profiles of Figure 31 it is once again convenient to examine the contributions of the rate of energy entering or leaving the plate. Figures 32, 33 and 34 provide plots that allow for this assessment.

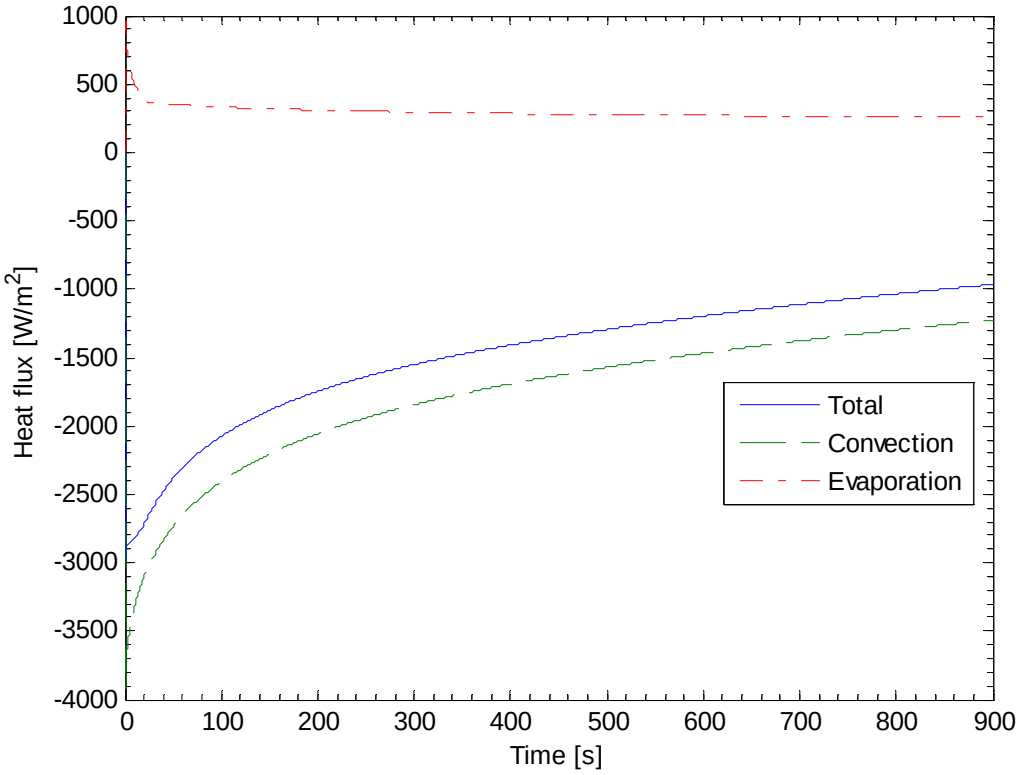

Figure 33. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s, M_0=25%).

It should be noted from Figure 32 that for D_0=0.5269 m^2/s, heat transfer due to evaporation is once again negligible. This specimen however is observed to possess a lower temperature profile than a dry plate. This distinction is again attributed to differences in specific heats rather than the effects of evaporation. From Figure 33 it is evident that the D_0=52.69 m^2/s specimen produces a slightly more significant amount of evaporation. As a result the temperature profile is slightly less than the D_0=0.5269 m^2/s specimen. For the D_0=5,269 m^2/s specimen, however, the effect of evaporation is much more pronounced as is evident in Figure 34. As a result, this specimen is again shown to

have a much lower temperature profile than the other two (D_0=0.5269 m^2/s and D_0=52.69 m^2/s).

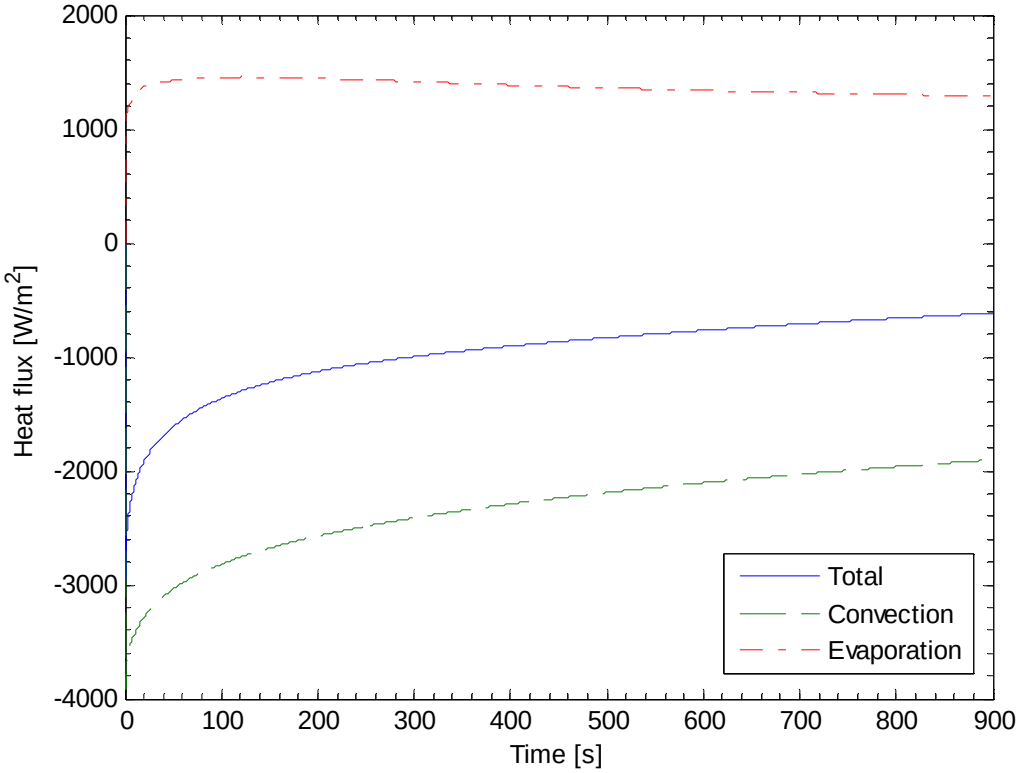

Figure 34. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s, M_0=25%).

FLAT PLATE WITH 30 % INITIAL MOISTURE CONTENT

In this section the temperature and energy transfer results (at z=L) are presented for plates where M_0=30% and all other conditions are as specified in the beginning of the chapter. In Figure 35 surface temperature vs. time profiles are provided for plates with

three different D_0 values as per Table 2. The temperature vs. time profile of a dry plate is also presented in this figure.

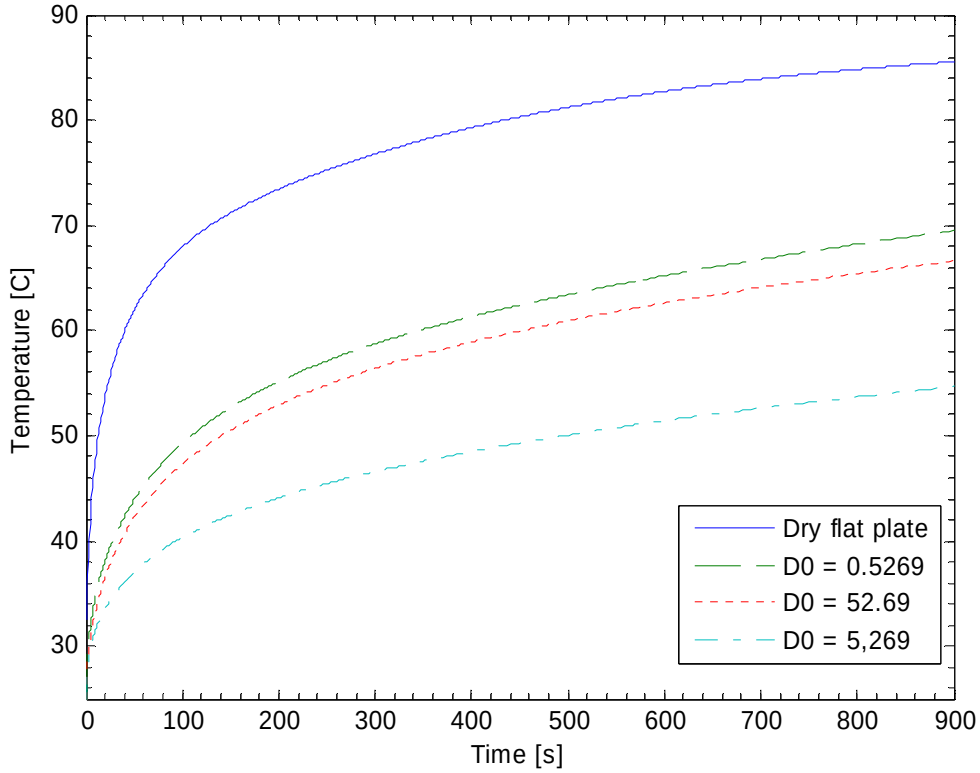

Figure 35. Surface temperature (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, M_0=30%).

As was the case in the previous sections, it is apparent from Figure 35 that an increase in D_0 results in a reduction of surface temperature. This is again attributed to the fact that the resistance of molecular moisture diffusion in the plate decreases with increasing D_0. Thus, as D_0 increases moisture is able to diffuse more rapidly through the plate to the surface where it is able to evaporate.

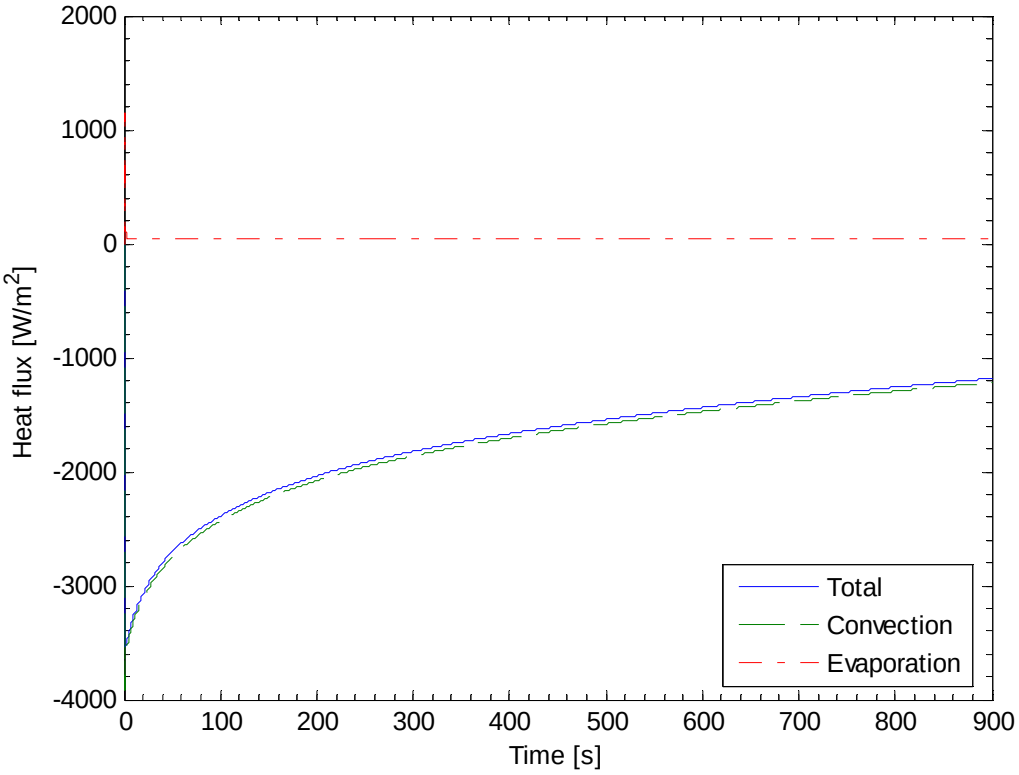

Figure 36. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s, M_0=30%).

It should again be noted from the energy equation, that an increase in D_0 (increased evaporation rate) is not the only factor affecting the surface temperature profile. The change in specific heat of the plate due to the presence of molecular moisture also plays an important role on the surface temperature profile. This effect is enhanced as M_0 increases.

To assess the effect of evaporation on the surface temperature profiles of Figure 35 it is once again convenient to examine the contributions of the rate of energy entering or leaving the plate. Figures 36, 37 and 38 provide plots that allow for this assessment.

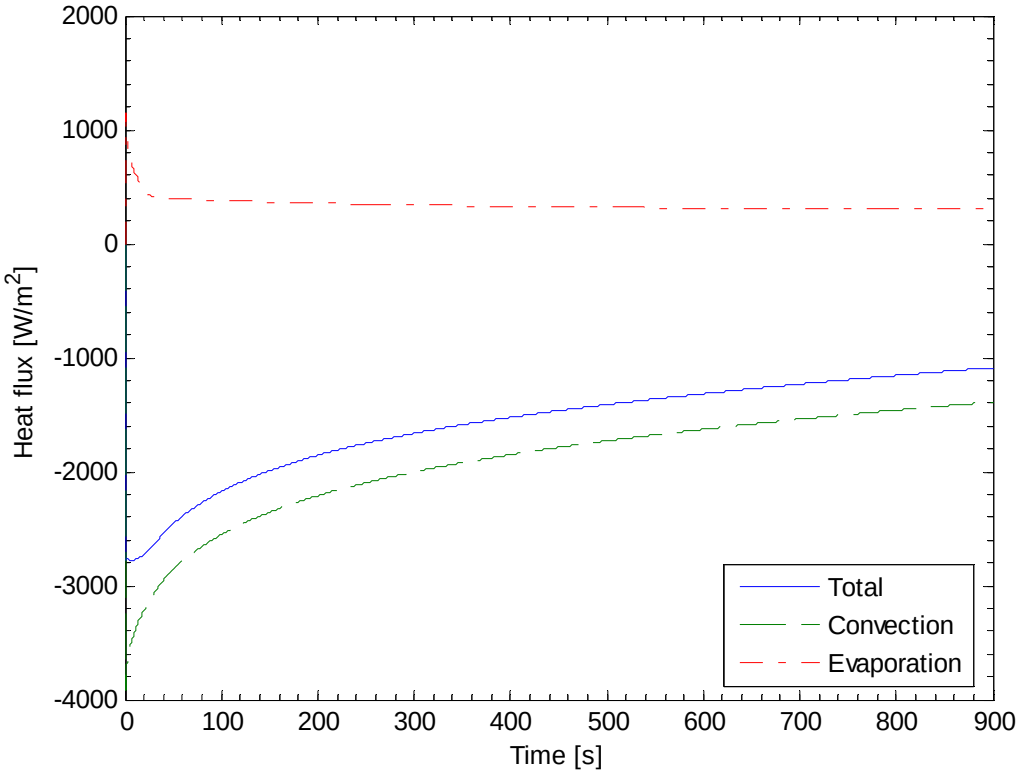

Figure 37. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s, M_0=30%).

It should be noted from Figure 36 that for D_0=0.5269 m^2/s, heat transfer due to evaporation is once again negligible. This specimen however is observed to possess a lower temperature profile than a dry plate. This distinction is again attributed to differences in specific heats rather than the effects of evaporation. From Figure 37 it is evident that the D_0=52.69 m^2/s specimen produces a slightly more significant amount of evaporation. As a result the temperature profile is slightly less than the D_0=0.5269 m^2/s specimen. For the D_0=5,269 m^2/s specimen, however, the effect of evaporation is much more pronounced as is evident in Figure 38. As a result, this specimen is again shown to

have a much lower temperature profile than the other two (D_0=0.5269 m^2/s and D_0=52.69 m^2/s).

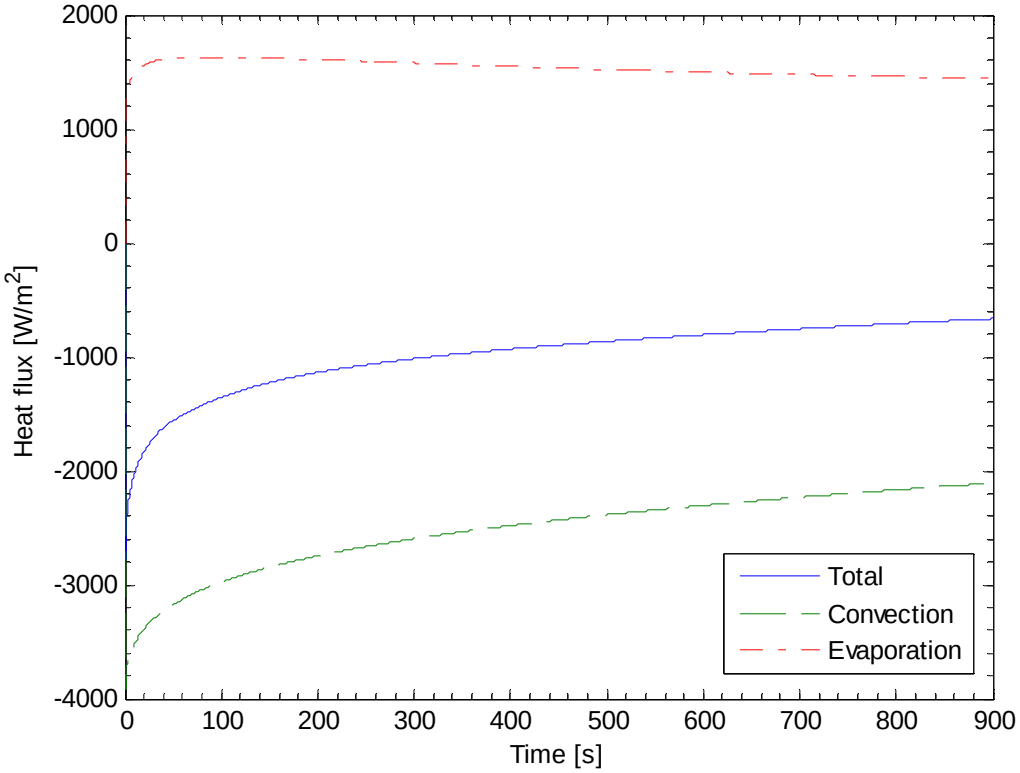

Figure 38. Heat flux contributions at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s, M_0=30%).

COMPARISON OF RESULTS

In this section the results of the preceding sections (M_0=0%, M_0=5%, M_0=10%, M_0=15%, M_0=20%, M_0=25% and M_0=30%) are compared. In these preceding sections, figures are provided which allow for the comparison of the results of plates with varying D_0 at a given M_0. For the sake of completeness Figures 39 through 44 and Table 6 are

provided in this section as a means of comparing the results of different M_0 for a given D_0.

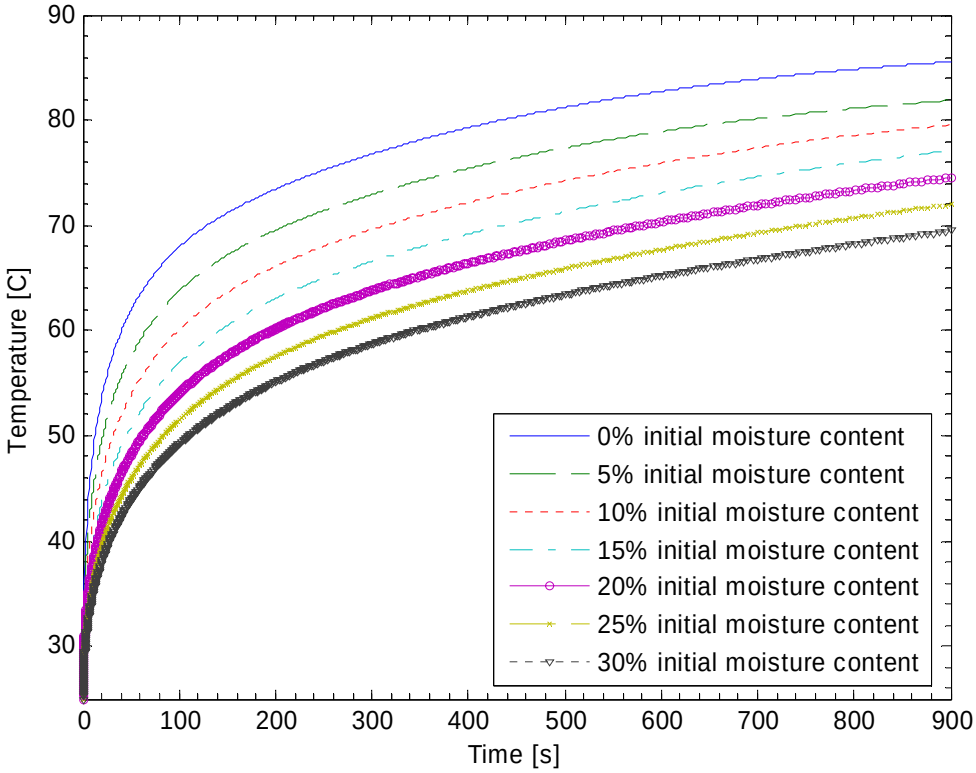

Figure 39. Surface temperature at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s).

Figure 39 provides the results for plate specimens where D_0=0.5269 m^2/s. The heat transfer figures contained in the preceding sections make it evident that the temperature profiles of the specimens presented in Figure 39 are unaffected by moisture transport since the corresponding evaporation rates are nearly zero.. It may be concluded nevertheless, that the initial moisture content, M_0, exhibits a significant affect upon the surface temperature profile of these plates. This effect is attributed to the

change in the overall specific heat of a plate due to the presence of molecular moisture. It is apparent from Figure 39 that the magnitude of this effect increases with increased initial moisture content, M_0.

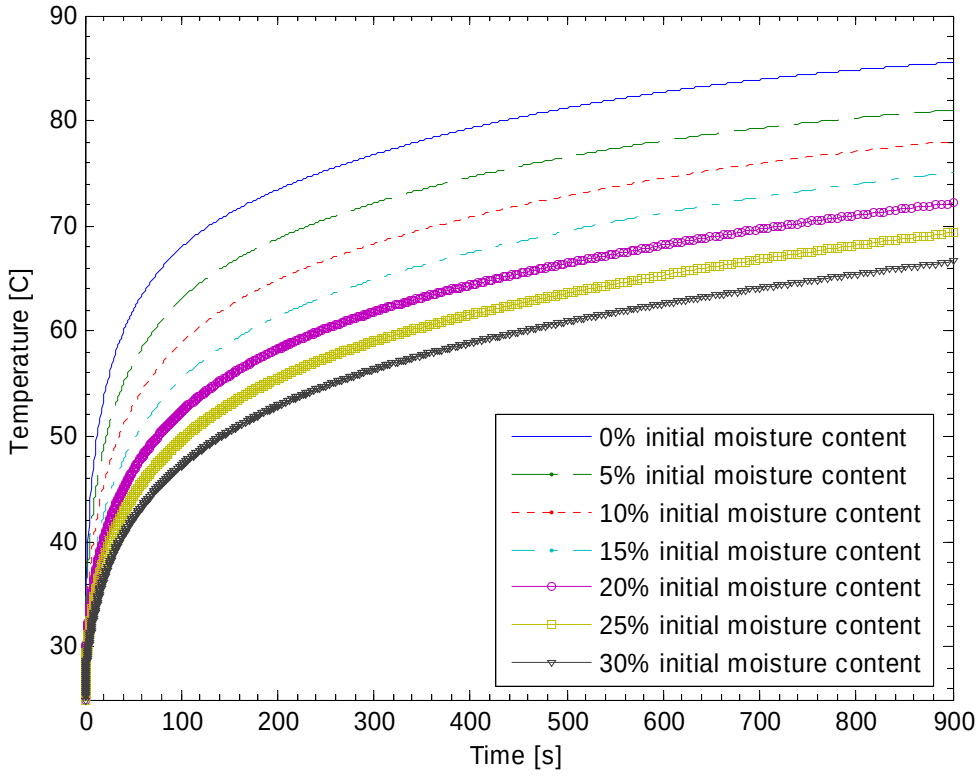

Figure 40. Surface temperature at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s).

A comparison of Figures 39 and 40 reveals that plate samples with D_0=52.69 m^2/s experience slightly lower temperature profiles than specimens where D_0=0.5269 m^2/s. The difference, however, is only on the order of about 1°C to 3°C (see Table 6). The order of the difference increases with increasing M_0.

The temperature profiles of the plate specimens as per Figure 41 (D_0=5,269 m^2/s) exhibit significant improvements over the samples previously presented in Figures 39 and 40. The temperature curves are much lower as energy transfer due to evaporation is of similar magnitude to the convection heat transfer.

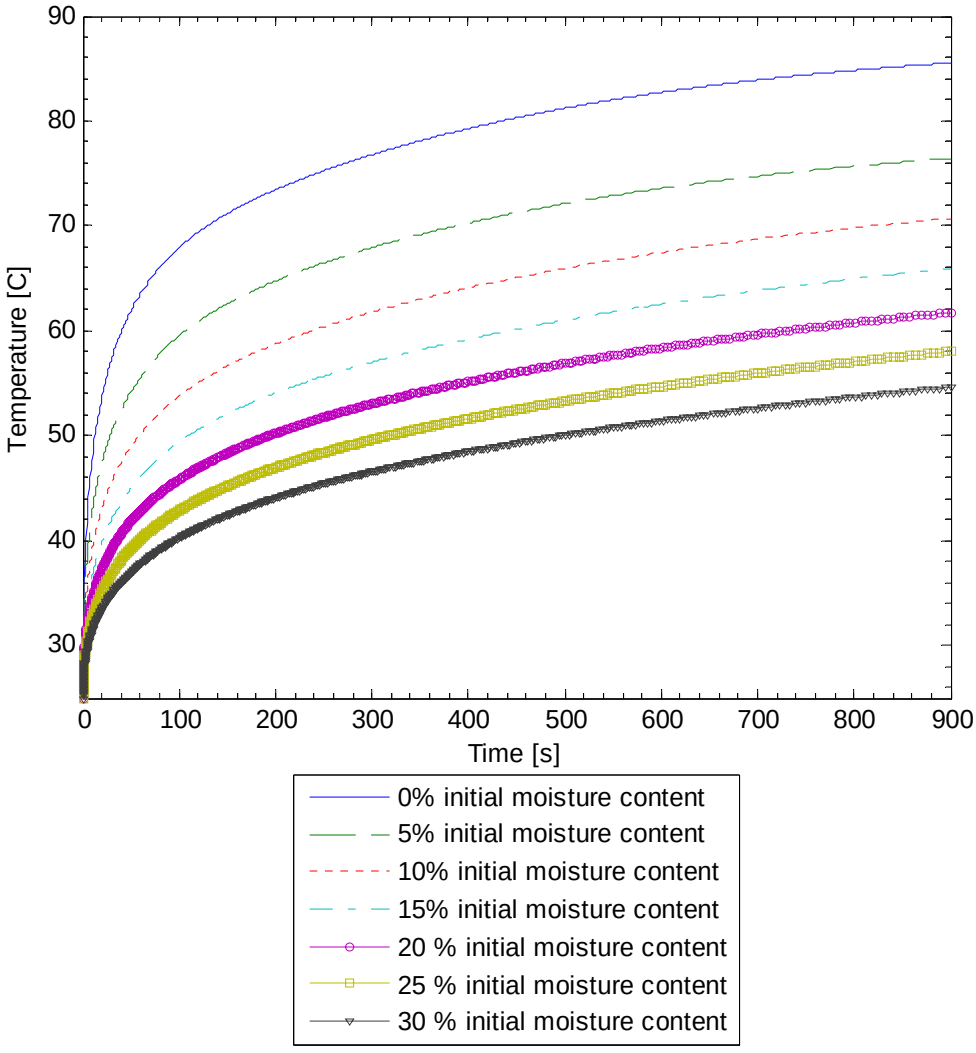

Figure 41. Surface temperature at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s).

In addition to Figures 39, 40 and 41; Figures 42, 43 and 44 are also provided to represent more clearly the surface temperature profiles during the early stages of the plate's exposure to the environment.

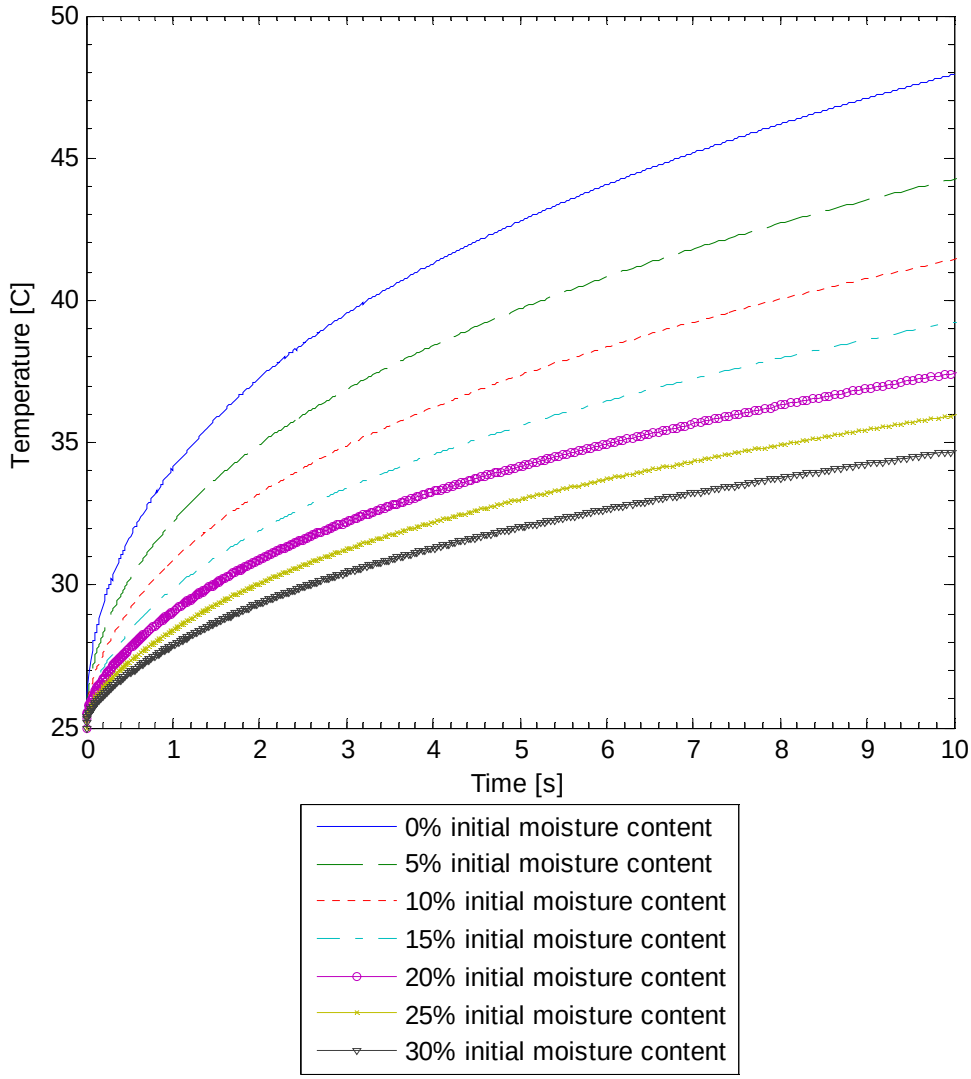

Figure 42. Surface temperature at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=0.5269 m^2/s).

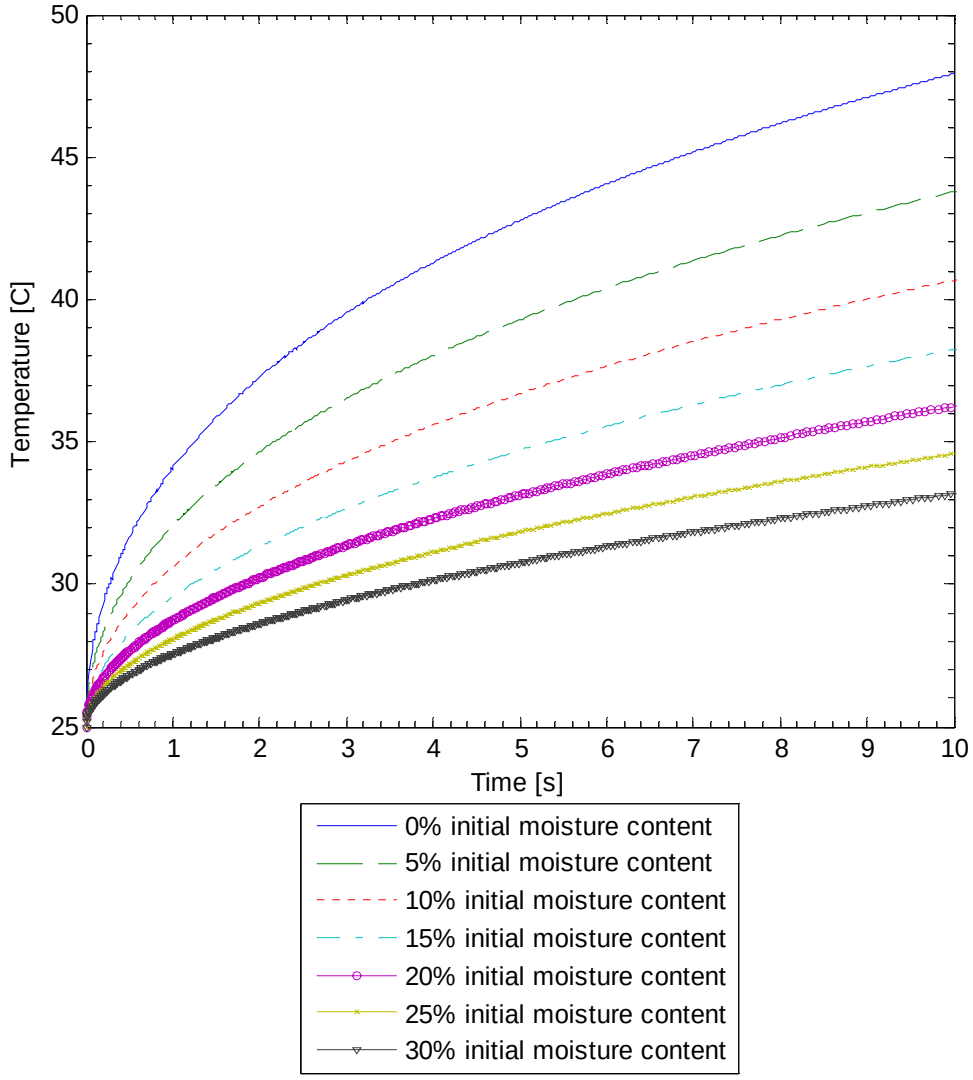

Figure 43. Surface temperature at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=52.69 m^2/s).

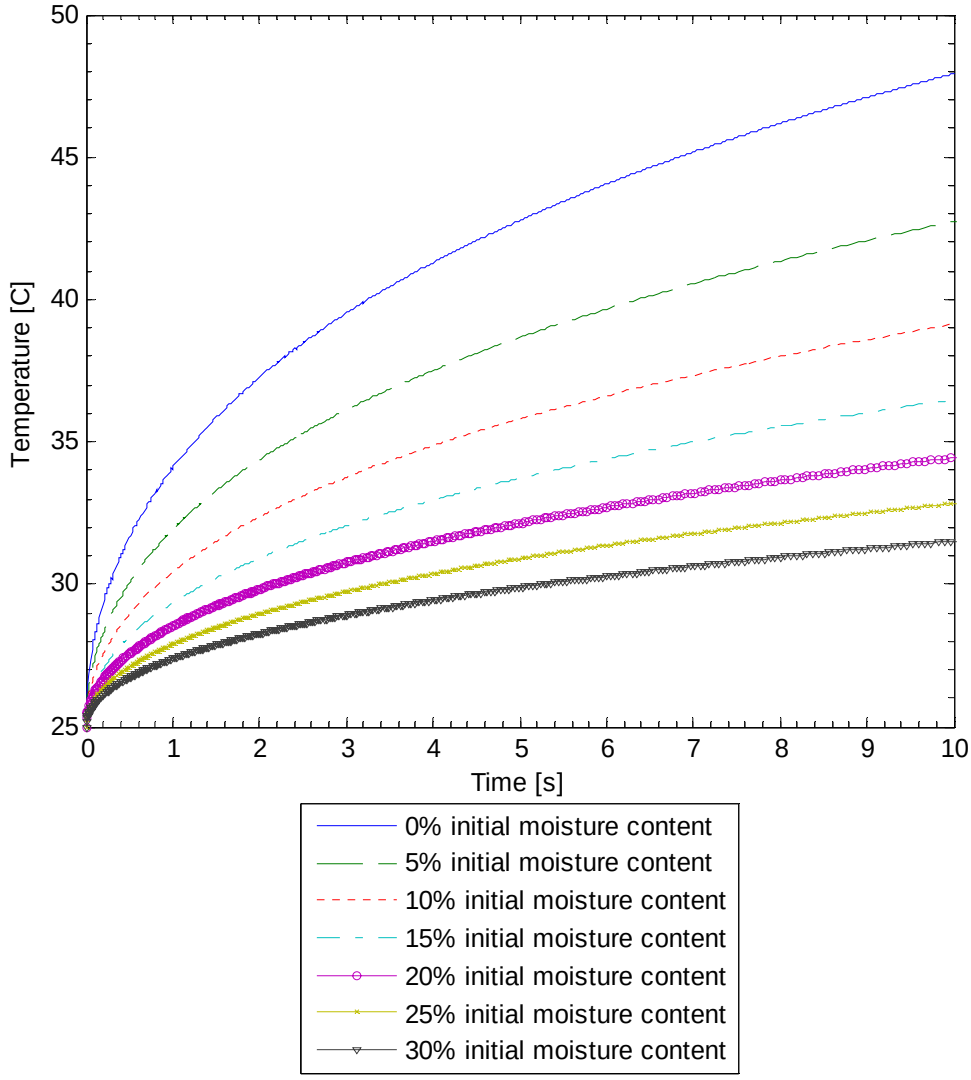

Figure 44. Surface temperature at plate surface (z=L) vs. time (T_∞=90°C, h=60 W/m^2°C, D_0=5,269 m^2/s).

Table 6. A comparison of final temperature values (t=900 s).

M_0	D_0=0.5269 m^2/s	D_0=52.69 m^2/s	D_0=5,269 m^2/s
0%	85.5 [C]	85.5 [C]	85.5 [C]
5%	81.9 [C]	81.0 [C]	76.5 [C]
10%	79.6 [C]	78.0 [C]	70.7 [C]
15%	77.1 [C]	75.1 [C]	65.9 [C]
20%	74.6 [C]	72.2 [C]	61.8 [C]
25%	72.0 [C]	69.4 [C]	58.0 [C]
30%	69.5 [C]	66.6 [C]	54.7 [C]

Table 6 provides a summary of temperature values at t=900 s as per Figures 39, 40 and 41. From this Table it is further evident that surface temperature is a function of both M_0 and D_0. As initial moisture content and diffusion coefficients increases the surface temperature profile decreases.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

SUMMARY AND CONCLUDING REMARKS

In this work evaporative cooling of composite plate structures was mathematically studied as a potential for reducing the external temperatures of military aircraft. A simplified model of the phenomena was developed and the governing equations for the coupled energy and mass transfer were derived from the conservation laws (mass and energy) per the continuum mechanics hypothesis. Constitutive relationships and additional assumptions were introduced to provide two governing equations containing two dependent variables (T and ρ_w). These coupled nonlinear partial differential equations were then approximated using a weak form Galerkin finite element formulation and solved at discrete time steps through the α–family of time approximation using the *Crank–Nicolson* scheme.

Temperature profiles versus time were presented in Chapter IV for a variety of initial molecular moisture contents, M_0, and diffusion coefficients, D. From the results it is evident that evaporative cooling possesses a potential for reducing the surface temperatures of plate structures that are exposed to external airflows at elevated temperatures. To achieve a significant reduction in surface temperature over time it is necessary for the plate structure to possess a molecular moisture diffusion coefficient, D, on the order of about 10^{-8} m^2/s (D_0=5,269 in the present study). The initial moisture content, M_0, is also shown to reduce the surface temperature.

In conclusion, evaporative cooling of molecular moisture bearing composite plates possesses the potential for reducing the external temperature of military aircraft. It should, however, be emphasized from the results of the present study that evaporative cooling only *reduces* the temperature versus time profile of flat plate structures. It *does not eliminate* the increase of surface temperatures due to exposures to external flows. For the findings of the present work to be useful it is necessary that the selected moisture bearing epoxy composite (i.e., high moisture particular fillers incorporated into high temperature AFR–PEPA–N polyimides) possesses the properties shown to be favorable in the present study. There are two necessary criteria. First the material must possess a diffusion coefficient, D, on the order of 10^{-8} m^2/s. In addition it is important that the material be capable of absorbing a significant amount of moisture (i.e., allows for a favorable value for M_0).

FUTURE RESEARCH

The model used in the present study relied on several key assumptions which were presented in Chapter II. The validity of the numerical results presented in Chapter IV therefore, depend significantly upon the validity of these assumptions. It may therefore be of interest to revisit several of these assumptions to refine the model.

It should be noted that in the 4^{th} assumption of Chapter II it was presumed that the diffusion of molecular moisture may be described by Fick's law of diffusion. Many studies [17,18,19] of molecular moisture transport in epoxy plate specimens have utilized this relationship with some success to describe the phenomenon. However,

some of this research [18,19] has demonstrated that moisture transport does deviate to some degree from Fick's law. In fact the diffusion coefficients, D, in the literature are often shown to vary with a change in boundary conditions [17,18]. The reason that Fick's law suffers in such a manner may be attributed to the fact that diffusion is a much more complex form of transport than heat transfer as it may depend upon more than just the concentration gradient of a single dependent variable. As a result several models have been proposed for more accurately describing the diffusion process of molecular moisture in epoxy composites [18]. One of these theories might therefore be utilized in the future to develop a more accurate set of governing equations than what is presented in this work.

The 6^{th} and 7^{th} assumptions presented in Chapter II may also need to be reevaluated. Eq. (2.1) presents a theorized assumption relating the partial densities of moisture within the plate (at the surface) and the water vapor (just above the plate surface). This assumption follows from Assumption 6, which assumes that evaporation does not occur within the plate. The validity of these assumptions (along with improved correlations) will have to be inferred by the subsequent experimental results. These results will allow the boundary conditions to be expressed more accurately and thus provide for more accurate numerical simulations.

It is also worth noting an additional implication implied by Assumption 6. Since evaporation is assumed to occur at the plate surfaces, the model is limited to the study of external flows that do not produce temperatures within the plate that will exceed the boiling temperature of water (i.e., $T_{\infty,t}$<100°C and $T_{\infty,b}$<100°C). It may therefore be of

interest to refine the present model such that evaporation is allowed to occur inside the plate. This type of model would be much more complicated as it would allow the *molecular water*/*water vapor* interface to move from the surface into the plate as a function of time. At this moving interface a discontinuity would exist in the partial density of water. In addition values for two separate diffusion coefficients, D, within the plate would need to be defined and quantified. One coefficient would describe the diffusion of water in its molecular liquid form (as in the case of the present study) and the other would describe the diffusion process of the molecular water vapor.

In addition to reevaluating some of the assumptions utilized in the present work it is also of interest to perform experimental work to obtain results that may be compared with the results of the present study. This will allow for the validation of the present model and provide additional insight into which assumptions may need to be reevaluated.

REFERENCES

1. Li, Y., Morgan, R. J., and Tschen, F. (2005). Structure–Property Relations of Phenylethynyl Terminated Imide Oligomers, In: *Proceedings of 50th ISSE*, *SAMPE 2005*, May 1–5, Long Beach, CA. In press.

2. Cengel, Y. A. and Boles, M. A. (2002). *Thermodynamics: An Engineering Approach*, 4th Edition, McGraw-Hill, New York.

3. Ju, J., Morgan, R. J., Creasy, T. C., Hafernik, S., and Shin, E. E. (2005). Statistical Approach to Evaluate Composite Damage, *ANTEC 2005 of Society of Plastics Engineering*, May 1–5, Boston, MA. In press.

4. Ribeiro, R., Morgan, R. J., Bonnaud, L., Choi, J., Lopata, V., Lu, J., and Sue, H.-J. (2005). Characterization of the Electon Beam Curing of an Epoxy Resin, *Journal of Composite Materials*, In press.

5. Bird, B. R., Stewart, W. E. and Lightfoot, E. N. (1960). *Transport Phenomena*, John Wiley & Sons, Inc., New York.

6. Reddy, J. N. (2006). *Introduction to Continuum Mechanics with Applications*, Lecture Notes (to appear under the imprint of Cambridge University Press, New York), Texas A&M University, College Station, TX.

7. Chadwick, P. (1976). *Continuum Mechanics: Concise Theory and Problems*, 2nd Edition, Dover Publications, New York.

8. Panton, R. L. (1996). *Incompressible Flow*, 2nd Edition, John Wiley & Sons, Inc., New York.

9. Welty, J. R., Wicks, C. E. and Wilson, R. E. (1969). *Fundamentals of Momentum, Heat and Mass Transfer*, John Wiley & Sons, Inc., New York.

10. Lienhard, J. H. IV. and Lienhard, J. H. V. (2005). *A Heat Transfer Textbook*, 3rd Edition, Phlogiston Press, Cambridge, MA.

11. Incropera, F. P. and DeWitt, D. P. (2002). *Fundamentals of Heat and Mass Transfer*, 5th Edition, John Wiley & Sons, Inc., New York.

12. Braud, L. M., Moreira, R. G. and Castell-Perez, M. E. (2001). Mathematical Modeling of Impingement Drying of Corn Tortillas, *Journal of Food Engineering*, **50**(3): 121-128.

13. Reddy, J. N. (2002). *Energy Principles and Variational Methods in Applied Mechanics*, 2nd Edition, John Wiley & Sons, Inc., New York.

14. Reddy, J. N. (2004). *An Introduction to Nonlinear Finite Element Analysis*, Oxford University Press, New York.

15. Reddy, J. N. (2006). *An Introduction to the Finite Element Method*, 3rd Edition, McGraw Hill, New York.

16. Mills, A. F. (2001). *Mass Transfer*, 2nd Edition, Prentice Hall, Inc., Upper Saddle River, NJ.

17. Browning, C. E. (1976). *The Mechanisms of Elevated Temperature Property Losses in High Performance Structural Epoxy Resin Matrix Materials After Exposures to High Humidity Environments*, Ph.D. dissertation, School of Engineering, University of Dayton, Dayton, OH.

18. Maggana, C., Pissis, P. (1999). Water Sorption and Diffusion Studies in an Epoxy Resin System, *Journal of Polymer Science: Part B: Polymer Physics*, **37**(11): 1165-1182.

19. Vanlandingham, M. R., Eduljee, R. F., Gillespie, J. W. Jr. (1999). Moisture Diffusion in Epoxy Systems, *Journal of Applied Polymer Science*, **71**(5): 787-798

APPENDIX A

This appendix contains a Matlab computer program for solving transient one–dimensional linear heat transfer problems. It is applicable to flat plate analysis with forced convection boundary conditions (at z=0 and z=L).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Linear Element FE Code for Cartesian 1-D Transient Heat Transfer%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Clear Workspace and Command Window%
clear;
clc;

%Finite Element Parameters%
NE = 100;       %Number of elements in model%
Time = 100;     %Run time [s]%
DeltaT = 0.001; %Time step [s]%
NPLOT = 10;     %Number of temperature field solutions to be plotted%

%Physical Parameters of System%
t = 0.02;       %Thickness of plate [m]%
rho = 1200;     %Density of plate [kg/m^3]%
Cp = 1000;      %Specific heat of plate at constant temperature
[J(kg*K)]%
k = 0.16;       %Thermal conductivity [W/(m*K)]%

Ti = 25;        %Initial plate temperature [K]%
Tinf_b = 30;    %Farfield temperature on bottom side of plate [K]%
Tinf_t = 150;   %Farfield temperature on top side of plate [K]%

h_bot = 17.2304;%Conv. coef. on bottom side of plate [W/(m^2*K)]%
h_top = 12.2333;%Conv. coef. on top side of plate [W/(m^2*K)]%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%Finite Element Calculations%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Begin calculation timer%
tic

%Number of nodes in the model%
```

```
NN = NE + 1;

%Nodal coordinates%
z = zeros(NN,1);
for i=1:1:NN
    z(i,1) = t/2*(1 - cos(4*atan(1)/(NN-1)*(i-1)));
end

%Length of each element%
L = zeros(NE,1);
for i=1:1:NE
    L(i,1) = z(i+1,1) - z(i,1);
end

%Element connectivity matrix%
con1 = zeros(1,NE);
con2 = con1;
for j = 1:1:NE
    con1(1,j) = j;
    con2(1,j) = j + 1;
end
econ = [con1 ; con2];

%Local [K] matrices%
k_temp = [1 -1 ; -1 1];
K_local = zeros(2,2,NE);
for n = 1:1:NE
    for i = 1:1:2
        for j = 1:1:2
            K_local(i,j,n) = k/L(n,1)*k_temp(i,j);
        end
    end
end

%Global [K] matrix%
K = zeros(NN,NN);
for n = 1:1:NE
    for i = 1:1:2
        for j = 1:1:2
            ii = econ(i,n);
            jj = econ(j,n);
            K(ii,jj) = K(ii,jj) + K_local(i,j,n);
        end
    end
end
K(1,1) = K(1,1) + h_bot;
K(NN,NN) = K(NN,NN) + h_top;

%Local [C] matrices%
c_temp = [1/3 1/6 ; 1/6 1/3];
C_local = zeros(2,2,NE);
for n = 1:1:NE
```

```
    for i = 1:1:2
        for j = 1:1:2
            C_local(i,j,n) = rho*Cp*L(n,1)*c_temp(i,j);
        end
    end
end

%Global [C] matrix%
C = zeros(NN,NN);
for n = 1:1:NE
    for i = 1:1:2
        for j = 1:1:2
            ii = econ(i,n);
            jj = econ(j,n);
            C(ii,jj) = C(ii,jj) + C_local(i,j,n);
        end
    end
end


%%%%%%%%%%%%%%%Time Dependent [Time-step] Calculations%%%%%%%%%%%%%%%%%

%Alpha-Family of Approximation (Crank-Nicolson scheme)%
alpha = 0.5;

%Global [Beta] matrix%
Beta = zeros(NN,NN);
Beta = C + DeltaT*alpha*K;

%Inverse of global [Beta] matrix%
Beta_In = inv(Beta);

%Global [Gamma] matrix%
Gamma = zeros(NN,NN);
Gamma = C - DeltaT*(1 - alpha)*K;

%Global {f} matrix%
f = zeros(NN,1);
f(1,1) = DeltaT*alpha*h_bot*Tinf_b + DeltaT*(1 - alpha)*h_bot*Tinf_b;
f(NN,1) = DeltaT*alpha*h_top*Tinf_t + DeltaT*(1 - alpha)*h_top*Tinf_t;

%Temperature calculations%
T = Ti*ones(NN,1);
T_plot = zeros(NN,NPLOT);
for i = 1:1:ceil(Time/DeltaT)
    T = Beta_In*(Gamma*T + f);
    for j = 1:1:NPLOT
        if i == ceil(Time/DeltaT*(j/NPLOT))
            T_plot(:,j) = T;
        end
    end
end
```

```
T_plot = [Ti*ones(NN,1)' ; T_plot']';

%Temperature plots%
Plot_Times = zeros(NPLOT + 1,1);
for i = 1:1:NPLOT
    Plot_Times(i + 1,1) = (i/NPLOT)*Time;
end
Plot_Times
plot(z,T_plot,'b')

%End calculation timer%
toc
```

APPENDIX B

This appendix contains code for a Matlab computer program for solving nonlinear coupled heat and mass transfer problems for a vertical flat plat with free convection at one surface (z=0) and forced convection at the other surface (z=L).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% LINEAR ELEMENT FINITE ELEMENT CODE FOR 1-D CARTESIAN %%%%%%%%
%%%%%%%% NON-LINEAR COUPLED CONVECTIVE HEAT AND MASS TRANSFER %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Clear Workspace and Command Window%
clear;
clc;

%Finite Element Parameters%
NE = 100;           %Number of elements in model%
NN = NE + 1;        %Number of nodes in model%
Time_F = 900;       %Run time [s]%
STEPS = 10000;      %Total number of time steps%
ERROR = 10^(-6);    %Allowable error between iterations%

%Physical Parameters of System%
t = 0.01;           %Thickness of plate [m]%
Lp = 0.06;          %Length of plate [m]%
H = 0.051;          %Distance from exit of nozzle to flat plate [m]%
d_h = 9.525*10^-3;  %Inner diameter of nozzle [m]%
r = 0.06;           %Radial distance for averaging of h [m]%

V_t = 9.935;        %Free stream velocity of air (top of plate) [m/s]%

rho_e = 1130;       %Mass density of epoxy [kg/m^3]%
rho_w = 125;        %Initial mass density of liquid water [kg/m^3]%
rho_w_b = 0;        %Free stream mass density of water vapor [kg/m^3]%
rho_w_t = 0;        %Free stream mass density of water vapor [kg/m^3]%

Cp_e = 1000;        %Specific heat of epoxy [J/(kg*K)]%
Cp_w = 4181;        %Specific heat of liquid water [J/(kg*K)]%
Cp_s = 1900;        %Specific heat of water vapor [J/(kg*K)]%

k_e = 0.16;         %Thermal conductivity of epoxy [W/(m*K)]%
k_w = 0.606;        %Thermal conductivity of water [W/(m*K)]%
```

```
Ti = 25;            %Initial plate temperature [C]%
Tinf_b = 25;        %Farfield temperature on bottom side of plate [C]%
Tinf_t = 90;        %Farfield temperature on top side of plate [C]%
g = 9.81;           %Gravitational constant [m/s^2]%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%Finite Element Calculations%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Initiate calculation timer%
tic

%Initial Conditions - Temperature and Partial Density of Water%
T_s = Ti*ones(NN,1);
rho_s = rho_w*ones(NN,1);
rho = rho_s;

%Nodal coordinates%
z = zeros(NN,1);
for i=1:1:NN
    z(i,1) = t/2*(1 - cos(4*atan(1)/(NN-1)*(i-1)));
end

%Length of each element%
L = zeros(NE,1);
for i=1:1:NE
    L(i,1) = z(i+1,1) - z(i,1);
end

%Element connectivity matrix%
con1 = zeros(1,NE);
con2 = con1;
for j = 1:1:NE
    con1(1,j) = j;
    con2(1,j) = j + 1;
end
econ = [con1 ; con2];

%Matrices for creating local [C] and [K]%
matrix_K = [1 -1 ; -1 1];
matrix_C = [1/3 1/6 ; 1/6 1/3];

%Initialize temperature plotting matrix%
T_plot = zeros(NN,STEPS);

%Initialize heat flux quantities%
Heat_Flux_Plate = zeros(STEPS,1);
Heat_Flux_Fluid = zeros(STEPS,1);
Heat_Flux_Convection = zeros(STEPS,1);
Heat_Flux_Evaporation = zeros(STEPS,1);
```

```
%Zero out time step data output%
STEP_SIZE = zeros(STEPS,1);
TIME = zeros(STEPS,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%Time Step Calculations%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for ALPHA = 1:1:STEPS

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%% Time Step t = ts %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Time step%
    DeltaT = (Time_F + 1)^((ALPHA+1)/STEPS) - (Time_F +
1)^(ALPHA/STEPS);
    STEP_SIZE(ALPHA,1) = DeltaT;
    TIME(ALPHA,1) = (Time_F + 1)^(ALPHA/STEPS) - 1;

    %Time derivative of partial water density vector (Finite
difference)%
    rho_dot = (rho_s - rho)/DeltaT;

    %Time derivative of partial water density of each element%
    rho_dot_elem = zeros(NE,1);
    for i = 1:1:NE
        rho_dot_elem(i,1) = 1/2*(rho_dot(i,1) + rho_dot(i+1,1));
    end

    %Temperature at beginning of time step%
    T = T_s;
    rho = rho_s;

    %Film temperatures of air%
    T_f_b = 1/2*(Tinf_b + T(1,1));
    T_f_t = 1/2*(Tinf_t + T(NN,1));

    %Thermodynamic properties of air evaluated at film temperatures%
    if T_f_b >= -23
        rho_a_b = abs(-0.000000000000079282051301l733*T_f_b^5 +
0.000000000095330116556368 6*T_f_b^4 - 0.000000050163723735009*T_f_b^3 +
0.000016847201636927 6*T_f_b^2 - 0.00470217379842505*T_f_b +
1.27700146883161);
        Cp_b = abs(-0.000000000000088888902465953*T_f_b^6 +
0.000000000099528207068 4165*T_f_b^5 - 0.000000042891658 7677406*T_f_b^4
+ 0.000008231940895321 04*T_f_b^3 - 0.000200457865622728*T_f_b^2 +
0.0152668183846889*T_f_b + 1006.57290237493);
        nu_b = abs(-0.000000000000000000410256409748*T_f_b^5 +
0.000000000000000031179487194130 6*T_f_b^4 -
0.000000000000132687738890731*T_f_b^3 +
```

```
0.000000000120357440778083*T_f_b^2 + 0.000000088726868752095*T_f_b +
0.0000134143539102198);
        k_therm_b = abs(0.0000000000000018461538460742 4*T_f_b^5 -
0.000000000001463682983607 35*T_f_b^4 +
0.000000000366064521972289*T_f_b^3 - 0.000000057872362103955 2*T_f_b^2 +
0.0000793825265292941*T_f_b + 0.0241661248040929);
        Pr_b = abs(0.000000000000001377777781502 72*T_f_b^6 -
0.000000000001282174362774 89*T_f_b^5 +
0.000000000433497624824249*T_f_b^4 - 0.000000063108056562291 1*T_f_b^3 +
0.0000382756694647818*T_f_b^2 - 0.00023382828946905*T_f_b +
0.711681203873734);
    else
        rho_a_b = 1.3947;
        Cp_b = 1006;
        nu_b = 1.144*10^-5;
        k_therm_b = 0.0223;
        Pr_b = 0.720;
    end

    if T_f_t >= -23
        rho_a_t = abs(-0.000000000000079282051301173 3*T_f_t^5 +
0.000000000095330116556368 6*T_f_t^4 - 0.000000050163723735009*T_f_t^3 +
0.0001684720163692 76*T_f_t^2 - 0.00470217379842505*T_f_t +
1.27700146883161);
        Cp_t = abs(-0.000000000000008888889024659 53*T_f_t^6 +
0.000000000995282070684165*T_f_t^5 - 0.000000042891658767740 6*T_f_t^4
+ 0.000008231940895321 04*T_f_t^3 - 0.000200457865622728*T_f_t^2 +
0.0152668183846889*T_f_t + 1006.57290237493);
        nu_t = abs(-0.000000000000000000041025640974 8*T_f_t^5 +
0.000000000000000311794871941306*T_f_t^4 -
0.000000000000132687738890731*T_f_t^3 +
0.000000000120357440778083*T_f_t^2 + 0.000000088726868752095*T_f_t +
0.0000134143539102198);
        k_therm_t = abs(0.0000000000000001846153846074 24*T_f_t^5 -
0.000000000001463682983607 35*T_f_t^4 +
0.000000000366064521972289*T_f_t^3 - 0.000000057872362103955 2*T_f_t^2 +
0.0000793825265292941*T_f_t + 0.0241661248040929);
        Pr_t = abs(0.000000000000001377777781502 72*T_f_t^6 -
0.000000000001282174362774 89*T_f_t^5 +
0.000000000433497624824249*T_f_t^4 - 0.000000063108056562291 1*T_f_t^3 +
0.0000382756694647818*T_f_t^2 - 0.00023382828946905*T_f_t +
0.711681203873734);
    else
        rho_a_t = 1.3947;
        Cp_t = 1006;
        nu_t = 1.144*10^-5;
        k_therm_t = 0.0223;
        Pr_t = 0.720;
    end

    %Saturated water (latent heat and vapor density%
    if T(1,1) >= 0.01
```

```
        h_fg_b = abs(0.000000000275007280836626*T(1,1)^6 -
0.000000292752383970474*T(1,1)^5 + 0.000076712800163433*T(1,1)^4 -
0.0211024525112885*T(1,1)^3 + 0.900861539092667*T(1,1)^2 -
2372.60570828709*T(1,1)+ 2501365.48344311);
        rho_sat_b = abs(0.000000000000016976176180498343*T(1,1)^6 -
0.000000000001803126775305 23*T(1,1)^5 +
0.000000004323455857058 31*T(1,1)^4 - 0.000000045823891003591 2*T(1,1)^3
+ 0.000018577021457836 5*T(1,1)^2 + 0.000213313865582776*T(1,1) +
0.00515858825086024);
    else
        h_fg_b = 2501300;
        rho_sat_b = 0.004851;
    end

    if T(NN,1) >= 0.01
        h_fg_t = abs(0.000000000275007280836626*T(NN,1)^6 -
0.000000292752383970474*T(NN,1)^5 + 0.000076712800163433*T(NN,1)^4 -
0.0211024525112885*T(NN,1)^3 + 0.900861539092667*T(NN,1)^2 -
2372.60570828709*T(NN,1)+ 2501365.48344311);
        rho_sat_t = abs(0.000000000000016976176180498343*T(NN,1)^6 -
0.000000000001803126775305 23*T(NN,1)^5 +
0.000000004323455857058 31*T(NN,1)^4 -
0.000000045823891003591 2*T(NN,1)^3 + 0.000018577021457836 5*T(NN,1)^2 +
0.000213313865582776*T(NN,1) + 0.00515858825086024);
    else
        h_fg_t = 2501300;
        rho_sat_t = 0.004851;
    end

    %Diffusion coefficients of water vapor in air%
    if T_f_b >= 7
        D_a_b = abs(1.87*10^-10*(abs(T_f_b + 273))^2.072);
    else
        D_a_b = 2.19965*(10^-05);
    end

    if T_f_t >= 7
        D_a_t = abs(1.87*10^-10*(abs(T_f_t + 273))^2.072);
    else
        D_a_t = 2.19965*(10^-05);
    end

    %%%%%%%%%%%%%%%Convective heat transfer coefficients%%%%%%%%%%%%%%%
    Ra_L = g*(1/(T_f_b + 273))*abs(T(1,1) -
Tinf_b)*Lp^3/(nu_b*(k_therm_b/(rho_a_b*Cp_b)));
    Nu_L = 0.68 + 0.670*Ra_L^(1/4)/(1 + (0.492/Pr_b)^(9/16))^(4/9);

    Re_t = V_t*d_h/nu_t;

    h_bot = Nu_L*k_therm_b/Lp;
    h_top = (2*k_therm_t*Re_t^(0.5)*Pr_t^(0.42)*(1 - 1.1*d_h/r)*(1 +
0.005*Re_t^(0.55))^(0.5))/(r*(1 + 0.1*(H/d_h - 6)*d_h/r));
```

```
    %%%%%%%%%%%%%%%Convective mass transfer coefficients%%%%%%%%%%%%%%%
    h_m_bot =
h_bot/(rho_a_b*Cp_b*(k_therm_b/(rho_a_b*Cp_b*D_a_b))^(2/3));
    h_m_top =
h_top/(rho_a_t*Cp_t*(k_therm_t/(rho_a_t*Cp_t*D_a_t))^(0.58));

    %Mass density of water of each element%
    rho_w_elem = zeros(NE,1);
    for i=1:1:NE
        rho_w_elem(i,1) = 1/2*(rho(i,1) + rho(i+1,1));
    end

    %Mass density of each element%
    rho_elem = zeros(NE,1);
    for i=1:1:NE
        rho_elem(i,1) = rho_w_elem(i,1) + rho_e;
    end

    %Thermal conductivity of each element%
    k_therm_elem = zeros(NE,1);
    for i = 1:1:NE
        k_therm_elem(i,1) = 1/rho_elem(i,1)*(rho_e*k_e +
rho_w_elem(i,1)*k_w);
    end

    %Specific heat of each element%
    Cp_elem = zeros(NE,1);
    for i = 1:1:NE
        Cp_elem(i,1) = 1/rho_elem(i,1)*(rho_e*Cp_e +
rho_w_elem(i,1)*Cp_w);
    end

    %Mass diffusion coefficient of each element%
    D_n = zeros(NN,1);
    for i = 1:1:NN
        if T(i,1) >= 37.8
            D_n(i,1) = abs(0.526913365598052*exp(-
9177.934261262040000000000000000000*(1/(T(i,1) + 273))));
        else
            D_n(i,1) = 7.93*10^-14;
        end
    end
    D_elem = zeros(NE,1);
    for i = 1:1:NE
        D_elem(i,1) = 1/2*(D_n(i,1) + D_n(i+1,1));
    end

    %%%%%%%%%%%%%%%Construction of Heat Transfer Matrices%%%%%%%%%%%%%%%

    %Local [K] matrices%
    K_h_local = zeros(2,2,NE);
```

```
    for k = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
            K_h_local(i,j,k) = k_therm_elem(k,1)/L(k,1)*matrix_K(i,j);
            end
        end
    end
    for k = 1:1:NE
        K_h_local(1,1,k) = -Cp_w*D_elem(k,1)*(rho(k+1,1) -
rho(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(3*rho_dot(k,1) + rho_dot(k+1,1))
+ K_h_local(1,1,k);
        K_h_local(1,2,k) = -Cp_w*D_elem(k,1)*(rho(k+1,1) -
rho(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(rho_dot(k,1) + rho_dot(k+1,1)) +
K_h_local(1,2,k);
        K_h_local(2,1,k) = Cp_w*D_elem(k,1)*(rho(k+1,1) -
rho(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(rho_dot(k,1) + rho_dot(k+1,1)) +
K_h_local(2,1,k);
        K_h_local(2,2,k) = Cp_w*D_elem(k,1)*(rho(k+1,1) -
rho(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(rho_dot(k,1) + 3*rho_dot(k+1,1))
+ K_h_local(2,2,k);
    end

    %Global [K] matrix%
    K_h = zeros(NN,NN);
    for n = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                ii = econ(i,n);
                jj = econ(j,n);
                K_h(ii,jj) = K_h(ii,jj) + K_h_local(i,j,n);
            end
        end
    end

    K_h(1,1) = K_h(1,1) + h_m_bot*(Cp_w -
Cp_s)*(rho_sat_b*rho(1,1)/(rho_e + rho(1,1)) - rho_w_b) + h_bot;
    K_h(NN,NN) = K_h(NN,NN) + h_m_top*(Cp_w -
Cp_s)*(rho_sat_t*rho(NN,1)/(rho_e + rho(NN,1)) - rho_w_t) + h_top;

    %Local [C] matrices%
    C_h_local = zeros(2,2,NE);
    for k = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                C_h_local(i,j,k) =
rho_elem(k,1)*Cp_elem(k,1)*L(k,1)*matrix_C(i,j);
            end
        end
    end
    for k = 1:1:NE
        C_h_local(1,1,k) = Cp_w*L(k,1)/12*(3*rho(k,1) + rho(k+1,1)) +
C_h_local(1,1,k);
```

```
        C_h_local(1,2,k) = Cp_w*L(k,1)/12*(rho(k,1) + rho(k+1,1)) +
C_h_local(1,2,k);
        C_h_local(2,1,k) = Cp_w*L(k,1)/12*(rho(k,1) + rho(k+1,1)) +
C_h_local(2,1,k);
        C_h_local(2,2,k) = Cp_w*L(k,1)/12*(rho(k,1) + 3*rho(k+1,1)) +
C_h_local(2,2,k);
    end

    %Global [C] matrix%
    C_h = zeros(NN,NN);
    for n = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                ii = econ(i,n);
                jj = econ(j,n);
                C_h(ii,jj) = C_h(ii,jj) + C_h_local(i,j,n);
            end
        end
    end

    %Global {f} vector%
    f_h = zeros(NN,1);
    f_h(1,1) = -h_m_bot*(h_fg_b +
Cp_s*Tinf_b)*(rho_sat_b*rho(1,1)/(rho_e + rho(1,1)) - rho_w_b) +
h_bot*Tinf_b;
    f_h(NN,1) = -h_m_top*(h_fg_t +
Cp_s*Tinf_t)*(rho_sat_t*rho(NN,1)/(rho_e + rho(NN,1)) - rho_w_t) +
h_top*Tinf_t;

    %%%%%%%%%%%%%%%Construction of Mass Transfer Matrices%%%%%%%%%%%%%%%

    %Local [K] matrices%
    K_m_local = zeros(2,2,NE);
    for k = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                K_m_local(i,j,k) = (D_elem(k,1)/L(k,1))*matrix_K(i,j);
            end
        end
    end

    %Global [K] matrix%
    K_m = zeros(NN,NN);
    for n = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                ii = econ(i,n);
                jj = econ(j,n);
                K_m(ii,jj) = K_m(ii,jj) + K_m_local(i,j,n);
            end
        end
    end
    K_m(1,1) = K_m(1,1) + h_m_bot*rho_sat_b/(rho_e + rho(1,1));
```

```
    K_m(NN,NN) = K_m(NN,NN) + h_m_top*rho_sat_t/(rho_e + rho(NN,1));

    %Local [C] matrices%
    C_m_local = zeros(2,2,NE);
    for k = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                C_m_local(i,j,k) = L(k,1)*matrix_C(i,j);
            end
        end
    end

    %Global [C] matrix%
    C_m = zeros(NN,NN);
    for n = 1:1:NE
        for i = 1:1:2
            for j = 1:1:2
                ii = econ(i,n);
                jj = econ(j,n);
                C_m(ii,jj) = C_m(ii,jj) + C_m_local(i,j,n);
            end
        end
    end

    %Global {f} vector%
    f_m = zeros(NN,1);
    f_m(1,1) = h_m_bot*rho_w_b;
    f_m(NN,1) = h_m_top*rho_w_t;

    %Iteration criteria%
    T_s = 1.01*T;
    rho_s = 1.01*rho;
    ER = 1;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%Iterative Calculations%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%% Time Step t = ts+1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while ERROR < ER

        %Vectors for iteration criteria%
        T_temp = T_s;
        rho_temp = rho_s;

        %Film temperatures of air%
        T_f_b_s = 1/2*(Tinf_b + T_s(1,1));
        T_f_t_s = 1/2*(Tinf_t + T_s(NN,1));

        %Thermodynamic properties of air evaluated at film temps%
        if T_f_b_s >= -23
            rho_a_b_s = abs(-0.000000000000079282051301173 3*T_f_b_s^5 +
0.000000000095330116556368 6*T_f_b_s^4 -
```

```
0.000000050163723735009*T_f_b_s^3 + 0.00001684720163692 76*T_f_b_s^2 -
0.00470217379842505*T_f_b_s + 1.27700146883161);
            Cp_b_s = abs(-0.000000000000088888902465953*T_f_b_s^6 +
0.000000000099528207068 4165*T_f_b_s^5 -
0.000000042891658767 7406*T_f_b_s^4 + 0.000008231940895321 04*T_f_b_s^3 -
0.00020045786562272 8*T_f_b_s^2 + 0.0152668183846889*T_f_b_s +
1006.57290237493);
            nu_b_s = abs(-0.000000000000000000410256409748*T_f_b_s^5 +
0.000000000000000311794871941306*T_f_b_s^4 -
0.000000000000132687738890731*T_f_b_s^3 +
0.000000000120357440778083*T_f_b_s^2 + 0.000000088726868752095*T_f_b_s
+ 0.0000134143539102198);
            k_therm_b_s = abs(0.000000000000001846153846074 24*T_f_b_s^5
- 0.000000000001463682983607 35*T_f_b_s^4 +
0.000000000366064521972289*T_f_b_s^3 -
0.000000057872362103955 2*T_f_b_s^2 + 0.000079382526529294 1*T_f_b_s +
0.0241661248040929);
            Pr_b_s = abs(0.000000000000001377777781502 72*T_f_b_s^6 -
0.000000000001282174362774 89*T_f_b_s^5 +
0.000000000433497624824249*T_f_b_s^4 -
0.000000063108056562291 1*T_f_b_s^3 + 0.000003827566946478 18*T_f_b_s^2 -
0.00023382828946905*T_f_b_s + 0.711681203873734);
        else
            rho_a_b_s = 1.3947;
            Cp_b_s = 1006;
            nu_b_s = 1.144*10^-5;
            k_therm_b_s = 0.0223;
            Pr_b_s = 0.720;
        end

        if T_f_t_s >= -23
            rho_a_t_s = abs(-0.000000000000079282051301173 3*T_f_t_s^5 +
0.000000000095330116556368 6*T_f_t_s^4 -
0.000000050163723735009*T_f_t_s^3 + 0.00001684720163692 76*T_f_t_s^2 -
0.00470217379842505*T_f_t_s + 1.27700146883161);
            Cp_t_s = abs(-0.000000000000088888902465953*T_f_t_s^6 +
0.000000000099528207068 4165*T_f_t_s^5 -
0.000000042891658767 7406*T_f_t_s^4 + 0.000008231940895321 04*T_f_t_s^3 -
0.00020045786562272 8*T_f_t_s^2 + 0.0152668183846889*T_f_t_s +
1006.57290237493);
            nu_t_s = abs(-0.000000000000000000410256409748*T_f_t_s^5 +
0.000000000000000311794871941306*T_f_t_s^4 -
0.000000000000132687738890731*T_f_t_s^3 +
0.000000000120357440778083*T_f_t_s^2 + 0.000000088726868752095*T_f_t_s
+ 0.0000134143539102198);
            k_therm_t_s = abs(0.000000000000001846153846074 24*T_f_t_s^5
- 0.000000000001463682983607 35*T_f_t_s^4 +
0.000000000366064521972289*T_f_t_s^3 -
0.000000057872362103955 2*T_f_t_s^2 + 0.000079382526529294 1*T_f_t_s +
0.0241661248040929);
            Pr_t_s = abs(0.000000000000001377777781502 72*T_f_t_s^6 -
0.000000000001282174362774 89*T_f_t_s^5 +
0.000000000433497624824249*T_f_t_s^4 -
```

```
0.000000063108056562291l*T_f_t_s^3 + 0.00000382756694647818*T_f_t_s^2 -
0.00023382828946905*T_f_t_s + 0.711681203873734);
        else
            rho_a_t_s = 1.3947;
            Cp_t_s = 1006;
            nu_t_s = 1.144*10^-5;
            k_therm_t_s = 0.0223;
            Pr_t_s = 0.720;
        end

        %Saturated water (latent heat and vapor density%
        if T_s(1,1) >= 0.01
            h_fg_b_s = abs(0.000000000275007280836626*T_s(1,1)^6 -
0.00000029275238397047 4*T_s(1,1)^5 + 0.000076712800163433*T_s(1,1)^4 -
0.0211024525112885*T_s(1,1)^3 + 0.900861539092667*T_s(1,1)^2 -
2372.60570828709*T_s(1,1)+ 2501365.48344311);
            rho_sat_b_s = abs(0.000000000000016977618049 8343*T_s(1,1)^6
- 0.000000000001803126775305 23*T_s(1,1)^5 +
0.000000043234558570583 1*T_s(1,1)^4 -
0.000000458238910035912*T_s(1,1)^3 + 0.000185770214578365*T_s(1,1)^2
+ 0.00213313865582776*T_s(1,1) + 0.00515858825086024);
        else
            h_fg_b_s = 2501300;
            rho_sat_b_s = 0.004851;
        end

        if T_s(NN,1) >= 0.01
            h_fg_t_s = abs(0.000000000275007280836626*T_s(NN,1)^6 -
0.00000029275238397047 4*T_s(NN,1)^5 + 0.000076712800163433*T_s(NN,1)^4
- 0.0211024525112885*T_s(NN,1)^3 + 0.900861539092667*T_s(NN,1)^2 -
2372.60570828709*T_s(NN,1)+ 2501365.48344311);
            rho_sat_t_s =
abs(0.000000000000016977618049 8343*T_s(NN,1)^6 -
0.000000000001803126775305 23*T_s(NN,1)^5 +
0.000000043234558570583 1*T_s(NN,1)^4 -
0.000000458238910035912*T_s(NN,1)^3 +
0.000185770214578365*T_s(NN,1)^2 + 0.00213313865582776*T_s(NN,1) +
0.00515858825086024);
        else
            h_fg_t_s = 2501300;
            rho_sat_t_s = 0.004851;
        end

        %Diffusion coefficients of water vapor in air%
        if T_f_b_s >= 7
            D_a_b_s = abs(1.87*10^-10*(abs(T_f_b_s + 273))^2.072);
        else
            D_a_b_s = 2.19965*(10^-05);
        end

        if T_f_t_s >= 7
            D_a_t_s = abs(1.87*10^-10*(abs(T_f_t_s + 273))^2.072);
        else
```

```
            D_a_t_s = 2.19965*(10^-05);
        end

        %%%%%%%%%%%%%Convective heat transfer coefficients%%%%%%%%%%%%%
        Ra_L_s = g*(1/(T_f_b_s + 273))*abs(T_s(1,1) -
Tinf_b)*Lp^3/(nu_b_s*(k_therm_b_s/(rho_a_b_s*Cp_b_s)));
        Nu_L_s = 0.68 + 0.670*Ra_L_s^(1/4)/(1 +
(0.492/Pr_b_s)^(9/16))^(4/9);

        Re_t_s = V_t*d_h/nu_t_s;

        h_bot_s = Nu_L_s*k_therm_b_s/Lp;
        h_top_s = (2*k_therm_t_s*Re_t_s^(0.5)*Pr_t_s^(0.42)*(1 -
1.1*d_h/r)*(1 + 0.005*Re_t_s^(0.55))^(0.5))/(r*(1 + 0.1*(H/d_h -
6)*d_h/r));

        %%%%%%%%%%%%%Convective mass transfer coefficients%%%%%%%%%%%%%
        h_m_bot_s =
h_bot_s/(rho_a_b_s*Cp_b_s*(k_therm_b_s/(rho_a_b_s*Cp_b_s*D_a_b_s))^(2/3
));
        h_m_top_s =
h_top_s/(rho_a_t_s*Cp_t_s*(k_therm_t_s/(rho_a_t_s*Cp_t_s*D_a_t_s))^(0.5
8));

        %Mass density of water of each element%
        rho_w_elem_s = zeros(NE,1);
        for i=1:1:NE
            rho_w_elem_s(i,1) = 1/2*(rho_s(i,1) + rho_s(i+1,1));
        end

        %Mass density of each element%
        rho_elem_s = zeros(NE,1);
        for i=1:1:NE
            rho_elem_s(i,1) = rho_w_elem_s(i,1) + rho_e;
        end

        %Thermal conductivity of each element%
        k_therm_elem_s = zeros(NE,1);
        for i = 1:1:NE
            k_therm_elem_s(i,1) = 1/rho_elem_s(i,1)*(rho_e*k_e +
rho_w_elem_s(i,1)*k_w);
        end

        %Specific heat of each element%
        Cp_elem_s = zeros(NE,1);
        for i = 1:1:NE
            Cp_elem_s(i,1) = 1/rho_elem_s(i,1)*(rho_e*Cp_e +
rho_w_elem_s(i,1)*Cp_w);
        end

        %Mass diffusion coefficient of each element%
        D_n_s = zeros(NN,1);
```

```
        for i = 1:1:NN
            if T_s(i,1) > 37.8
                D_n_s(i,1) = abs(0.526913365598052*exp(-
9177.9342612620400000000000000000000*(1/(T_s(i,1) + 273))));
            else
                D_n_s(i,1) = 7.93*10^-14;
            end
        end

        D_elem_s = zeros(NE,1);
        for i = 1:1:NE
            D_elem_s(i,1) = 1/2*(D_n_s(i,1) + D_n_s(i+1,1));
        end

        %%%%%%%%%%%%Construction of Heat Transfer Matrices%%%%%%%%%%%%%
        %Local [K] matrices%
        K_h_local_s = zeros(2,2,NE);
        for k = 1:1:NE
            for i = 1:1:2
                for j = 1:1:2
                K_h_local_s(i,j,k) =
(k_therm_elem_s(k,1)/L(k,1))*matrix_K(i,j);
                end
            end
        end
        for k =1:1:NE
            K_h_local_s(1,1,k) = -Cp_w*D_elem_s(k,1)*(rho_s(k+1,1) -
rho_s(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(3*rho_dot(k,1) +
rho_dot(k+1,1)) + K_h_local_s(1,1,k);
            K_h_local_s(1,2,k) = -Cp_w*D_elem_s(k,1)*(rho_s(k+1,1) -
rho_s(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(rho_dot(k,1) + rho_dot(k+1,1))
+ K_h_local_s(1,2,k);
            K_h_local_s(2,1,k) = Cp_w*D_elem_s(k,1)*(rho_s(k+1,1) -
rho_s(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(rho_dot(k,1) + rho_dot(k+1,1))
+ K_h_local_s(2,1,k);
            K_h_local_s(2,2,k) = Cp_w*D_elem_s(k,1)*(rho_s(k+1,1) -
rho_s(k,1))/(2*L(k,1)) + Cp_w*L(k,1)/12*(rho_dot(k,1) +
3*rho_dot(k+1,1)) + K_h_local_s(2,2,k);
        end

        %Global [K] matrix%
        K_h_s = zeros(NN,NN);
        for n = 1:1:NE
            for i = 1:1:2
                for j = 1:1:2
                    ii = econ(i,n);
                    jj = econ(j,n);
                    K_h_s(ii,jj) = K_h_s(ii,jj) + K_h_local_s(i,j,n);
                end
            end
        end
```

```
        K_h_s(1,1) = K_h_s(1,1) + h_m_bot_s*(Cp_w -
Cp_s)*(rho_sat_b_s*rho_s(1,1)/(rho_e + rho_s(1,1)) - rho_w_b) +
h_bot_s;
        K_h_s(NN,NN) = K_h_s(NN,NN) + h_m_top_s*(Cp_w -
Cp_s)*(rho_sat_t_s*rho_s(NN,1)/(rho_e + rho_s(NN,1)) - rho_w_t) +
h_top_s;

        %Local [C] matrices%
        C_h_local_s = zeros(2,2,NE);
        for k = 1:1:NE
            for i = 1:1:2
                for j = 1:1:2
                    C_h_local_s(i,j,k) =
rho_elem_s(k,1)*Cp_elem_s(k,1)*L(k,1)*matrix_C(i,j);
                end
            end
        end
        for k = 1:1:NE
            C_h_local_s(1,1,k) = Cp_w*L(k,1)/12*(3*rho_s(k,1) +
rho_s(k+1,1)) + C_h_local_s(1,1,k);
            C_h_local_s(1,2,k) = Cp_w*L(k,1)/12*(rho_s(k,1) +
rho_s(k+1,1)) + C_h_local_s(1,2,k);
            C_h_local_s(2,1,k) = Cp_w*L(k,1)/12*(rho_s(k,1) +
rho_s(k+1,1)) + C_h_local_s(2,1,k);
            C_h_local_s(2,2,k) = Cp_w*L(k,1)/12*(rho_s(k,1) +
3*rho_s(k+1,1)) + C_h_local_s(2,2,k);
        end

        %Global [C] matrix%
        C_h_s = zeros(NN,NN);
        for n = 1:1:NE
            for i = 1:1:2
                for j = 1:1:2
                    ii = econ(i,n);
                    jj = econ(j,n);
                    C_h_s(ii,jj) = C_h_s(ii,jj) + C_h_local_s(i,j,n);
                end
            end
        end

        %Global {f} vector%
        f_h_s = zeros(NN,1);
        f_h_s(1,1) = -h_m_bot_s*(h_fg_b_s +
Cp_s*Tinf_b)*(rho_sat_b_s*rho_s(1,1)/(rho_e + rho_s(1,1)) - rho_w_b) +
h_bot_s*Tinf_b;
        f_h_s(NN,1) = -h_m_top_s*(h_fg_t_s +
Cp_s*Tinf_t)*(rho_sat_t_s*rho_s(NN,1)/(rho_e + rho_s(NN,1)) - rho_w_t)
+ h_top_s*Tinf_t;

        %%%%%%%%%%%%%Construction of Mass Transfer Matrices%%%%%%%%%%%%

        %Local [K] matrices%
        K_m_local_s = zeros(2,2,NE);
```

```
        for k = 1:1:NE
            for i = 1:1:2
                for j = 1:1:2
                    K_m_local_s(i,j,k) =
(D_elem_s(k,1)/L(k,1))*matrix_K(i,j);
                end
            end
        end

        %Global [K] matrix%
        K_m_s = zeros(NN,NN);
        for n = 1:1:NE
            for i = 1:1:2
                for j = 1:1:2
                    ii = econ(i,n);
                    jj = econ(j,n);
                    K_m_s(ii,jj) = K_m_s(ii,jj) + K_m_local_s(i,j,n);
                end
            end
        end
        K_m_s(1,1) = K_m_s(1,1) + h_m_bot_s*rho_sat_b_s/(rho_e +
rho_s(1,1));
        K_m_s(NN,NN) = K_m_s(NN,NN) + h_m_top_s*rho_sat_t_s/(rho_e +
rho_s(NN,1));

        %The [C] matrix does not vary between time steps%

        %Global {f} vector%
        f_m_s = zeros(NN,1);
        f_m_s(1,1) = h_m_bot_s*rho_w_b;
        f_m_s(NN,1) = h_m_top_s*rho_w_t;

        %Alpha-family of approximation%
        alpha = 0.5;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%Heat Equation Solution At Time Step%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        K_h_1 = (1 - alpha)*C_h + alpha*C_h_s + DeltaT*alpha*K_h_s;
        K_h_2 = (1 - alpha)*C_h + alpha*C_h_s - DeltaT*(1 - alpha)*K_h;
        f_h_1 = K_h_2*T;
        f_h_2 = DeltaT*((1-alpha)*f_h + alpha*f_h_s);
        f_h_3 = f_h_1 + f_h_2;
        T_s = K_h_1\f_h_3;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%Mass Equation Solution At Time Step%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        K_m_1 = C_m + DeltaT*alpha*K_m_s;
        K_m_2 = C_m - DeltaT*(1 - alpha)*K_m;
        f_m_1 = K_m_2*rho;
        f_m_2 = DeltaT*((1 - alpha)*f_m + alpha*f_m_s);
```

```
        f_m_3 = f_m_1 + f_m_2;
        rho_s = K_m_1\f_m_3;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%Iteration Termination Criteria%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        ER_h_1 = 0;
        ER_h_2 = 0;
        ER_m_1 = 0;
        ER_m_2 = 0;
        for d=1:1:NN
            ER_h_1 = (T_s(d,1) - T_temp(d,1))^2 + ER_h_1;
            ER_h_2 = T_s(d,1)^2 + ER_h_2;
            ER_m_1 = (rho_s(d,1) - rho_temp(d,1))^2 + ER_m_1;
            ER_m_2 = rho_s(d,1)^2 + ER_m_2;
        end
        ER_h = (ER_h_1/ER_h_2)^(0.5);
        ER_m = (ER_m_1/ER_m_2)^(0.5);
        ER = max(ER_h,ER_m);

        %Output data%
        T_plot(:,ALPHA) = T_s;
        Heat_Flux_Plate(ALPHA,1) = -k_therm_elem_s(NE,1)*(T_s(NN,1) -
T_s(NN-1,1))/L(NE,1) - Cp_w*T(NN,1)*D_elem_s(NE,1)*(rho_s(NN,1) -
rho_s(NN-1,1))/L(NE,1);
        Heat_Flux_Fluid(ALPHA,1) = h_m_top*(Cp_w*T_s(NN,1) + h_fg_t_s +
Cp_s*(Tinf_t - T(NN)))*(rho_sat_t_s*rho_s(NN,1)/(rho_e + rho_s(NN,1)) -
rho_w_t) + h_top*(T(NN,1) - Tinf_t);
        Heat_Flux_Convection(ALPHA,1) = h_top*(T(NN,1) - Tinf_t);
        Heat_Flux_Evaporation(ALPHA,1) = h_m_top*(Cp_w*T_s(NN,1) +
h_fg_t_s + Cp_s*(Tinf_t - T(NN)))*(rho_sat_t_s*rho_s(NN,1)/(rho_e +
rho_s(NN,1)) - rho_w_t);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%% SOLUTION OUTPUT %%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Heat flux quantities%
Heat_Flux_Plate = [0 ; Heat_Flux_Plate];
Heat_Flux_Fluid = [0 ; Heat_Flux_Fluid];
Heat_Flux_Convection = [0 ; Heat_Flux_Convection];
Heat_Flux_Evaporation = [0 ; Heat_Flux_Evaporation];
Heat_Flux_Comparison = [Heat_Flux_Fluid' ; Heat_Flux_Convection' ;
Heat_Flux_Evaporation']';

%%%%%%%%%%%%%%%%%%%%Check solution for convergence%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%Flux values at interface should be nearly equal%%%%%%%%%%%%
Percent_Difference = zeros(STEPS+1,1);
for i = 2:1:(STEPS+1)
    Percent_Difference(i,1) = abs((Heat_Flux_Plate(i,1) -
Heat_Flux_Fluid(i,1))/Heat_Flux_Plate(i,1))*100;
```

```
end

%Nodal temperatures and time values at time steps%
T_plot = [Ti*ones(NN,1)' ; T_plot']';
TIME = [0 ; TIME];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLOTS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(z,T_plot);
plot(TIME,T_plot(NN,:)');
plot(TIME,Heat_Flux_Comparison);

%Terminate calculation timer%
toc
```

VITA

GREGORY STEVEN PAYETTE

PERMANENT ADDRESS:

Texas A&M University
Department of Mechanical Engineering
c/o Dr. Roger J. Morgan
3123 TAMU
College Station, TX 77843-3123

EDUCATIONAL BACKGROUND:

Master of Science, Mechanical Engineering. Texas A&M University, College Station, Texas, May 2006.
Bachelor of Science, Mechanical Engineering, University of Idaho, Moscow, Idaho, May 2004.

CERTIFICATION:

Engineer-in-Training Certificate, June 2004, Idaho, No. 5930.

WORK EXPERIENCE:

GDS Engineers, Baytown, TX 05/04-08/04
Staff Engineer – Pressure vessel rerate group
Stress analysis of pressure vessel components

GDS Engineers, Baytown, TX 05/03-08/03
Intern – Pressure vessel rerate group
Pressure vessel and piping analysis