EXPERIMENTAL AND NUMERICAL STUDY OF EVAPORATING FLOW HEAT TRANSFER IN MICRO-CHANNEL

By

HOKI LEE

A thesis submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Mechanical and Materials Engineering

December 2008

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of

HOKI LEE find it satisfactory and recommend that it be accepted.

_____

Chair

_____

_____

_____

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor Dr. Bob Richards for his knowledge, patience guidance and continual encouragement through my graduate work. This research could not have been done without a continuous source of guidance and helpful comments from Dr. Cill Richards, Dr. Dave Bahr, and Dr. Jeongmin Ahn in relation to the efficiency of evaporator, experimentation, and fabrication.

I am also thankful to Dr. D.J Morris and Tiffany Quy for training me with RTD circuitry design, chrome mask designs and fabrication procedures. I would like to thank Steve Brown, Dawn Findley and Joshah Jennings who have been great at their efforts to keep the cleanroom up and running. Finally, I would like to thank the entire MEMS group, both past and present, for helpful comments and being there. I am also grateful for the financial support provide by the DARPA and the NSF.

Experimental and Numerical Study of Evaporating Flow

Heat Transfer in Micro-Channel

Abstract

By HoKi Lee, Ph.D.
Washington State University
December 2008

Chair: Robert Richards

This dissertation presents the design of a MEMS-based micro-channel evaporator, fabricated and characterized to maximize the efficiency of evaporation. Efficiency of evaporator is defined to be the amount of energy used to evaporate fluid over the amount of energy input to the micro-channel evaporator. Experiment and numerical results are presented for steady and transient evaporation heat transfer from open top micro-channels.

The radial channels were fabricated in two geometries: one was a rectangular SU8 wick structure 40μm high and 5μm wide with channel widths range from 10 to 70μm. The second was a tapered 40μm high and 80μm wide at the outer radius and narrowing to 5μm wide at inner radius. The radial channels were fabricated on two membrane materials one was two-micron thick silicon and the other was the three hundred-nanometer silicon nitride. Transient state evaporation tests were done at cycle frequencies of 10 Hz, 20Hz and 50Hz.

An energy balance was experimentally determined on the radial micro-channel evaporators, including heat into the channels, conduction heat transfer radially along the channels and latent heat transfer through evaporation of the working fluid from the channels.

A numerical analysis was used to simulate the experimental measurements. The

numerical integration calculated conduction heat transfer using axisymmetric FDTD integration and mass transfer from evaporation using heat balance equation in the micro-channel.

The experimental and the numerical results were compared to validate the numerical model. The 40μm high and 5μm wide SU8 features with 70μm channels and the tapered channels were found to have the best overall performance of evaporators with silicon membrane. These wick dimensions yielded a mass evaporation rate of 6.5mg/min and 6.2mg/min, and wick efficiencies of 28% and 26% with an input energy of 34mW for steady state conditions. For transient state conditions of 10Hz 10% duty cycle, the tapered channel evaporator yielded a mass evaporation rate of 8.2mg/min and wick efficiency of 33.6% with an average input power of 34mw. Use of a low conductivity SiN membrane evaporator increased the mass evaporation rate to 8.2mg/min and the wick efficiency to 88% for an input energy of 13mW.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Thermal actuators are used in MEMS devices such as pumps, value and gear trains. The variety of applications has resulted in a diversity of thermal actuation methods. One of the thermal actuation methods uses forces generated by liquid-vapor phase changes (liquid-evaporation).

An application example for the thermal actuator is the P3 micro-engine under development at Washington State University (WSU). This device contains a cavity between two square membranes. The working fluid is heated and evaporates by using heat transfer from the lower (micro-channel evaporator) membrane. The evaporation causes increase of pressure inside of cavity and deforms the upper membrane. The work is produced by mechanical deformation.

This work is focused on design of micro-channel evaporators for use in phase-change actuators and engines. The goal of work was to build an effective design tool for the development of efficient micro-channel evaporator base on micro-channel with dimensions from ten to one hundred microns.

1.2 CONCEPT

A typical thermo-pneumatic actuator consists of a sealed cavity filled with a thermally expansive medium. A flexible diaphragm bound one side of the pressure cavity.

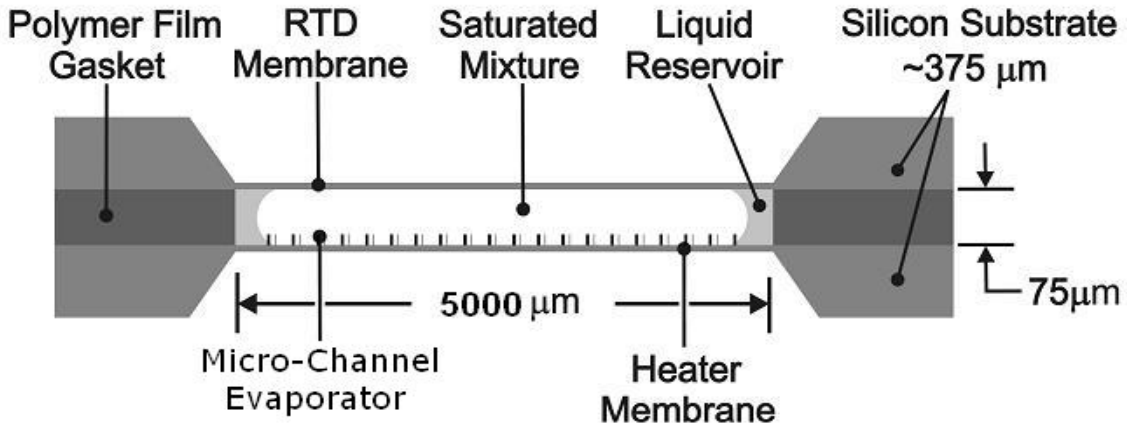


Figure 1.1 Schematic of actuator and a micro-channel evaporator

As a result of heating the chamber, chamber pressure increases through gas expansion or vaporization, displacing the diaphragm. The design of the thermal actuator motivating this is shown in Figure 1.1. The actuator consists of a cavity bound by two square membranes. The cavity is filled with working fluid, a two-phase mixture of 3M^{TM} Fluorinert FC77. The upper and lower membranes are fabricated from silicon or silicon nitride.

A thin platinum film of resistance heater is fabricated on the lower membrane to provide the heat source for the actuator. A capillary wicking structure, SU-8, is fabricated on the lower membrane to control the thickness of the working fluid layer over the heat addition region.

The micro-channel evaporator or micro-channels are used to increase the evaporation of the working. Use of the micro-channel on the lower membrane provides a means of controlling the thickness, thermal mass, and position of the working fluid relative to the heat source. The micro-channel is used to deliver the working fluid over the heater to promote more efficient phase change in the cavity. The goal of the present work was to

use a combination of both experimental and numerical work to provide insight into the controlling parameters and design constraints of the micro-channel evaporator.

1.3 LITERATURE REVIEW

Thermal actuators are often used in MEMS devices. One common thermal actuation method uses liquid-vapor phase change (liquid evaporation) to generate force. A key aspect of the design of a phase change actuator is the evaporator. To design an efficient evaporator for a thermal actuator it is necessary to understand the fluid flow and heat transfer from micro-channels. In this literature survey, we will first consider single-phase flow. We will review experimental heat transfer measurements in micro-channels. Next, we will review analytical solutions for single flow through micro-channels. Of great importance is the question of whether classical continuum theory applies in micro-channels. Then we will consider numerical solution to single phase flow and heat transfer in micro-channels.

Second, the literature survey is focus on the two-phase flow. We will review experimental heat transfer measurements in micro-channels. Next, we will review analytical solutions for two-phase through micro-channels, then we will consider numerical solution to two-phase flow and heat transfer in micro-channels.

Next, we will consider capillary action in the micro-channel. We will review analytical solutions for the capillary action. Then this literature survey will review the basic MEMS fabrication methods that include photolithography, wet etching, and chemical vapor deposition.

Finally, we will review the visualization techniques for the micro-channels.

1.3.1 HEAT TANSFER

Thermal actuators are used in MEMS devices, and one of the thermal actuation methods are uses liquid-vapor phase change (liquid-evaporation) to generates the forces. P. Bergstrom et al. and C. Rich et al have researched phase-change actuators in the steady-state condition that maximize the force and displacements produced with a minimum of energy used [1,2].

In the phase-change actuators, micro-channels can be used as an effective actuation method due to the potential of micro-channels for maintain high latent heat transfer rates [3] that causes a greater liquid-vapor volume change. Micro-channels have been shown to be very effective thermal management method in micro-scale deceives. One of the benefits of this heat transfer method is that the micro-channels can be included directly into the heat generating substrate, and it allows the contact resistance between the channel and substrate to be ignored [6]. Micro-channels are easy to fabricate and are compact, inexpensive, and maintain relatively high dissipation capacities [4,5].

For these reasons, micro-channels have been widely considered for heating and cooling in micro scale devices [6, 12] and serve to actuate these devices. To design the effective micro-channel devices, some of the major challenges faced in the study of micro-channel devices are in their heat transfer analysis. The study of heat transfer in micro-channels is still not completely understood, and a better understanding of heat transfer in micro-channels is needed in order to provide a rational basis for the design of these systems. The heat transfer in a micro-channel is dependent on physical properties, flow method, channel geometry, and heat flux. The following literature review exams these dependents to the micro-channels experimentally and numerically.

1.3.1.1 SINGLE-PHASE FLOW

The heat transfer from micro-channels has been a highly interesting area to many researchers since the early 1980's. The heat transfer of single-phase flows was explored in the initial research of micro-channels, and Y.P. Peles et al. and H.R. Chen et al. showed that the electronics could be cooled effectively through forced convective flow [3, 7].

T.M. Harms et al. presented experimental data (heat transfer coefficients and pressure drop) for single-phase forced convection in deep rectangular micro-channels. The channels were fabricated in a 2mm thick silicon substrate and covered a total projected area of 2.5cm by 2.5cm. All tests were performed with de-ionized water, and a critical Reynolds number for this laminar-turbulent transition was found to be 1500 [20].

Rahman presented new experimental measurements for pressure drops and heat transfer coefficients in a micro-channel heat sink in rectangular micro-channels on a silicon wafer. Channels of different depths, 180μm to 320μm, (or aspect ratios, 0.1 to 0.9) were studied [29].

The flow passage dimensions in convective heat transfer applications have been shifting towards smaller dimensions for the heat transfer enhancement, higher heat flux dissipation in micro-electronic devices, and the emergence of micro-scale devices that all require cooling. Employing smaller channel dimensions results in higher heat transfer performance, although it is accompanied by a higher pressure drop per unit length [21].

Wang and Peng [26] reported heat transfer experimental data for single-phase force convection of water and methanol in rectangular micro-channels. Channels with

hydraulic diameters between 311 and 747μm were tested, and results show that heat transfer characteristics cannot be described well with standard liquid forced convection correlations for rectangular ducts. However, Peng and Peterson did the heat transfer experimental investigation with single phase force convection in micro-channels that have hydraulic diameters between 133 and 367μm, and the results show that the channel aspect ratio has a very strong influence on the heat transfer coefficient.

Much research has been done in the analysis of micro-channels. Most research is focused on the heat transfer through the micro-channels. One of the most important steps in the heat transfer analysis is to verify the validity of the classical continuum macro theory on the micro-scale. Heat transfer analysis of micro-channels is important for validating modeling, numerical analysis, and design of MEMS devices. Many researchers have been attempting to apply macro-scale analysis to the micro-scale. There have been many arguments over whether the classical continuum model e.g. the conventional Navier-Stokes equations are still valid for micro-scale analysis [5, 6, 11, 12]. This argument is involving varying Reynolds numbers, working fluids, Nusselt numbers, flow geometries. This argument is important to determine a valid method for numerical analysis and modeling of both heat transfer and fluid motion through micro-channels. Kroeker [5] presented the classical correlations under prediction of the Nusselt values in micro-channels in his study. Owhaib [12] states in his study that the classical correlations still remain valid on the micro-scale (down to 50μm). In Tiselj's publication [11], the experiments have shown a Reynolds and Prandlt number dependency of the Nusselt number in laminar flow that strays from the classical theory. C.B. Sobhan et al. [6] maintained that the conventional theory was valid for micro-channels as small as 50μm

assuming that measurements were taken correctly and experimental conditions were identified and simulated correctly. Gad-el-Hat [33] argued that liquids such as water should be treated as continuous media (with classical theory being applicable) in channels larger than 1μm. Gao et al. [34] agreed that the classical laws were valid for gap sizes of 0.5mm or higher. Reviewed experiments seem to agree that the conventional Navier-Stokes and energy equations are valid at the micro-scale (larger than 1μm), and it seems the classical model can be used for micro-channel heat transfer and flow analysis.

Single-phase flow works for cooling and involves much less complicated physics in its modeling and analysis and agreement with the classical theory is expected to hold. The majority of the mathematical analysis and the numerical approaches use a very coarse mesh or simple 2-dimensional heat transfer model to determine the possible optimal structure for micro-channel devices, however three-dimensional heat transfer model also developed.

Kou et al. [39] presented a three-dimensional numerical model on the micro-channel heat sink to study the effects of heat transfer characteristics of various channel heights and widths. The study investigated the minimum thermal resistance of a heat sink including a micro-channel with a fixed dimension to search for its optimum channel width. The results show that a larger flow area, larger flow power, and shorter substrate thickness can result in lower thermal resistance. Kou [39] concluded the optimal channel width is not significantly influenced by a decrease in the channel height when the flow power is in the range of at 0.01Watt and 0.1Watt.

J.H. Ryu et al [36] presented a three-dimensional analysis for the thermal performance of a micro-channel heat sink and the optimal fin-channel shape that minimized the

thermal resistance. In this analysis a coolant passes through a number of micro-channels and convects heat away from a heat dissipating electronic component attached below. The flow is assumed to be laminar, incompressible, and hydro-dynamically fully developed. All thermo-physical properties were assumed constant. The channel depth, the channel width and the fin thickness were varied for 302μm-320μm, 44μm-50μm, and 50μm-56μm respectively. Among various design variables, the channel width appeared to be the most crucial quantity in dictating the performance of a micro-channel heat sink. The optimal dimension of micro channel was 45.3μm of width and 320μm, and corresponding thermal resistance was 0.069 [36].

Wen's [40] thermal resistance model is set up to study an optimum thermal design of the heat sink. The lowest total thermal resistance found was 0.054^{o}C Cm^{2}/W within a channel aspect ratio of 15, the channel width of 50μm, and the ratio of the fin width to the channel width was 0.65μm [40].

Another study by Li, et al [41] developed full three-dimensional numerical simulation (conduction heat transfer model) for heat transfer in silicon based micro-channel heat sink in order to optimize the geometric structure. Under a constant pumping power of 0.05W for a water-cooled micro-heat sink, the optimized geometric parameters of the structure as determined by the model were a channel width of 60μm and a channel depth of 700μm. The overall cooling capacity could be enhanced by more than 20% using the optimized spacing and channel dimensions, the overall thermal resistance was 0.068 ^{o}C/W for a pumping power of 2W [41].

K.K Ambatipudi and M.M. Rahman [30] developed a three-dimensional numerical model for conduction heat transfer in micro-channels. The numerical model was used to

investigate variation of the channel depth, channel width, the number of channels, and different flow rates through the channel. Heat transfer results compared reasonably well with experimental measurement [20, 29]. It was found that the Nusselt number is more for a system with a larger number of channels and larger Reynolds number. For Re=673, the optimum channel depth was 300μm. Also, results showed that the outlet temperature as well as the maximum temperature inside the solid decreases with Reynolds number because a larger mass of fluid is available to carry away the heat.

J. Li et al. [22] developed a three-dimensional numerical simulation for heat transfer in silicon based micro-channel heat sinks in order to optimize the geometric structure. J. Li et al. [22] performed a detailed numerical simulation for single-phase flow and found that the thermo-physical properties of the working fluid can also greatly affect the heat transfer characteristics of the device.

Shevade and Rahman [53] analyzed the convective heat transfer in micro-channels with rectangular and square cross sections for volumetric heat generation in the substrate. The conservation of mass and momentum equations were used. A thorough investigation for velocity and temperature distributions performed by varying hydraulic diameter (0.06~0.2cm), Reynolds number (Re=1600~3000), and heat generation rate in the substrate (g_o=6.4E8 W/m^3). The results showed that by increasing the Reynolds number, the outlet temperature decreases and an average Nusselt number increasings.

In 1995, Bailey et al [23] did the extensive review of the available cooling data for single-phase micro-channel flows, and showed that single-phase micro-channels can effectively cool miniature devices. However, L. Zhang et al. [24] showed that one disadvantage of single-phase flow is a large temperature gradient in the device from the

rise in coolant temperature compared to the two-phase flow that takes less pumping power to maintain a given thermal resistance.

1.3.1.2 TWO-PHASE FLOW

Two-phase flow is a more efficient heat dissipation method than single-phase flow [7,8]. Two-phase heat dissipation can achieve very high heat fluxes for a constant flow rate while maintaining a relatively constant surface temperature. The latent heat of vaporization of the fluid can be used in the two-phase flow to maximize the heat transfer rate.

Browers and Mudawar [25] performed an experimental study of boiling flow with a micro-channel (d=510μm) heat sink. This study showed that a high value of heat flux could be achieved with two-phase flow.

L. Zhang et al. [24] developed silicon test devices with nearly constant heat flux boundary conditions to study two-phase forced boiling convection in micro-channels with dimensions below 100μm. Heat transfer was optimized with hydraulic diameters between 25 and 65μm. The channel wall widths are below 350μm, which minimizes solid conduction and reduces variations in the heat flux boundary condition.

Two-phase flows involve more difficult physics in their modeling and analysis compared to the single-phase flows. However, due to the potential of micro-channels for maintaining a high heat dissipation rate, a better understanding of evaporative heat transfer in micro-channel is needed in order to provide a rational basis for the design of phase-change actuator. Like single-phase flows, much research has been done in the analysis of two-phase flow in micro-channels. Again, most research is focused on the

heat transfer through the micro-channels, and heat transfer analysis of micro-channels is important for validating modeling, numerical analysis, and design of MEMS devices. Most of recent research has been focused on the rectangular micro-channels. The fluid flow and the heat transfer characteristics of the channels could be changed by changing the channel geometry because of the aspect ratio of a rectangular channel influences capillary pumping power in the micro-channel [24].

Landerman [31] developed an analytical model for two-phase boiling heat transfer in a high aspect ratio rectangular channel, and the heat transfer and wall temperature was evaluated. In these high aspect ratio channels, the role of sub-cooled boiling was found to be insignificant relative to saturated boiling.

R.H. Nilson et al. [32] presented analytical solutions of an axially tapered micro-channel with a high aspect ratio for evaporative cooling devices and compared it with a rectangular micro-channel. The results demonstrate that tapered channels provide substantially better heat transfer capacity than straight channels of rectangular cross section, particularly under opposing gravitational forces.

S.W. Tchikanda et al. [43] derived analytical expressions for the mean velocity of a liquid flowing in an open rectangular micro-channel based on the Navier-Stokes equations. Solutions were decomposed into additive components driven by pressure gradients and by shear stresses on the liquid-vapor interface. Speed were computed for meniscus contact angles ranging from 0^o to 90^o, arbitrary channel aspect rations, and for wetting regimes from liquid-full to nearly-dry corner flows. These numerical results were used to guide the development of analytical expressions for the mean velocity.

Recently, some efforts have been made to analyze the possible optimal structure for micro-channel devices. Ryu [36] used a very coarse mesh (three dimensional) to determine the optimal dimension of the micro-channel heat sink. Hwang [35] used a simple 2-dimensional heat transfer model to determine the optimal dimensions of micro channel. Knight [37] even used a simple one-dimensional model as an approximation for the thermal conduction along the channel height. Koh et al. [38] used Darcy's law to describe the fluid flow.

Peles et al. [52] presented a quasi-one-dimensional model to investigated two-phase laminar flow in a heated capillary slot, driven by liquid evaporation from the interface.

The theoretical description of the phenomenon was based on the assumption of uniform distribution of hydrodynamic and thermal parameters over the cross-section of the liquid and vapor flows. With this approximation, the mass, thermal and momentum equations for the average parameters were obtained. The developed model allowed the estimation of the effects of capillary, inertia, frictional and gravity forces on the shape of the interface surface as well as the velocity and temperature distributions.

Scott [9, 10] used a 2-dimensional axisymmetric finite difference conduction model to study a liquid-vapor phase change membrane actuator, and it was found that the device efficiency was maximized when the energy input into the actuator was equal to the energy required to dry out the evaporator.

Browers and Mudawar [28] developed the mathematical model for the pressure drop in the micro-channels using Collier and Wallis homogenous equilibrium model. This homogenous equilibrium model assumes the liquid and vapor phases are a homogenous

mixture with uniform velocity, and properties were assumed to be uniform with each phase.

L. Zhang et al. [24] developed a homogeneous two-phase convection model with finite volume method. The results showed that optimum heat transfer occurred in rectangular channels with hydraulic diameters between 25 to 60μm. The two-phase micro-channel flow model yielded predictions in reasonable agreement with the measured pressure drop and wall temperature distribution.

Nilson et al. [32, 42] derived numerical and analytical solutions for steady evaporating flow in open micro-channels. Nilson et al. considered micro-channels of a rectangular cross section with uniform depth (500μm) and a width that decreases along the channel axis (tapered channel). The heat transfer results of tapered channel were compared to uniform rectangular channel analytical solution [42], and the results showed the tapered channels having better performance.

Wang et al. [45] presented investigation of an evaporating meniscus in a micro-channel using the Young-Laplace model (Capillary pressure calculation) and the kinetic theory-based expression for mass transport across a liquid-vapor interface. The results showed that thin layer of liquid had relatively larger mass evaporation than the thick layer of liquid.

Kim et al. [46] developed a mathematical model (based on force balance, mass conservation, and momentum equation) for heat and mass transfer in a miniature heat pipe with a grooved wick structure and solved analytically to yield the maximum heat transfer rate and the overall thermal resistance under steady-state conditions. The model is used for the thermal optimization of the grooved wick structure with respect to the

width and the groove height. The effects of the liquid-vapor interfacial shear stress, the contact angle, and the amount of initial liquid charge have been considered in the model. The calculation showed a narrow and deep groove had higher heat transport capability and higher overall thermal resistance.

Kyu Hyung et al. [47] developed a mathematical model for predicting the thermal performance of a flat micro heat pipe with a rectangular grooved wick structure. The axial variations of the wall temperature and the evaporation and condensation rates were considered by solving the one-dimensional conduction equation for the wall and the evaporation. Kyu Hyung et al. [47] showed that the maximum heat transport rate is 128W under the optimum conditions of groove width=0.1437mm and height=0.525mm for vapor temperature=90^{o}C, which reflects an enhancement of approximately 20% compared to the experimental result obtained by Hopkins et al [48]. Hopkins et al. [48] experimentally showed that a 120mm long flat heat pipe with an inner hydraulic diameter of 900μm had a temperature drop from the evaporator to the condenser end cap of 25^{o}C at a heat load of 100W. This showed that the axial variations of the wall temperature and the evaporation and condensation rates should be taken into account to accurately predict the thermal performance of a micro heat pipe.

Park and Lee [49,50] developed a mathematical model (conservation and force balance equation) that describes the two-phase flow and thermal characteristics of the extended meniscus region in an open micro-capillary channel. A working fluid in an open micro-capillary channel evaporates due to the applied heat, resulting in the thinner liquid film thickness and an increased in the radius of curvature at the liquid-vapor interface. The

results showed that the local heat transfer has an extremely large value in the thin film region.

A.J. Jiao et al. [51] developed a mathematical model predicting the effect of contact angle on the meniscus radius of thin film profile and heat flux distribution occurring in the micro heat pipe. This was done to have a better understanding of thin film evaporation and its behavior on the effective thermal conductivity in a grooved heat pipe and contribute to the optimize the design of conventional grooved heat pipes. The numerical results showed that while the capillary governs the heat transport capability in a heat pipe, the thin film evaporation determines the effective thermal conductivity in a pipe. The ratio of the heat transfer through the thin film region to the total heat transfer through the wall to the vapor phase decreased when the contact angle increased.

1.3.2 FLOW ANALYSIS: CAPILLARY ACTION

Capillary action or capillary motion is the tendency of a liquid to flow in narrow tubes or channels. Capillarity is a result of intermolecular attraction within the liquid and solid materials. Capillary action occurs by the axial gradient of capillary pressure, and capillary action can eliminate the need for active pumping. The capillary action is controlled by the surface tension, radius of the interface (meniscus), and wetting angle of the liquid on the surface of the capillary.

A great deal of work has been done on capillary pumping by researchers motivated by the optimization of heat pipes. One measure of the performance of a heat pipe is the maximum heat flux, the capillary limit or dry out point, supported by capillary pumping. An investigation of evaporation from single grooves is useful because the overall

performance of a heat pipe is a function of the capability of the individual grooves [54]. Catton and Stores [55] presented a one-dimensional, semi-analytical model for prediction of the wetted length supported by inclined triangular capillary grooves subject to heating from below. The model utilizes a macroscopic approach employing the concept of an apparent contact angle. The concept of accommodation theory is introduced to account for the change in the radius of curvature of the liquid-vapor interface between the liquid reservoir and the groove proper. The results showed a series of design curves that can be used to estimate the capillary limit in inclined heated triangular capillary grooves for a variety of operating conditions.

R.H. Nilson et al. [32] presented results of an analytical study showed that straight-rectangular micro-channels have a disadvantage in comparison to triangular grooves in capillary action. The capillary pressure in a rectangular channel varies with the liquid height in the channel only if the meniscus remains attached to the top corners of the channel. S.W. Tchikanda et al. [43] presented analytical and numerical solutions of the mean velocity of a liquid flowing in a rectangular micro-channel for contact angles ranging from 0^o to 90^o. The overall solutions are decomposed into two components. The first one is the flow driven by capillary forces, and the second component is the flow driven by pressure. The range of channel aspect ratios (width/height) was 0.65 to 2. The pressure-driven solutions apply equally well to flows driven by capillary or gravity forces as well as applied pressures.

R.H. Nilson et al. [42] presented relations between flow geometry and capillary pressure. They showed that a flatter entry meniscus, and hence a larger contact angle, increases the liquid pressure at the inlet, thereby providing a larger pressure difference to

drive the axial capillary flow. Therefore the contact angle between liquid and solid wall can range anywhere from 0 to 90 degree angle associated with the fluid to solid interface energy. They define a dead zone, a region in which the contact angle remains constant and the meniscus moves down the channel wall. In the dead zone there is no capillary pressure differential. Geometric constrains ensure that the fluid may be no deeper than half the channel width.

T.S. Sheu et al [56] presented a study that explores the effect of surface characteristics on capillary flow in micro-grooves. In this experimental study, they verified the theoretical model presented by Peterson and Ha [57] and their model can estimate the dryout location in the triangular grooves by the closed-form solution of a simplified governing equation. T.S. Sheu et al. [56] used a series of triangular micro-grooves with upper width (W) of 0.4mm and the vertex angle (α) of 60^{o} was machined on oxygen-free copper plate. The microgrooves plates include one with non-etched surface texture with lined incisions left by machining tool, and another with chemically etched surface texture with micro cavities. Methanol and ethanol were used as working fluid. The study showed that the etched micro-grooves surface texture with micro cavities has increased the capillary performance by 10-35% for the etched micro-grooves surfaces compared to the non-etched surface. The surface characteristic proved to have larger influences on the capillary performance of working fluid (Ethanol) with lower surface energy.

W.R. Jong, et al. [58] formulated a mathematical model of flow in micro-channel driven by capillary force and gravity from the Navaier-Stokes equations to predict flow time. The flow time is period time required for flow moves inlet to outlet. The results showed that when the micro-channel height is 150μm or smaller, the effects of gravity

force become less obvious; the capillary force becomes the dominant source to drive flow in the micro-channel.

Naoki et al. [59] did an extended analytical theory of the capillary rise problem for a rectangular micro-channel, and it was presented to study the interface motion driven by capillary action in a micro-channel. They examined experiments with glass rectangular tubes with sizes of about 50 to 100μm square and PDMS micro-channels with dimensions of 85 x 68μm and 75 x 45μm. They compared the experimental results with the theory and introduced dimensionless driving force variables (for glass-water, glass ethanol, and PDMS-ethanol) to predict for the interface motion for any size of rectangular channels.

D. Yan et al. [60] presented a theoretical model to analyze the finite-reservoir effect in a rectangular micro-channel to estimate the flow velocity profile. The proposed model has been verified by experiment using micro-PIV (Particle Image Velocimetry) technique. They found that the size of the channels cross section significantly affects the effective pumping period.

1.3.3 FABRICATION

Basic MEMS fabrication methods, photolithography, wet etching, and chemical vapor deposition (CVD) are used to fabricate most micro-channels [13]. The technique used depends on the micro-channel material and design, and common materials used to fabricate micro-channels are silicon, copper and SU8. The common problems and challenges associated with these fabrication methods include wet etching contamination,

thick film stress, covering channels without wafer bonding and obtaining an adequate channel depth [14].

The micro-channel materials are often desired to have a high thermal conductivity to be used as heat sinks, and both copper and silicon are appropriate material choice for this type of application. Copper channels can be fabricated through photolithography and etching, and silicon channels can be fabricated through a wet and dry etching process. Other fabrication techniques are being developed to aid in the heat transfer analysis of micro-fluidic channels.

One of other material choices to forming the micro-channel is SU8. SU8 is a photosensitive epoxy resist developed and patented by IBM in 1989. SU8 has become very important in micro-fabrication due to its thermal properties, thermal and chemical stability, and its ability to form high aspect ratio with high resolutions. SU8 can be stable up to 200^{o}C and decomposes near 340^{o}C, and forms single layers from 2-200μm thick. SU8 forms high aspect ratio up to 25:1 with standard UV lithography. SU8 is a unique micro fabrication material that is deposited in a relatively simple photolithography process. SU8 is a negative resist meaning that whatever material is exposed will stay on the substrate while the unexposed portion is removed during development. The basic fabrication process begins with the cleaning and dehydration of the substrate, the SU8 is then spun on, soft baked, exposed to UV light, post baked, cooled, developed, rinsed, and finally hard baked if the device will be used in a high temperature thermal device. This is the common process for SU8 fabrication, however SU8 fabrication has a few challenges. J.H. Daniel et al. [15] discuss some of the fabrication challenges of SU8 resists. SU8 fabrication process is sensitive to both heat and UV light due to the chemistry of the resist.

If the resist is either over or under baked, the cross-linking can be improperly developed which can make the material brittle. This can form cracks in the surface or the shearing of fine features and lack of adhesion [16]. The cracking lithographic features are caused by internal stresses in SU8 microstructure, and a lot of research is currently being done to study different methods to process and develop SU8 to maintain its desirable thermal, electrical and mechanical properties while decreasing the inherent internal stresses developed during processing. Johnson et al [17] is focusing on developing composites of SU8 to reduce stress cracking of microstructures. In this study, SU8 is mixed with low molecular weight aromatic epoxies and developed to test some of its major properties such as adhesion, resolution, aspect ratio, and feature cracking of the composite mixture. Feng and Farris [18] focused on developing different fabrication procedures and studying different property results. They studied to see how the thermal and mechanical properties of SU8 change by different processing conditions such as soft bake time, exposure time, post exposure bake time, development time, and different substrate materials. SU8 thickness of 50mm, 100mm, and 220mm were studied to optimize properties such as sidewall profile and film adherence.

In the past, removal of SU8 has also been major challenge. This process has been perfected with the use of the Omnicoat layer with a recently developed Remover PG SU8 Stripper. The Remover PG SU8 Stripper is an important development in the use of SU8 because this remover makes SU8 more applicable for a variety of uses [19].

Although a lot of research is being done, the SU8 fabrication process is sensitive to all steps including baking times and temperatures, cooling or resting times, exposure, and development times.

C.W. Liu et al. [13] developed a fabrication procedure for micro-channel systems that can be used to study the flow and the local heat transfer coefficient within the channel. The process forms a one-channel wall that is uniformly heated with the opposing wall well insulated. This design is meant to minimize the heat loss through the substrate. In this work, SU8 is used to form the micro-channels to aid in the heat transfer analysis of micro-channels. SU8 is used to form the channels and its thickness can easily be varied using varying viscosities and spin speeds during photolithography. Temperature sensors can be fabricated on the channel walls to record their temperatures and find the local heat transfer coefficient of the flow [7].

1.3.4 VISUALIZATION

Visualization of the flow in micro-channels is an important technique to yield a better understanding of what is happening inside the channels. Visualization techniques of the flow in micro-channels have been used to see the motion of the liquid-vapor interface in two-phase flow, and the flow due to capillary action [24].

However visualization of flow in micro-channels has its own challenges. Due to the length to width ration of most micro-channel, it is difficult to visualize throughout the length of a micro-channel. Fluid motion is also difficult to capture. Different dyes and lighting techniques have been used to illuminate the fluid to observe its motion through the channel. Among the many requirements (and some problems to be solved) to obtain worthwhile images of micro-channel fluid flow are visualization tools with resolutions down to the micro level, flexible lighting, and methods to deal with internal reflection from the channels [8].

K.H. Chang and Chin Pan [61] used a charge-coupled device (CCD) camera with a micro lens to observe the two-phase flow pattern in the micro-channel with hydraulic diameter of 86.3μm. The width and depth of each channel were 99.4μm and 76.3μm, respectively. The maximum frame rate available of the camera was 10,000 frame/s and the maximum shutter speed is 1/20,000s. R.H. Liu et al. [62] also used a CCD Camera to capture a image of passive mixing in the micro-channel. The width and depth of each channel were 300μm and 150μm, respectively. Images of the changing color of phenolphthalein dye during the mixing process are captured through an Olympus BX60 microscope at 40 times magnification with a Sony 8-bit CCD camera. The microscope lens has a depth of focus of 7μm at this magnification. This setup gives clear images of the purple phenolphthalein dye, but only allows images to be obtained at positions where the silicon wafer is etched completely through.

S.S. Hsieh et al. [63] used a MPIV, micro particle image velocimetry, system for visualize the liquid flow in a micro-channel. The size of channel was 115μm deep, 200μm wide and 24,000μm long with a hydraulic diameter of about 146μm. The MPIV system consists of a new wave solo PIV Nd:, Yag double resonant tube laser providing frequency-doubled (λ=532nm) pulsed emission of 50mJ/pulse (maximum) via a Q-switching modules with a pulse duration of approximately 10ns. The light from the laser was delivered into the microsope illumination path with a fiber light guide fitted to the microscope. Particle images are recorded using a Dentec HiSense CCD camera with a 1280x1024 pixel array and 12-bit resolution. With the aid of computer, the microscope and CCD camera built into the MPIV system can record the images of the flow pattern

evolution in the micro-channel at 20 frames/s. A Sony DCR-TRV34 color video camera with an area of 320x240 pixels for each frame was used.

S. Hardt [64] used a high-speed camera mounted on microscope to visualize the flow patterns emerging during evaporation in parallel micro-channels. The width and depth of each channel were 50μm and 50μm, and a length was 64.5mm. A limiting factor for the flow-pattern visualization with the high-speed camera was the amount of light available for one frame. The amount of light collected is reduced by using a microscope and limiting the visible area of the chip surface. The brightness of images is further reduced by increasing the shutter speed of camera. With these constraints a maximum frame rate of 5000 frames/s could be reached when using four independent light sources.

1.4 FIXED SU8 HEIGHT

The goal of the present work was to use a combination of both experimental and numerical work to provide insight into the controlling parameters and design constraints of the micro-channel evaporator. One of the controlling parameter was the dimension of micro-channels. The dimensions of these micro-channels were varied to determine how this factor affects the test results. The dimension factors include micro-channel geometry, micro-channel width, and SU8 height. The SU8 height used in this work is fixed at 40μm and this value was determined by previous work by Tiffany [74]. Tiffany considered effect of micro-channel height and a summary of these results are shown in the Figure 1.2 and table 1.1. In general, the higher micro-channel decreases aspect ratio fluid micro-channel and increase the fluid mass over the membrane.

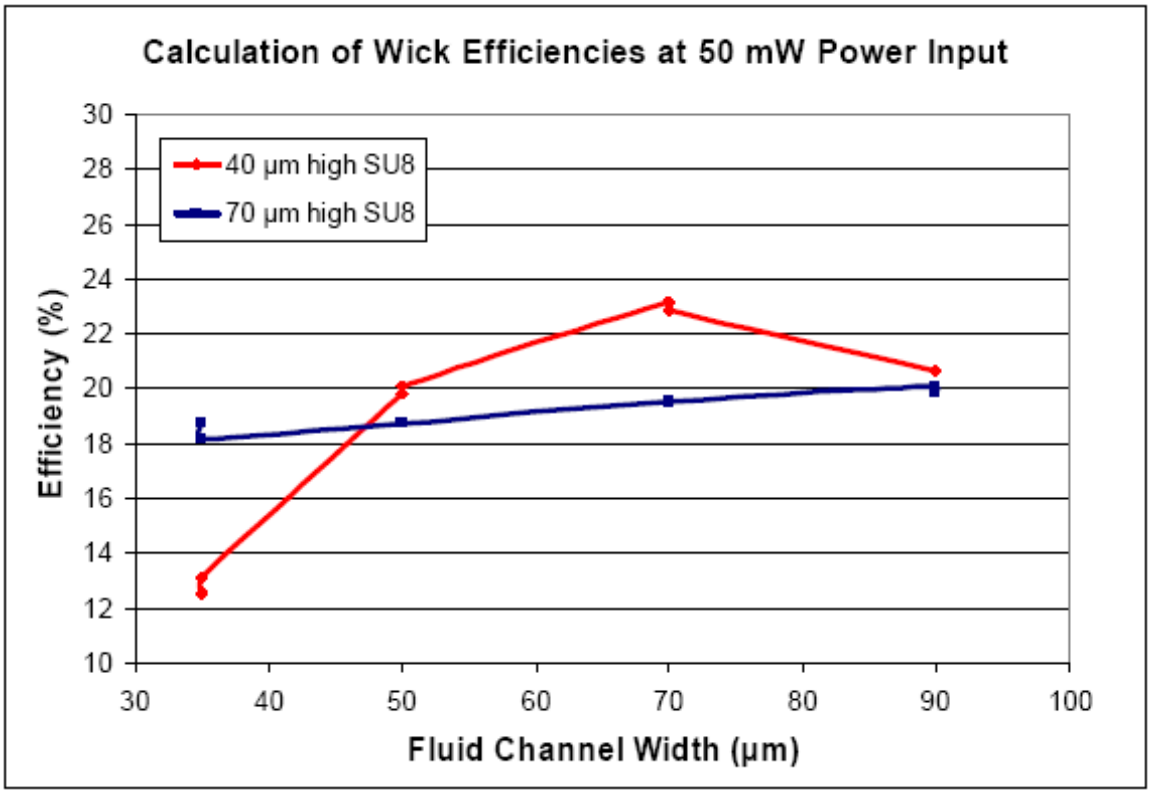


Figure 1.2 Calculation of wick efficiencies for 40 and 70µm channel heights for 50mW of power input assuming constant evaporation rates, Courtesy T.A. Quy

SU8 Height (µm)	Channel Width (µm)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (mg/min)	Efficiency (%)
40	35	37.7	35.2	30.0	30.6	6.3	4.5	16.6
	35	42.1	42.0	35.9	35.8	6.6	4.7	15.5
	50	36.9	36.3	29.8	38.9	10.2	7.1	26.8
	50	35.6	31.5	27.4	24.6	10.0	7.2	28.3
	70	41.2	34.0	29.0	29.7	11.6	8.3	27.8
	70	31.1	30.7	27.5	19.2	11.4	8.2	36.7
	90	33.5	29.8	26.1	22.2	10.3	7.4	30.8
70	35	46.2	34.9	29.0	35.1	9.4	6.7	20.2
	35	50.7	35.6	29.8	38.7	9.2	6.5	18.4
	50	47.8	35.0	29.1	38.7	9.2	6.7	19.6
	70	43.6	34.0	28.1	38.7	9.2	7.0	22.4
	90	40.3	33.3	27.7	33.1	7.7	7.2	24.9

Table 1.1 Wick efficiencies for 40 and 70µm high channels, Courtesy T.A. Quy

Table 1.1 shows the efficiencies of the 40μm high micro-channels are mostly higher than the efficiencies of the 70μm high micro-channels. However, the comparisons of the micro-channel evaporator performances were not accurate due to varied power inputs. The constant power input needs to be used to make an accurate comparison of the micro-channel evaporator performances since the micro-channel evaporator efficiencies were defined as the ratio of evaporation power over power input. The choice of 50mW was used for all experiments performed on 40μm and 70μm high wick structures. Figure 1.2 shows this comparison results. The 40μm high wicks reached a maximum efficiency at a channel width of 70μm. A general efficiency trend of the 40μm high wicks was higher than the efficiency of the 70μm high wicks. The efficiency of the 10μm high wick channels was also tested and compared, however the Figure 1.2 did not indicate the results of 10μm high wick channels because the efficiency values were significantly low compared to the efficiencies of 40μm and 70μm high wick channels. Since the results of previous work [74] conclude the 40μm high wick channels reached maximum efficiency compared to other channel heights, it was reasonable decision to use this 40μm high wick channels in this present work.

1.5 RESEARCH OBJECTIVES

Phase change actuators function by dissipating electrical power to heat a working fluid and change its phase. However the efficiency of conversion from electrical to mechanical power is very low. To maximize or optimize the mechanical power produced it is required to study the micro-channel evaporator, to increase the effectiveness of heat addition and evaporation.

The present work builds upon previous experimental and numerical work [65, 66, 67] to develop and validate a numerical model of evaporative heat transfer for square open-top micro-channels. The objective of the numerical work is to develop a three-dimensional finite-difference time domain integration for the analysis of the sensible and latent heat transfer.

The objective of the evaporation experiment is to validate the numerical model by documents the performance of the micro-channel evaporator with varying micro-channel dimensions and membrane materials. The ultimate goal of the work is to build an effective design tool for the development of efficient micro-channel evaporator with dimensions from ten to one hundred microns and to find the overall most effective micro-channel evaporator geometry and dimensions.

CHAPTER 2

FABRICATION

2.1 FABRICATION OVERVIEW

A micro-system is an engineering system that contains MEMS components, and many MEMS components can be produced in the size of micrometers. These micro-size components are not able to be fabricated with the traditional mechanical techniques such as machining, drilling, milling, forging, casting, and welding. The technologies used to produce these micro-size components are called micro-fabrication [68]. Almost all micro-fabrication techniques or processes involve physical and chemical treatment.

Most of the micro-fabrication techniques used in this dissertation are photolithography, wet etching, and physical vapor deposition (Sputtering). The fabrication of the micro-channel evaporator presented in this dissertation used the most fundamental and proven micro-fabrication techniques and materials. The significant micro-fabricated components mentioned in this chapter consist of a heater membrane, resistance thermometer device (RTD), and capillary wicking structure.

2.2 PREPARATION OF SILICON (Si) MEMBRANES

A boron doped silicon membrane is used for the basic building block for fabrication of the micro-channel evaporator components. The substrate wafer used in this dissertation is a standard three-inch diameter, double side polished, and [100] orientation silicon wafer. The thickness of the wafer is approximately 400μm with a tolerance of ±25μm.

High Temperature Oxidation (HTO) is used for the first processing step of the

fabrication in order to grow a layer of SiO_2 on both front side and backside of the wafer. Wafers are baked in a furnace at 1000^{o}C for 120 minutes to achieve 500nm oxide thickness. The wafer is soaked in a 10:1Buffered Oxide Etch (BOE) solution for 15 minutes to remove oxide on the front side of the wafer and to prepare boron doping. The boron doping is used as an etch stop for the wet etchants used in the fabrication process. The final thickness of the fabricated membrane is determined by the thickness of the boron layer. The thickness of the final membrane used in this dissertation is 2.2μm.

The boron diffusion occurs by placing boron nitride disks parallel to the exposed surface of the wafer at elevated temperatures. Placing the wafers in the furnace at 1125^{o}C for 110 minutes produced a boron layer of 2.2μm.

Several processing steps are still needed to complete the substrate wafer after boron diffusion. The appearance of the boron-doped surface is non-uniform color because of the borosilicate glass that is built up on the wafer. A three step process is needed to remove the borosilicate glass, and the three step process consists of a BOE etch, sacrificial oxide growth and a second BOE etch. The wafer is soaked in the BOE for 15 minutes to remove the borosilicate glass. A low temperature oxide (LTO) process is grown onto the wafer at 850^{o}C for 60 minutes. The sacrificial oxide and residual oxide from HTO are removed by soaking the wafer in BOE for 10 minutes. The final low temperature oxide (LTO) is grown onto the surface of the wafer to act as an electrical insulation layer. Finally, the substrate silicon wafer is ready for the next processing step.

2.3 PREPARATION OF SILICON NITRIDE (SiNx) MEMBRANES

Silicon nitride membranes are also used as the membrane material in this work. A 300nm layer of silicon nitride (SiNx) is deposited on both front and backside of [100] orientation silicon wafer using a Low-Pressure chemical vapor deposition (LPCVD) process instead of adding a boron-doped layer to the wafer. The bulk of silicon nitride wafers are ordered from outside of Washington State University (WSU) because of the LPCVD process is not available at WSU. The deposited silicon nitride on the front side of wafer forms the final membrane, and the deposited silicon nitride on the backside of wafer acts as an etch stop for KOH etchant.

2.4 PATTERNING WAFERS FOR MEMBRANE

The same photolithography technique is used to define membrane size, resistance heater, RTD, and micro-channel evaporator for both silicon and silicon nitride membranes. Figure 2.1 shows a typical mask for defining 5mm square membrane for both silicon and silicon nitride membranes. The thin horizontal and vertical lines in the figure form an 18mm by 10mm rectangle and are used for dicing guides. The four small crosses are used for alignment purposes. The masks are designed for positive processing. Figure 2.1 shows the positive field of design but the negative of mask image is actually used for fabrication.

The patterning on the silicon and silicon nitride wafers seem to have a lot in common, however two different processes are used to pattern membranes on each type. The process for the silicon membrane is explained in section 2.4.1 and for the silicon nitride membrane is explained in section 2.4.2.

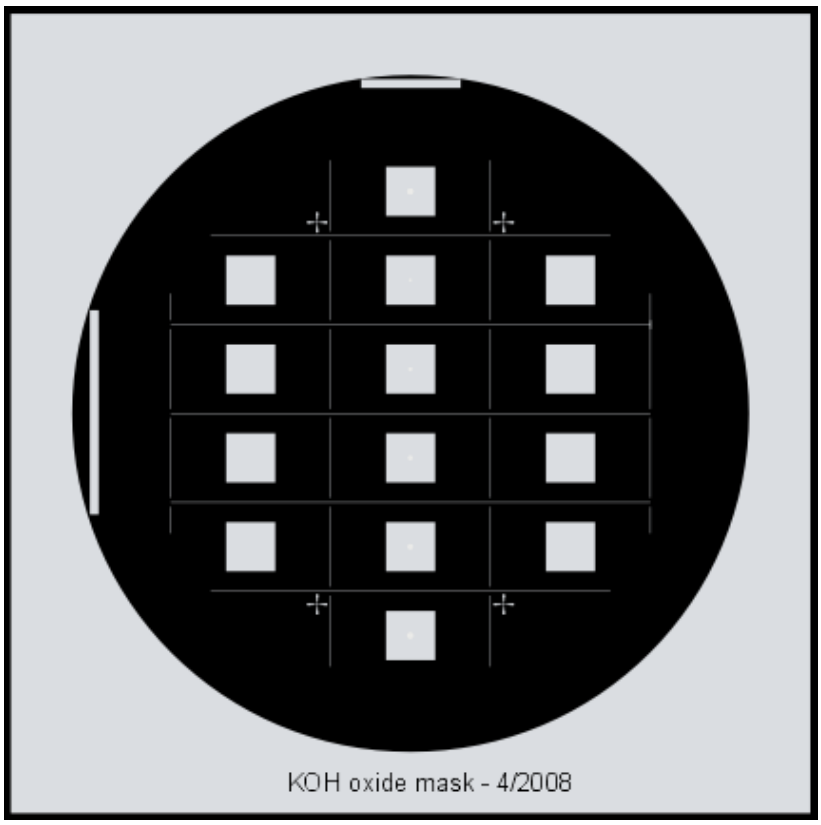


Figure 2.1 Mask for etching the oxide to define membranes.

2.4.1 SILICON (Si) MEMBRANE FABRICATION

Fabrication of silicon membranes begins with a prepared silicon wafer as described in section 2.2. The first step is to sputter 500nm of gold on the backside of the wafer along with a TiW adhesion layer. The hexamethyldisilazane (HMDS) is spin-coated to serve as an adhesion promoter. The HMDS is spun for 30 seconds at 3000 rpm. The wafer is immediately spin-coated with AZ5214 photoresist at 3000 rpm for 30 seconds, and baked for one minute at 110 ^{o}C. The backside of wafer is now patterned with the mask shown in figure 2.1 using a UV light source with exposure time of 12 seconds. AZ5214 is a positive photoresist, which means that the exposed portion of the wafer is removed away by developer.

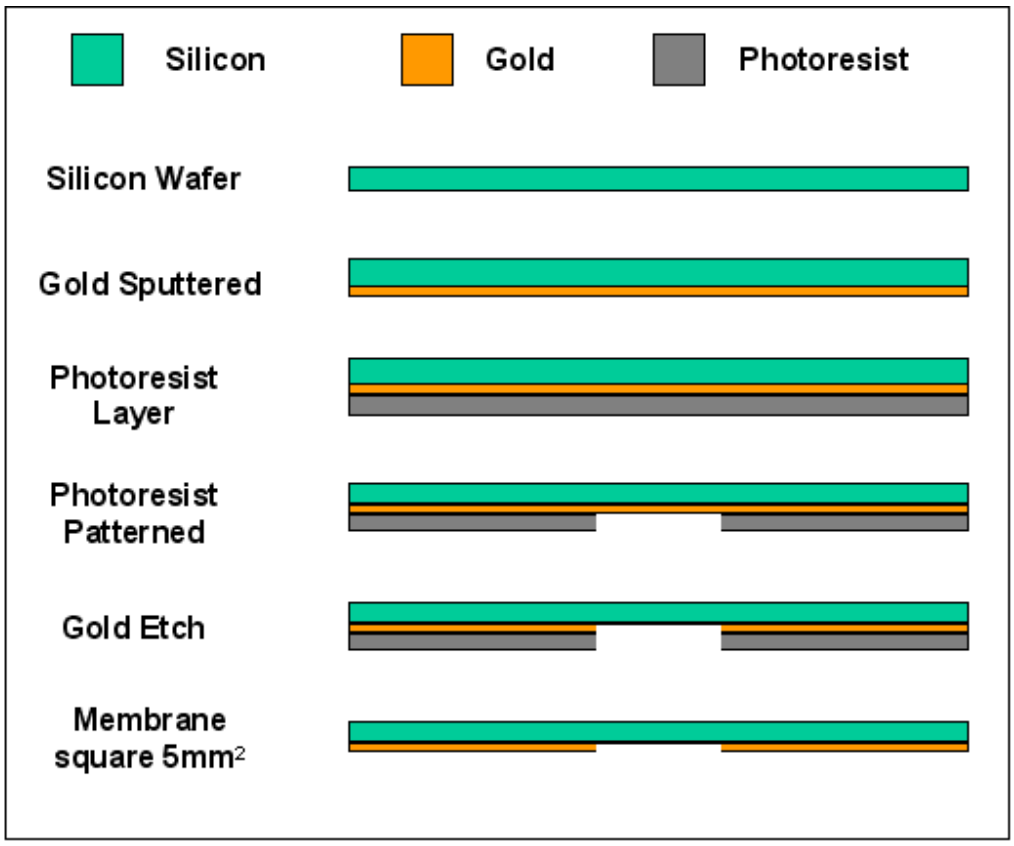


Figure 2.2 Illustration of photolithography process

400K-photoresist developer is used in this work with a 1:4 ratio with water (e.g. 12mL 400K with 48mL DI water). The photoresist works as an etch stop for etching through the gold, TiW, and SiO_2 layer during the backside of wafer processing. The wafer is placed in the gold etchant with constant agitation until gold is etched away and then the wafer is soaked in hydrogen peroxide for 40 seconds with constant agitation to remove TiW adhesion layer from the membrane pattern. The wafer is then placed in the Buffered Oxide Etch (BOE) for 15 minutes to remove the SiO_2 layer for membrane etching. The wafer is now placed in the spin-rinse dryer to be washed thoroughly. The gold on the backside of wafer is an etch stop for KOH membrane etching. The photolithography process used in this work is illustrated in Figure 2.2.

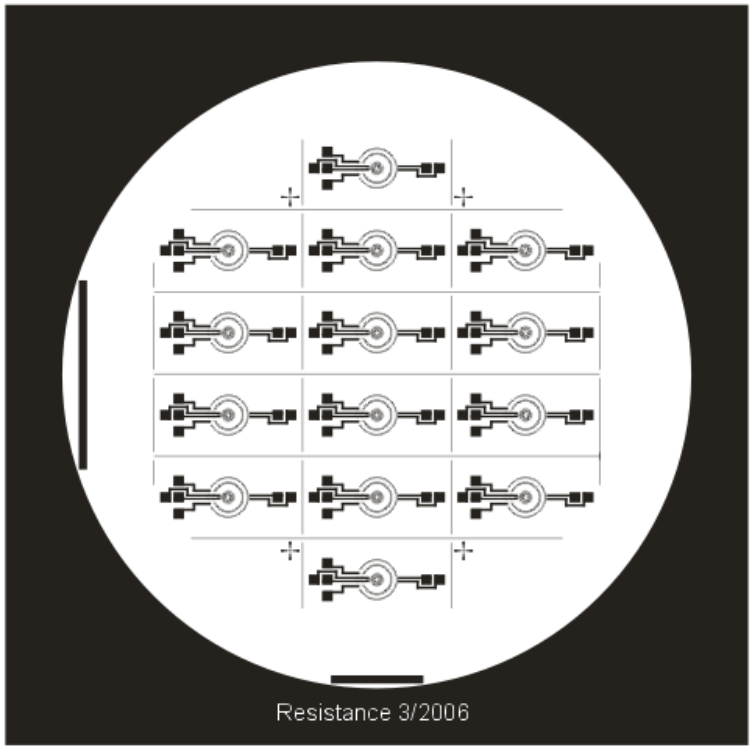


Figure 2.3 Photolithography mask for resistance heater and dual RTDs

With the backside patterning completed, the front side is now patterned using a lift-off process. The platinum lift-off process is used to fabricate the internal platinum resistance heater and the concentric Resistance Temperature Detector (RTD) structures. The photolithography mask for the resistance heater and the dual RTDs is shown in Figure 2.3. The outside radius of the internal heater is 0.8mm, and the average radius of the inner RTD and outer RTD is 1.7mm and 2.4mm respectively. Again, the thin horizontal and vertical lines in the figure forms an 18mm by 10mm rectangle are used for dicing guides. The four small crosses are used for alignment purposes. More design details for concentric resistance heater and RTDs are explained in chapter 3.

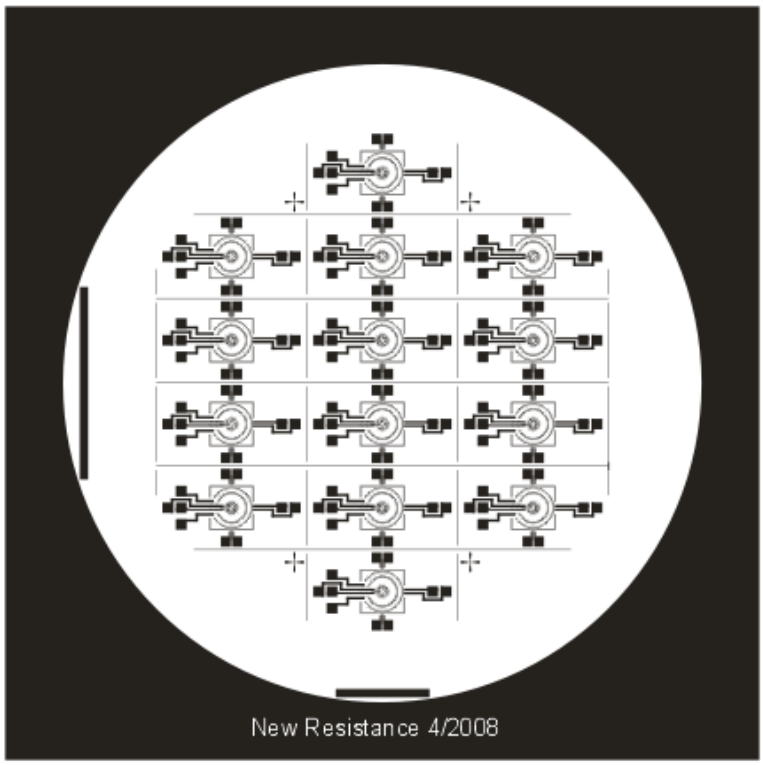


Figure 2.4 Photolithography mask for resistance heater and triple RTDs

A new photolithography mask has features of a resistance heater and triple RTDs. The new mask shown in Figure 2.4 has exactly the same features as the old mask shown in Figure 2.3, however the new mask has one more RTD around the 5mm square membrane defined in the backside patterning. This third RTD feature in the new mask is used to find the fixed boundary condition temperature for numerical calculations. To fabricate the internal platinum resistance heater and concentric RTD structures, first the wafer is spun with an adhesion promoter HMDS at 1500rpm for 30 seconds. The positive photoresist AZ5214 is used to pattern the heater and RTD structures. The spin rate and time for the photoresist is 1500rpm for 30 seconds, and soft baked for 1 minute at 110 ^{o}C. The wafer is exposed for 15seconds using the correct negative field mask shown in Figure 2.3 and 2.4.

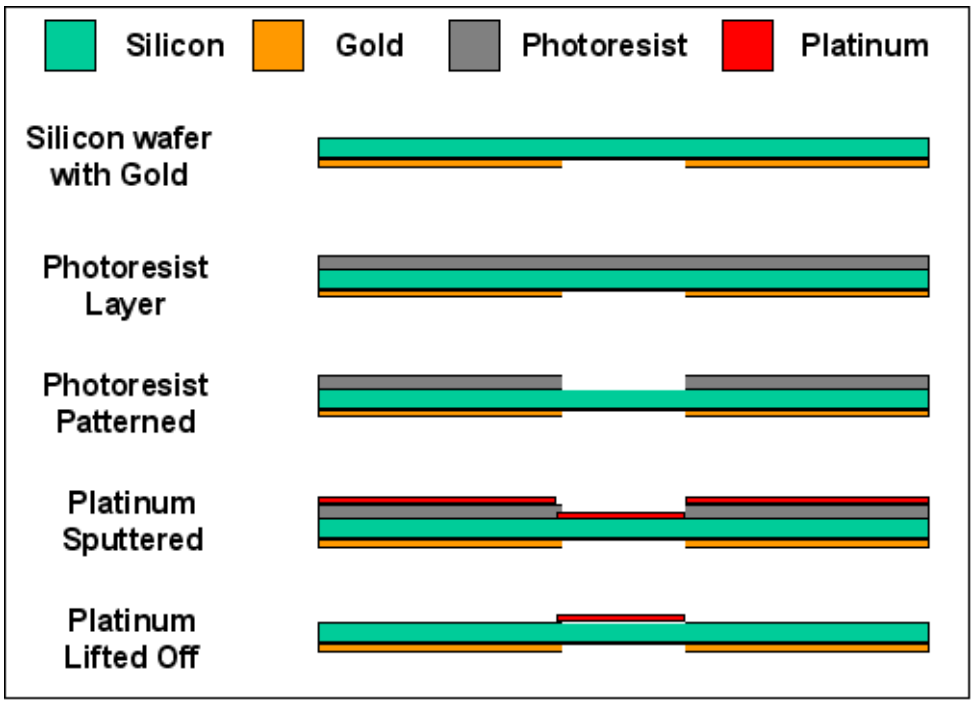


Figure 2.5 Illustration of platinum lift-off process

The wafer is covered in photoresist, and the heater and RTD designs are the only fraction of the wafer not covered with photoresist. The desired features are developed using the 400K-photoresist developer with a mixed ratio of a 1:4 with DI water. Platinum is now sputtered over the photoresist at a thickness of 175nm. After sputtering, the wafer is soaked in acetone for 30 or more minutes to remove the sacrificial photoresist layer and unwanted Platinum. The wafer needs to be "blasted" with acetone to remove any small chunks of Platinum remaining on the surface of wafer. The wafer is then cleaned and checked under the microscope. The features should be solid and clear. If there is still unwanted Platinum left on the wafer, soaking or blasting longer is required until it is mostly removed. The illustration of the platinum lift-off process is shown in Figure 2.5.

The wafer is now annealed at a temperature of 650^{o}C for duration of ten minutes in the vertical furnace that lowers the internal resistance of the platinum structures and

homogenizes the gold to help with potassium-hydroxide (KOH) membrane etching.

2.4.2 SILICON Nitride (SiNx) MEMBRANE FABRICATION

The fabrication of silicon membranes begins with a prepared silicon wafer as described in section 2.3. Photolithography begins by spinning an adhesion promoter, HMDS, at 3000rpm for 30seconds and soft baked at 110 ^{o}C for one minute. The wafer is then spin-coated with AZ5214 photoresist on the silicon nitride wafer to define a membrane pattern. The wafer is exposed to UV light for about 12 seconds, and developed to achieve the desired pattern. The photoresist acts as an etch stop during the reactive-ion etching process. The desired pattern used in this work is the same as that shown in Figure 2.1. The wafer is exposed to a CF_4 reactive-ion etcher (RIE) plasma to etch the backside of silicon nitride layer from the membrane squares. The backside silicon nitride layer acts as an etch stop for KOH membrane etching.

The patterning process of front side of silicon nitride membrane is the same process with silicon membrane patterning. The remaining fabrication steps are patterning, platinum lift-off, and annealing of the wafer. The detail patterning process of the front side membrane is explained in section 2.4.1. The same masks are also used to pattern the resistance heater and RTDs on the silicon nitride membrane. The masks used in this work are shown in the Figure 2.3 and Figure2.4.

2.5 FABRICATION OF MICROCHANNEL EVAPORATOR

After the wafer is patterned with membranes on both the front side and backside and annealed, the wafer is now ready to be fabricated with micro-channel evaporator on the

substrate surface. The micro-channel evaporator is fabricated concentrically over the platinum structures on the substrate membrane. The micro-channel evaporator is used to draw a thin layer of working fluid, and maintain the working fluid. Since the micro-channel evaporator is fabricated on the resistance heater, the micro-channel need to be able to tolerate high temperatures for the heat addition process of this work. SU8 is the material chosen for the micro-channel evaporator because of its fabrication ability, high aspect ratios forming ability, and good thermal properties. More details about SU8 materials are explained in section 1.3.1. SU8 is a negative photoresist, so the exposed chemical remains on the wafer while the unexposed fraction is removed during development. Two types of SU8 polymer are used. SU8 2010 is used for forming the 10μm high micro-channel evaporator, and SU8 2025 is used for 40μm high micro-channel evaporator. Several different micro-channel evaporator dimensions are presented in this dissertation. The one type of micro-channel evaporator consists of channels 10μm in width and 10μm high, and 10μm thick walls of SU8 delineate these channels. The other type of micro-channel evaporator was fabricated by spinning on 40μm height of SU8 with width of 35, 50 and 70 channels bounded by 5μm wide SU8 walls. Finally, the 80μm narrow to 5μm wide tapered channels and SU8 walls with 40μm height micro-channel evaporator are fabricated. The material is very sensitive to each process, so MicroChem publishes fabrication guidelines for all of their SU8 resists. Although SU8 is fabricated using a standard lithography technique, each fabrication process step needs to be reanalyzed for each channel thickness and geometry because the application parameters such as exposure equipment, bake times and methods, and development times are all affect the fabrication results.

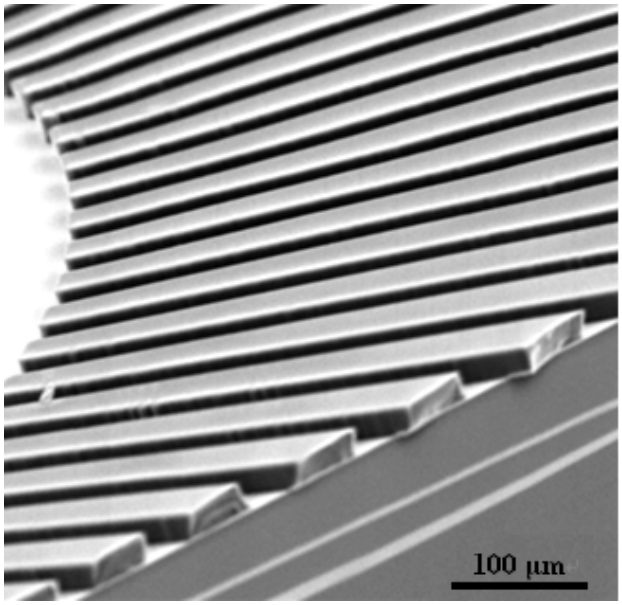


Figure 2.6 SEM examples of 10µm high SU8 wicks, Courtesy J. Martinez

To fabricate 10µm high micro-channel evaporator on the surface of the heater membranes, the photosensitive SU8 2010 epoxy is used. The first step for 10µm high micro-channel is the spin coating of Omnicoat at 3000rpm for 30 seconds as an adhesion layer. After Omnicoat deposition, the Omnicoat layer is baked 1 minute at 200^{o}C and left to cool to room temperature. The wafer is now spin-coated with SU8 2010 at 2000rpm for 30 seconds, and soft baked at 65^{o}C for 1minute and at 95^{o}C for 2minutes. The SU8 is now exposed via UV light for 15seconds and then post baked. The post bake times are at 65^{o}C for 2minutes and at 95^{o}C for 3minutes, then again at 65^{o}C for one minute.

With baking completed, the SU8 is developed via 3minutes with strong agitation in the MicroChem SU8 developer. The desired 10µm high micro-channel evaporators are now fabricated on top of the platinum heater and RTDs. Figure 2.6 shows the SEM photograph for an example of 10µm high SU8 micro-channel evaporator.

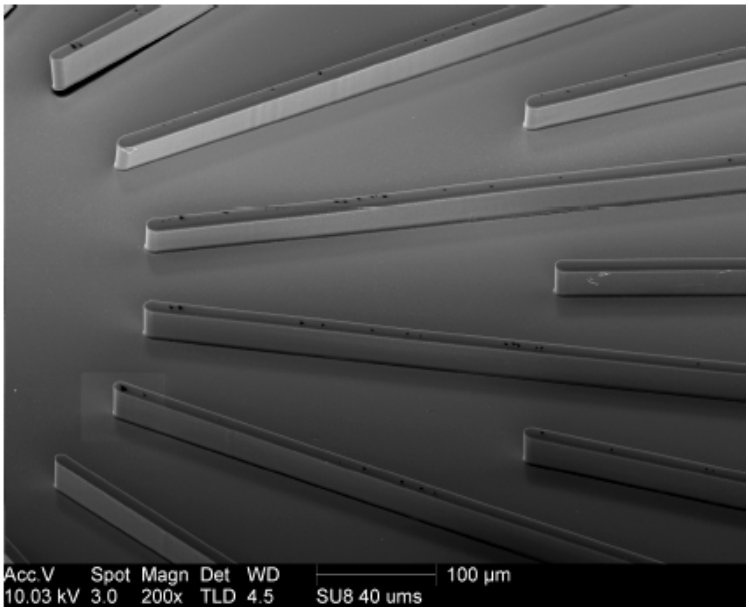


Figure 2.7 SEM examples of 40µm high SU8 micro-channels, Courtesy T. Quy

40µm high micro-channel evaporators are fabricated on the surface of the heater membranes using the photosensitive SU8 2025 epoxy. Figure 2.7 shows the SEM photograph for an example of 40µm high SU8 micro-channel evaporators. To fabricate these micro-channel evaporators, an adhesion layer Omnicoat is spin coated at 3000rpm for 30seconds and soft bake at 200 ^{o}C for 1minute. SU8 2025 is pre-spun at 500rpm for 7sec with acceleration of 100rpm/s, and is checked for any bubbles that might have occurred. If bubbles occur during the pre spin coat process, it is necessary to pop or remove the bubbles for successful fabrication. With bubbles removed, SU8 2025 is now post spun at 2000rpm for 30seconds with an acceleration of 300rpm/s for 40µm height.

Next, SU8 2025 is soft bake at 65^{o}C for 2minutes 30 seconds and then post bake at 95^{o}C for one hour in a convection oven. The SU8 2025 is now exposed via UV light source with predetermined time. Exposed time for 35µm wide channels are 55seconds, and the rest of channels sizes for 65 seconds.

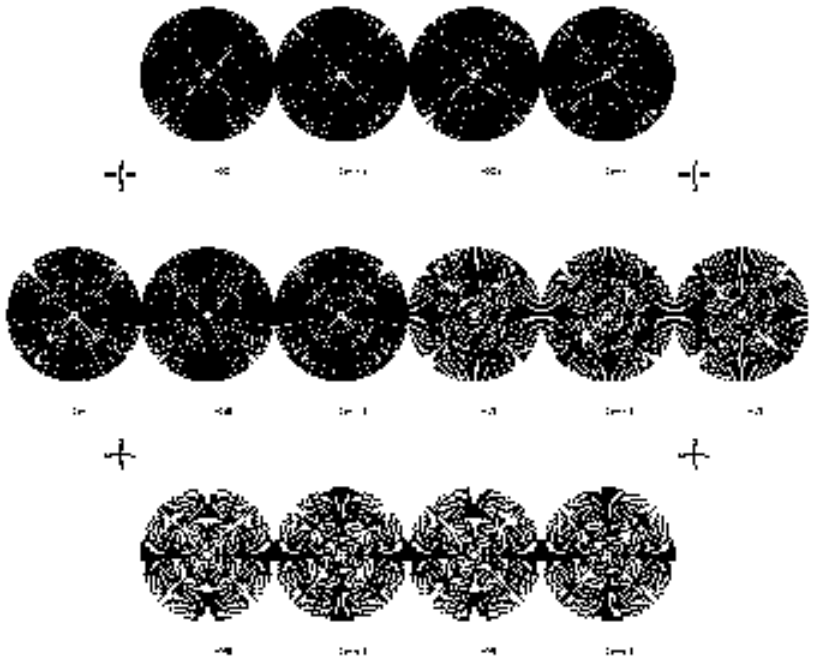


Figure 2.8 AutoCad designs of micro-channels with different channel widths

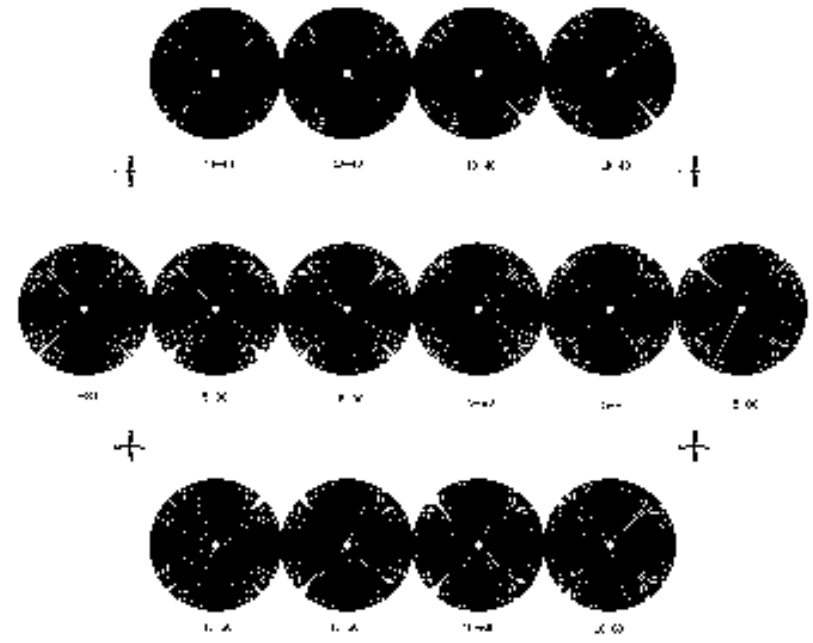
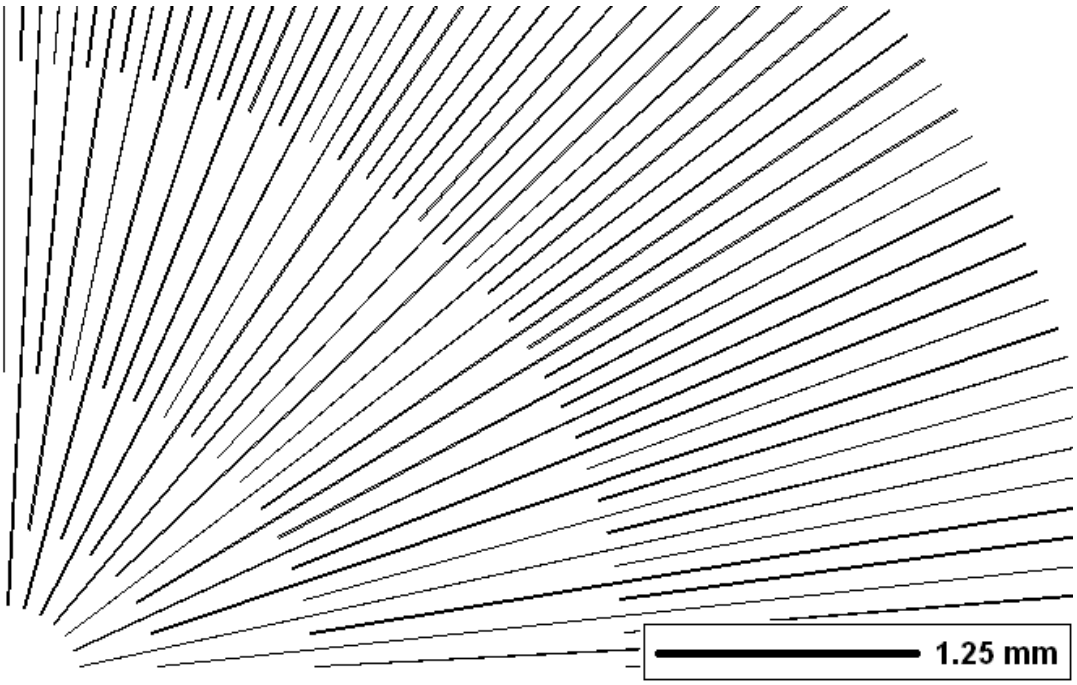
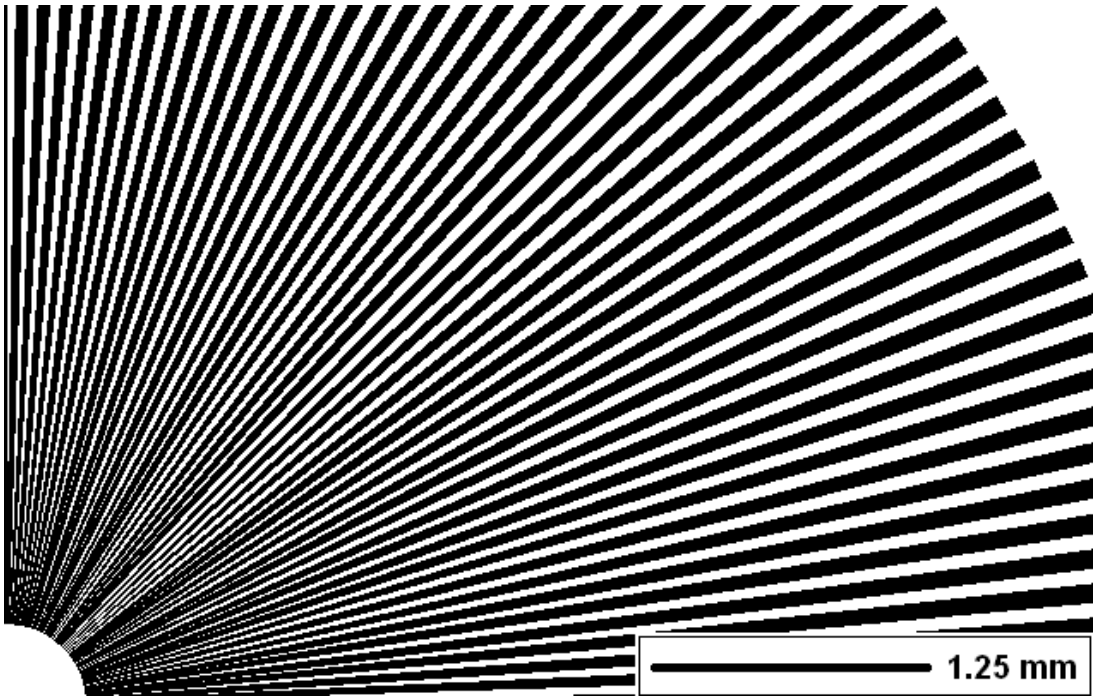


Figure 2.9 AutoCad designs of tapered micro-channels



(a) 70μm wide channels with 5μm wide SU8 walls



(b) 80μm narrow to 5μm wide tapered channels and SU8 walls

Figure 2.10 Magnification Image typical Masks for Different Wick Geometries

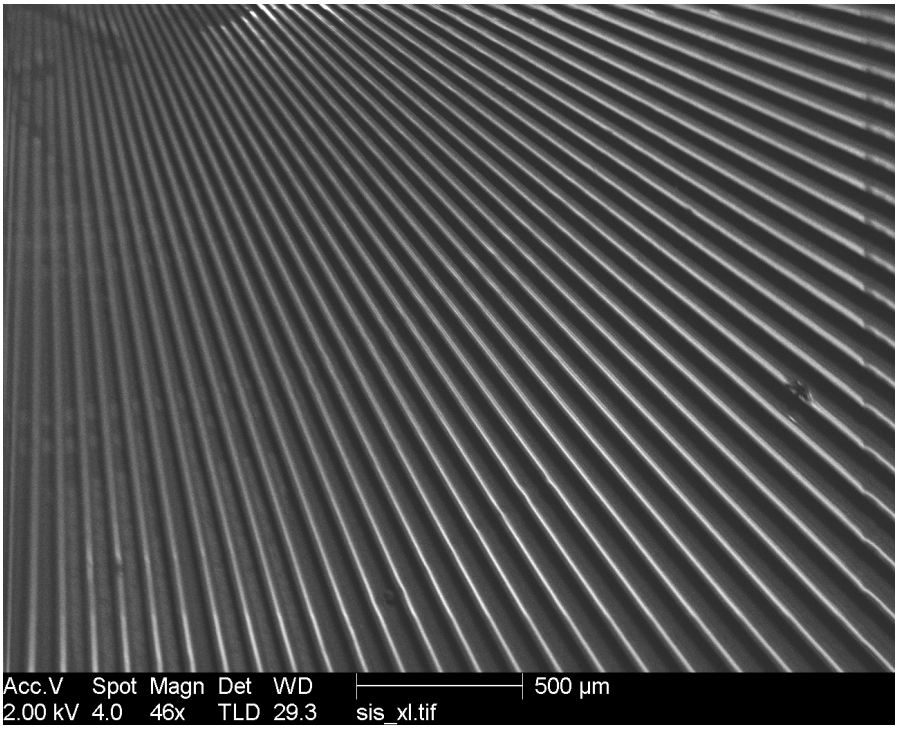


Figure 2.11 SEM photographic image of the tapered channels

High-resolution chrome masks are used to define the shape of the micro-channel evaporator during UV light exposure. The chrome masks are made at the University of Minnesota Nano Fabrication Center (NFC) from an AutoCad template. The AutoCad design for SU8 mask is shown in Figure 2.8 and 2.9. Figure 2.8 shows the design of micro-channels with different width, and Figure 2.9 shows the design of the tapered channels. With completion of UV light expose process, the SU8 is post baked at 65^{o}C for one minute on first hot plate, then place on the second hot plate for 4minutes at 95^{o}C. To prevent cracking of the fabricated wafer, it is now cool to room temperature.

Finally, SU8 2025 is developed using a full immersion in SU8 Developer for 5minutes and 15seconds with constant strong agitation. This leaves the desired SU8 2025 micro-channel evaporator on top of the platinum heaters and RTDs.

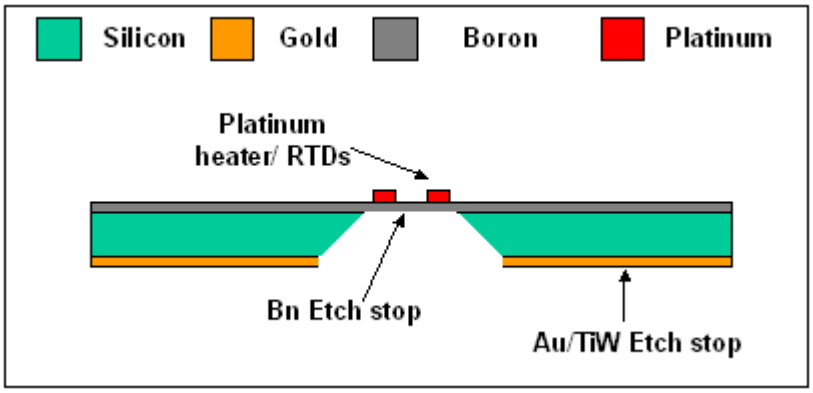
Figure 2.10 shows magnification image typical masks for different micro-channel geometries as shown in Figures 2.8 and 2.9. The radius of each pattern shown in Figure 2.10 is 5mm. Because the features are so small relative to the membrane size, only a small portion of entire feature is shown.

Figure 2.11 is a Scanning Electron Microscope (SEM) photographic image of a portion of the tapered channels that has corresponding dimensions to Figure 2.10 (b).

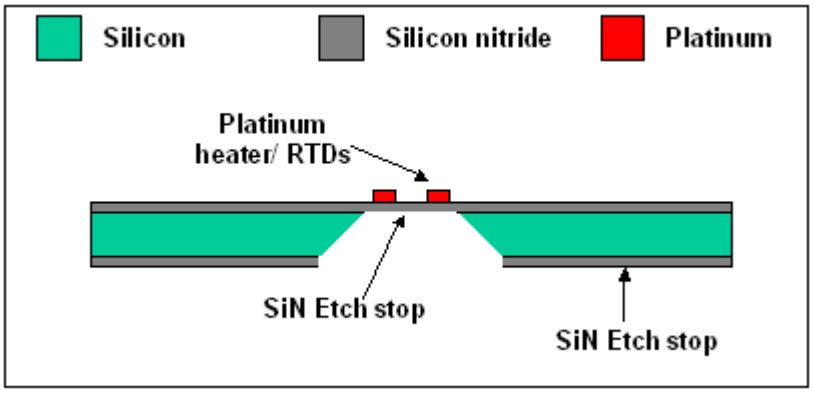
2.6 KOH ANISOTROPIC ETCHING

The final fabrication process in membrane creation is etching the wafer to define the membrane structures. A potassium hydroxide (KOH) solution is a highly selective wet etch for this process. The KOH solution is the mixture of 250g of KOH dry pellets with 400mL of deionized water, and this solution is heated to a temperature approximately 80^{o}C.

If the KOH solution temperature is between 70^{o}C and 80^{o}C, the solution is now ready for use. The wafer is placed and sealed in a carrier device, and immersed in the etching solution. The backside of wafer is exposed to the solution for about 5 to 6 hours depending on the etch rate. Since KOH solution is highly dependent on temperature to control etch rate, the etching progress should be checked frequently. Etching is completed when the membranes become translucent and the forming of bubbles on the etching surface slow. The bubble forming is slowed down because the KOH solution has now reached to the boron doped layer or the silicon nitride layer. Cross sections of the etched structures for silicon (Si) membrane and silicon nitride (SiNx) membrane are shown in figure 2.12.



(a) Cross section of KOH etch for silicon (Si) membrane



(b) Cross section of KOH etch for silicon nitride (SiNx) membrane

Figure 2.12 Cross section of KOH etch for different materials

Figure 2.13 shows a top view of a competed evaporator membrane. The figure shows that clear images of the annular lines in the middle that forms the resistance hater and two concentric annular sets of lines that make up the two RTDs. The black radial lines on the top of the resistance heater and RTDs form the micro-channel evaporator. Figure 2.14 shows the layout of a micro-channel evaporator.

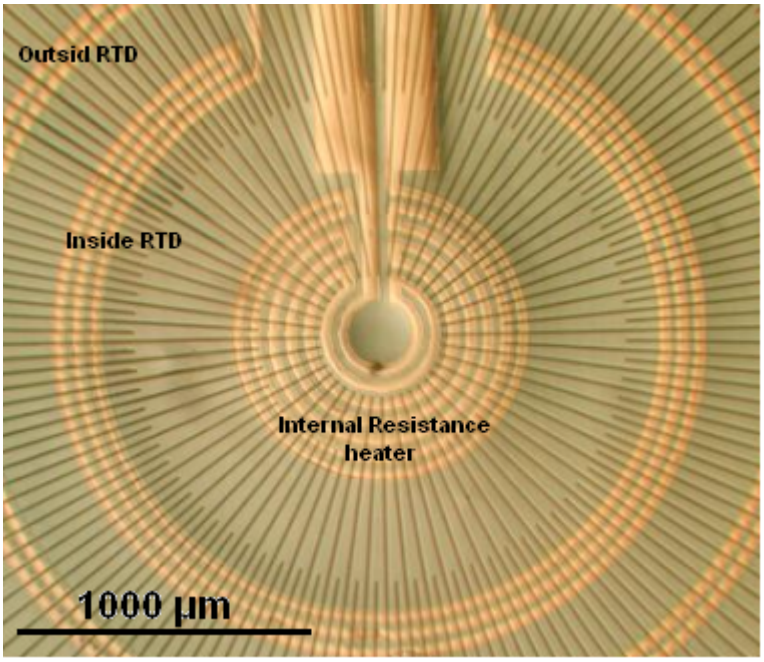


Figure 2.13 Top view of completed evaporator membrane

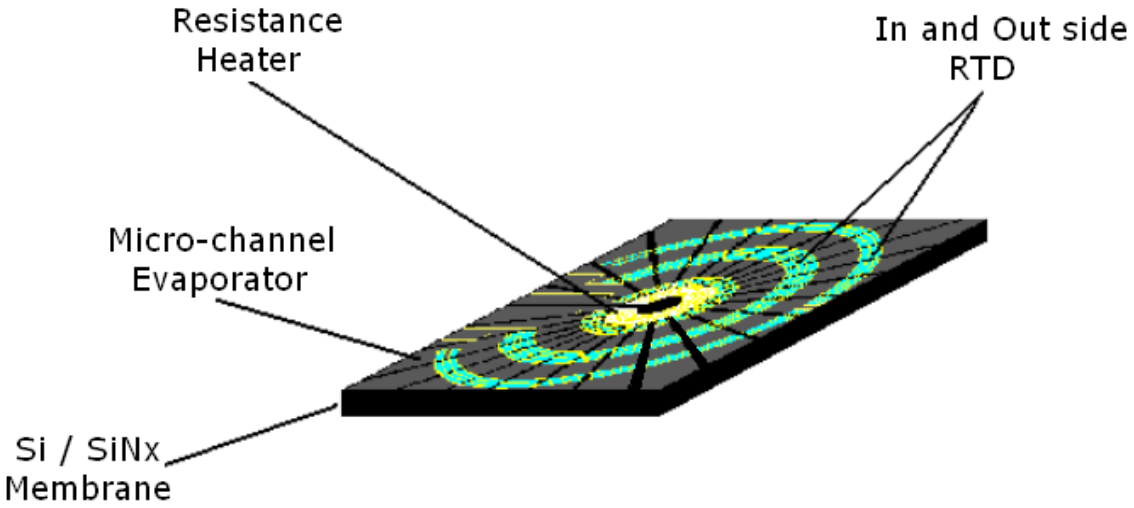


Figure 2.14 Layout of a micro-channel evaporator

2.7 DETAILED FABRICATION PROCEDURE

The completed evaporator consist an internal platinum resistance heater, concentric platinum RTDs, and micro-channels. To fabricate this micro-channel evaporator, several fabrication processes are needed as mentioned in sections from section 2.2 to section 2.6. Since these sections explained each process, the complete processing list for the fabrication is provided in this section.

2.7.1 FABRICATION STEPS FOR SILICON RTD/HEATER DESIGN

1) Sputter 500nm of gold on backside of stock LTO, boron doped wafer.
2) Spin both side of wafer with photoresist.
3) Pattern the backside of wafer with oxide pattern.
4) Develop photoresist and etch gold to leave oxide pattern gold free (see section 2.7.3).
5) Soak wafer in BOE to etch silicon dioxide from wafer.
6) Clean in the spin rinse dryer.
7) Pattern the front side of wafer with Dual or Triple RTD and concentric heater pattern.
8) Supper front of wafer with Platinum.
9) Lift-off sacrificial Pt with acetone (see section 2.7.4).
10) Anneal the wafer in the vertical furnace.
11) Spin wafer with SU8 to fabricate wicks over the RTDs and heater (see section 2.7.5 and section 2.7.6).
12) Etch membranes with KOH (see section 2.6)

2.7.2 FABRICATION STEPS FOR SILICON NITRIDE RTD/HEATER DESIGN

1) Spin both sides of wafer with photoresist.

2) Pattern the backside of wafer with oxide pattern.

3) Remove SiNx in RIE.

4) Pattern the front side of wafer with Dual or Triple RTD and heater pattern.

5) Sputter front of wafer with platinum.

6) Lift-off sacrificial Pt with acetone (see section 2.7.4).

7) Anneal the wafer in the vertical furnace.

8) Spin wafer with SU8 to fabricate wicks over the RTDs and heater (see section 2.7.5 and section 2.7.6).

9) Etch membrane with KOH (see section 2.6).

2.7.3 PHOTORESIST FABRICATION FOR GOLD ETCH

1) Sputter wafer with desired gold thickness.

2) Turn on mask aligner and set hot plate to 110^{o}C.

3) Clean wafer using 5-step process (Acetone, IPA, DI, Acetone, IPA, dry).

4) Dehydrate wafer for 1minute at 110^{o}C.

5) Spin wafer with HMDS for 30seconds at 3000rpm for both side.

6) Spin wafer with AZ5214 photoresist for 30seconds at 3000rpm for both side.

7) Bake for 1minute at 110^{o}C.

8) Expose for 12 seconds using correct negative field mask.

9) Develop using 400K-photoresist developer (mixed at a 1:4 ratio with water) with constant agitation about 1 minute and rinse with DI.

10) Soak in gold etch with constant agitation until gold is gone (should be 45seconds to 2minutes depending on gold thickness) and rinse with DI.

11) Soak in hydrogen peroxide for 40seconds with constant agitation for etch TiW adhesion layer.

12) Going though BOE to remove SiO_2 layer for membrane etching.

- Turn on the rinse tank and spin rinse dryer.
- Place wafer in white wafer carrier and submerge carrier into BOE for 15 minutes
- Rinse for 1minute in each rinse tank.
- Remove from hood and place in spin rinse dryer to wash thoroughly.

2.7.4 PHOTORESIST FABRICATION FOR PLATINUM LIFT-OFF

1) Turn on mask aligner and set hot plate to 110^{o}C.

2) Clean wafer using 5-step process (Acetone, IPA, DI, Acetone, IPA, dry)

3) Dehydrate wafer for 1minute at 110^{o}C.

4) Spin wafer with HMDS for 30seconds at 1500rpm.

5) Spin wafer with AZ5214 photoresist for 30 seconds at 1500 rpm.

6) Bake for 1minute at 110^{o}C.

7) Expose for 15seconds using correct negative field mask.

8) Develop using 400K-photoresist developer (mixed at a 1:4 ratio with water) with constant agitation about 1minute and rinse with DI.

9) Sputter wafer with 175nm platinum.

10) Soak wafer in acetone for 30 or more minutes to remove sacrificial photoresist and

unwanted platinum with “NO” agitation.

11) “Blast” wafer with acetone until small chunks of platinum can no longer be seen.

12) Clean and dry wafer and inspect under microscope.

13) If still a lot of unwanted platinum left on the wafer, soak or blast longer until it is gone.

14) Clean wafer when lift-off complete.

15) Anneal wafer in the vertical furnace at 650^{o}C.

2.7.5 SU8 2010 FABRICTION FOR 10μm HIGH MICRO-CHANNELS

1) Clean wafer with 5-step process. (Acetone, IPA, DI, Acetone, IPA, dry)

2) Place on hotplate at 200^{o}C for 5minutes to dehydrate wafer.

3) Spin on Omnicoat at 3000rpm for 30seconds.

4) Bake Omnicoat layer 1minute at 200^{o}C and let cool to room temperature.

5) Spin coats the wafer with SU8 2010 at 2000rpm for 30seconds for a 10μm thickness.

6) Soft bake the wafer at 65^{o}C for 1minute and immediately following 95^{o}C for 2minutes (on other hot plate).

7) Expose wafer using predetermined time for substrate and desired pattern (15 seconds for platinum patterns)

8) Post exposure bake for 2minutes at 65^{o}C and 2minutes at 95^{o}C, then again at 65^{o}C for 1minute.

9) Let wafer cool on workbench for 5minutes after baking to help prevent cracking.

10) Develop wafer using a full immersion in SU8 Developer for 3minutes with constant strong agitation.

11) Rinse with IPA, DI, and dry.

* SU8 Removal: Place wafer in Remover PG solvent at 80^{o}C about 30minutes or more to remove SU8 if needed.

2.7.6 SU8 2025 FABRICTION FOR 40μm HIGH MICRO-CHANNELS

1) Clean wafer with 5-step process. (Acetone, IPA, DI, Acetone, IPA, dry)
2) Place on hotplate at 200^{o}C for 5minutes to dehydrate wafer.
3) Spin on Omnicoat at 3000rpm for 30seconds.
4) Bake Omnicoat layer 1minute at 200^{o}C and let cool to room temperature.
5) Spin coats the wafer with SU8 2025 at 500rpm for 7seconds with acceleration of 100rpm/s. Check to see if any bubbles have occurred. If so, pop with razor blade.
6) Spin wafer for another 30seconds at 2000rpm with an acceleration of 300rpm/s for a 40μm thickness.
7) Soft bake the wafer at 65^{o}C for 2minute and 30seconds on hot plate.
8) Bake for one hour at 95^{o}C in convection oven.
9) Expose wafer using predetermined time for substrate and desired pattern (55seconds for 35μm channels and 65seconds for the rest of the wafer)
10) Post exposure bake for 1minutes at 65^{o}C and 4minutes at 95^{o}C on the second hot plate.
11) Let wafer cool on workbench for 10minutes after baking to help prevent cracking.
12) Develop wafer using a full immersion in SU8 Developer for 5minutes and 15seconds with constant strong agitation.
13) Rinse with IPA, DI, and dry.

CHAPTER 3

EXPERIMENTAL METHOD

3.1 EXPERIMENT OVERVIEW

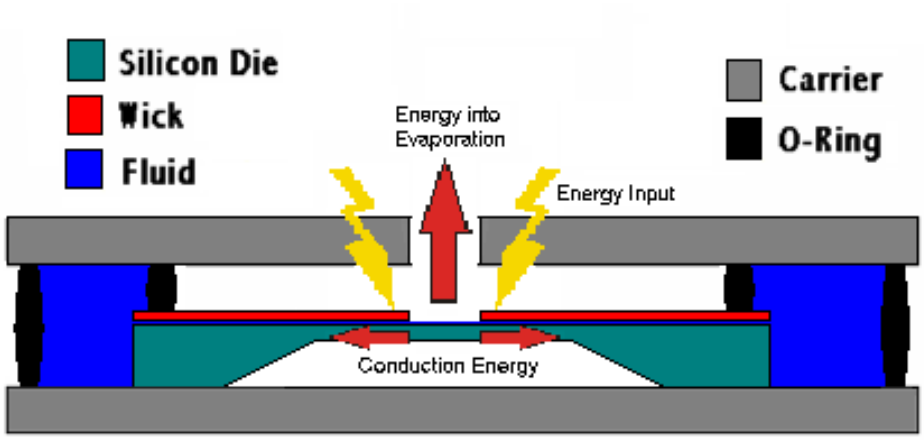


Figure 3.1 Schematic of the energy balance in the acrylic carrier

The goal of the experiments described in this dissertation is to determine the efficiency of each micro-channel evaporator through an energy balance. The schematic of the energy balance in the acrylic carrier is shown in Figure 3.1. The energy balance includes heat conduction from the micro-channel and electrical energy dissipated as heat in the center of the membrane across a micro-channel evaporator membrane, evaporation. Three different sets of experiments are performed in this work. The first set of experiments discussed is steady state evaporation experiments in which electrical power is continuously dissipated as heat in the micro-channel evaporator. The second set of experiments is the transient evaporation experiments in which electrical power dissipated as heat in the membrane is cycles on and off to result in transient operation. Calibration

of the temperature measurement devices, RTDs was required for both of these sets of experiment. Finally, the last set of experiments is the visualization of the liquid vapor interface in the capillary channels. The visualization test provides a view of the placement of liquid vapor interface, as well as the contact angle of liquid wall interface. The details of the experiment setup including equipment and experimental procedures are explained in next several sections.

3.2 HEATER/RTDs DESIGN, EVAPORATOR ASSEMBLY, RTD TESTING BOX

This experimental study was conducted using radial micro-channel evaporators fabricated on test die micro-machined from three-inch wafers (Si and SiNx). Each test die consisted of a circular platinum resistance heater, concentric annular platinum resistance thermometers (RTDs), and radial micro-channels.

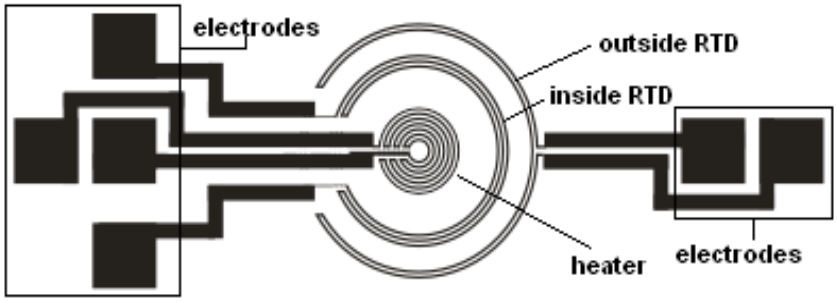
The test die is mounted in two 3"x3" acrylic carrier with an outer reservoir of working fluid contained between two O-rings. The working fluid is choose for its latent heat of vaporization value of 83.736J/g and value is determined at 25^{o}C with ambient pressure. The probes are bonded into the carrier for electrical contact to the heater and RTD's. The two O-rings control the working fluid flowing from the outer reservoir. Capillary forces pumped fluid from the outer reservoir under the inner O-ring and into the micro-channels covering the membrane. A schematic of the micro-channel evaporator die mounted in the acrylic carrier is shown in Figure 3.1. The die and acrylic carrier are placed on a scale with a precision of ±0.5mg. The major terms of interest in the experiments are the: (1) electrical power dissipated as heat in the central resistance heater, (2) sensible heat conducted radially outward through the membrane, and (3) latent heat carried away from

the membrane by the evaporation of the working fluid.

To accomplish the experiments performed in this work, the design of the concentric heater and RTDs need to be explained, and the evaporator needs to be assembled with the micro-channel evaporator carrier. The RTD box is used to convert the RTD resistance to the corresponding voltage. The design details of the heater and the RTDs as well as the assembly details are explained in the section 3.2.1 and section 3.2.2 respectively. The details of the RTD testing box are explained in the section 3.2.3.

3.2.1 HEATER/RTD DESIGN

Heat is added the center of the micro-channel evaporator through the use of the resistance heater. The concentric heater used in the experiment to adding heat to the micro-channel evaporator consisted of a platinum resistance heater. The platinum resistance heater is connected to the two inner electrodes for the connection of a power source as shown in Figure 3.1. The resistance of the platinum concentric heater is approximately 500Ω. The high resistance ensures that the heat is dissipated on the membrane and not in the resistance heater leads. The resistance is determined by its material resistivity, length, and cross-sectional area. The heater is made of 50μm wide lines with 50μm wide gaps in between and using an outside radius of 0.8mm. Heat transfer by conduction is determined the average temperature at two radial temperatures on the evaporator. These radial temperatures on the membrane are measured via two RTDs as shown in Figure 3.2. The overall heat conduction is calculated by assuming the radially symmetric heat conduction. The RTDs measure temperature with a change in resistance. The RTDs' typical resistances are between 500Ω and 1000 Ω.



3.2 Schematic of resistance heater and dual RTD

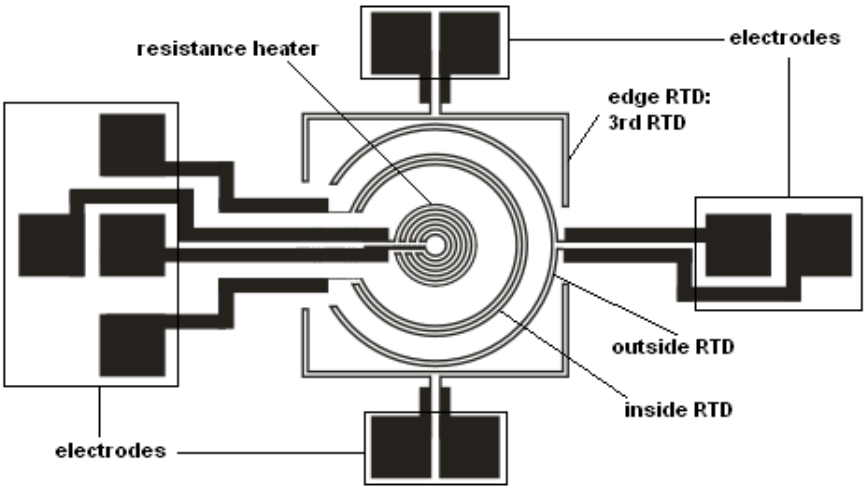


Figure 3.3 Schematic of resistance heater and triple RTD

The RTDs are connected to large electrodes to ensure that the leads do not affect the RTDs temperature measurements, thus the error can be minimized. The two concentric radial dual RTDs are designed with an inner radius of 1.7mm and outer radius of 2.4mm.

The ring features of the RTDs are designed as 50μm wide lines with 50μm wide gaps. The third RTD is placed at the edge of the defined membrane square (off of the membrane) and is 5mm by 5mm. The third RTD is intended to measure the temperature boundary condition at the edge of the evaporator membrane boundary as shown in Figure 3.3. The temperature measurement from this third RTD is then used to fix the temperature boundary condition in the numerical calculations.

3.2.2 EVAPORATOR ASSEMBLY

The Si die on which the micro-channel evaporator membrane and wicking structures are fabricated is mounted in a die carrier for the experiments. First, the die is checked to ensure that it is operational for the experiments. To do this, the resistance of each of the platinum resistance thermometers and the platinum resistance heater on the membrane are first measured to determine the resistance values of the RTDs and heater do not exceed acceptable limits. Second, the die is checked to verify that a full array of wicks covers the membrane.

After confirming that the membrane is operational, the micro-channel evaporator is mounted in the micro-channel evaporator carrier. The carrier is designed to hold the micro-channel evaporator in the working fluid, and seal the working fluid while providing electrical access to the heater and RTDs. PMMA acrylic is the material used for the micro-channel evaporator carrier in this work. The important features of the acrylic carrier are shown in Figure 3.4.

The carrier assembly consists of upper and lower acrylic micro-channel evaporator carrier plates which are squares of the size of 10cm by 10cm by 1/4” thick. The carriers contain drilled though-holes to provide electrical access to the heater and the RTDs for data collection. A central hole located above the membranes provides optical access for micro-channel visualization. Each carrier plate also contains through holes for alignment purposes. Machine screws at the corners provide the clamping force to seal the evaporator die in the working fluid. The micro-channel evaporator is first placed on semiconductor tape on the lower acrylic die carrier. A small O-ring is placed on top of membrane.

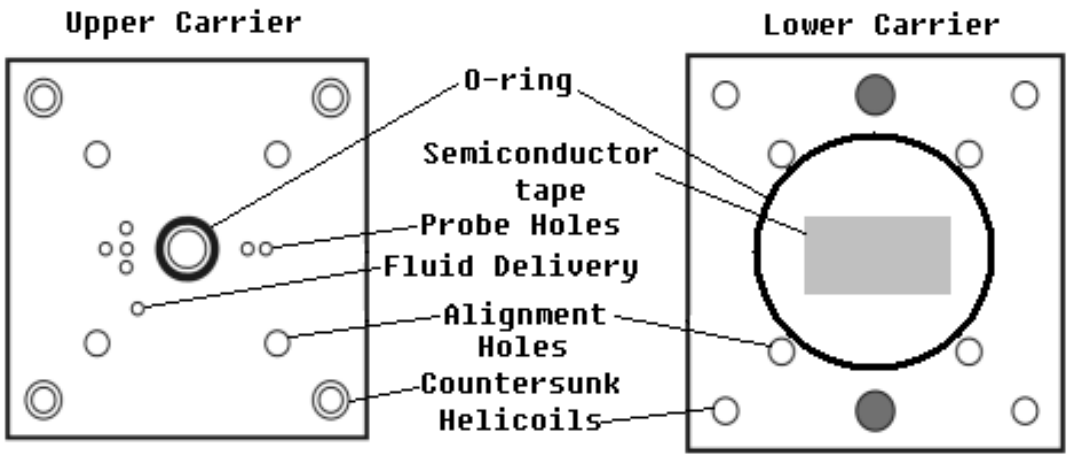


Figure 3.4 Descriptions of upper and lower acrylic die carrier components

A larger O-ring is placed on the lower carrier components to contain the liquid reservoir as shown in Figure 3.4. The upper carrier component is now positioned in place, ensuring that the probes line up with the corresponding electrodes. Once the carriers are aligned, the four screws are tightened to complete the assembly.

The screws are tightened to prevent a working fluid leakage and also provide good electrical connections between probes and electrodes. However, if the screws are too tight, the clamping force between carrier components can break the micro-channel evaporator membrane.

The electrode wires are checked to verify electrical connection after assembling all the components. If the resistance reading is stable and shows the initial resistance of the heater and RTDs, the micro-channel evaporator membrane and carrier assembly is complete. If the resistance reading is not stable or poor, the assembly is realigned. Figure 3.5 shows a completed assembly of the micro-channel evaporator membrane and the acrylic carrier.

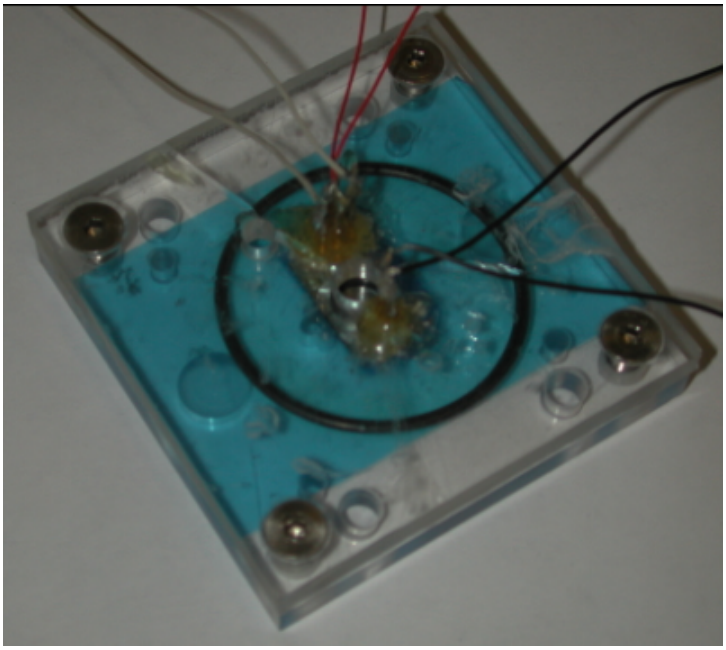


Figure 3.5 Completed assembly of micro-channel evaporator and the acrylic carrier

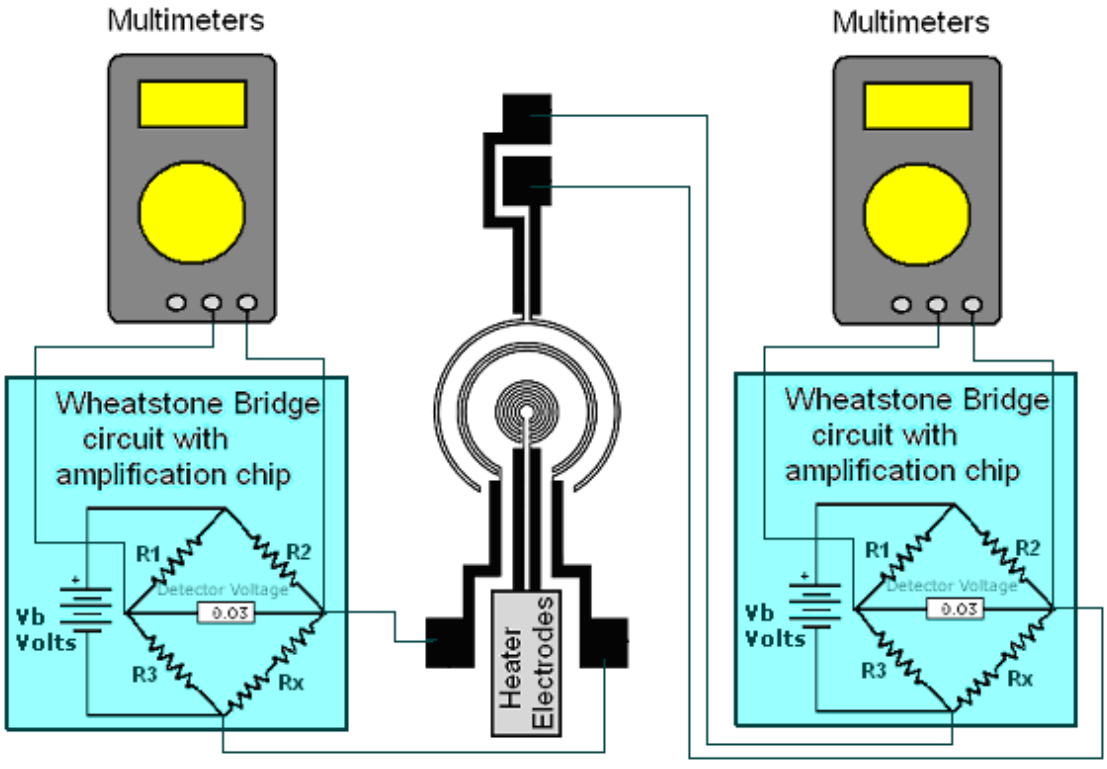


Figure 3.6 Schematics of RTD testing box, multimeters, and RTDs

3.2.3 RTD TESTING BOX

The RTD testing box consists of Wheatstone bridge circuit. The Wheatstone bridge circuit consists of four resistors, an input for a voltage signal from the RTDs, an output to the voltage gage, two potentiometers used to zero the circuit, an outside DC power input, an amplification chip and four nine volt batteries to power the amplification chip. The four resistors consist of two known resistors, one variable resistor, and one unknown resistor from RTD. The Wheatstone bridge circuit is balanced using the variable resistor and a multimeter to measure the voltage across the circuit. The multimeter reads zero when the circuit is balanced. If the resistance changes from any of the four resistors, a voltage change is displayed on the multimeter. The multimeter thus becomes an indicator of the balance of the circuit. The amplification chip amplifies the RTD signals. This is necessary because the signals from the RTDs are often smaller than the level of the noise in the system. Four nine volt batteries are used to power the amplification chip, and an AC power adapter with 1.5V output and 700mA is used to power the RTD box. Figure 3.6 shows a schematic of the Wheatstone bridge circuit connected to the RTDs and two multimeters.

3.3 RTD CALIBRATION

Before any experiments are run the RTD's on the micro-channel evaporators are calibrated. A series of calibration results are present in Appendix B.

The RTD calibration experiment setup consists of the micro-channel evaporator carrier assembly, two RTD testing boxes, a deionized water bath, a hot plate, a stir bar, an acrylic ring, a thermometer, and three multimeters. Figure 3.7 shows a photograph of the setup.

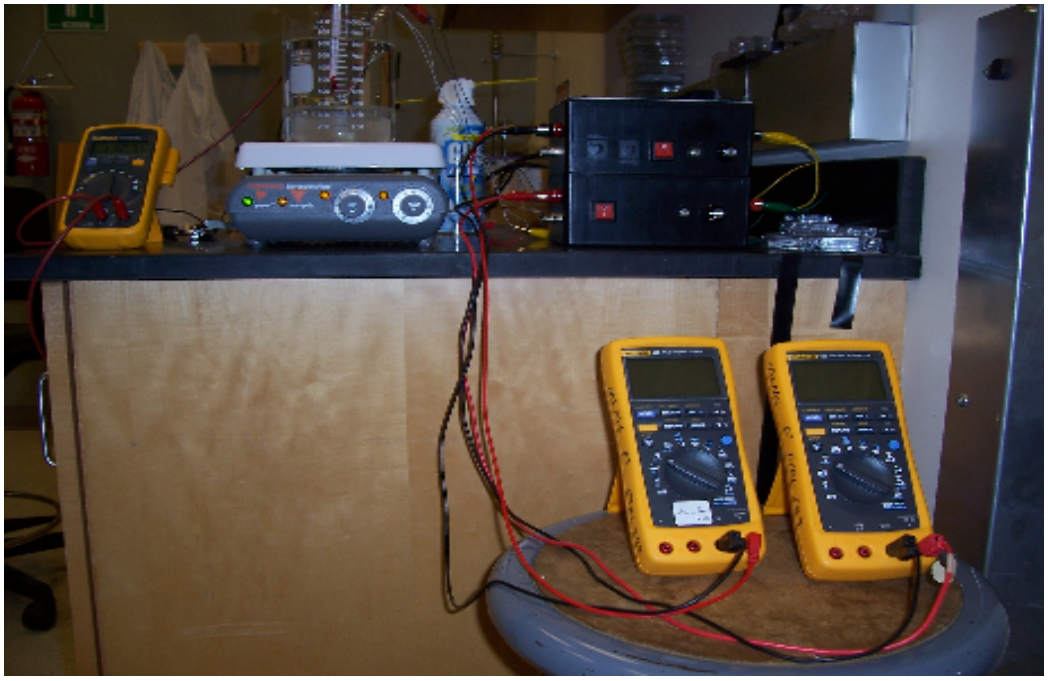


Figure 3.7 Picture of calibration experiment set up

First, the acrylic ring and stir bar are placed into the water bath. To maintain constant stirring without disruption of the micro-channel evaporator carrier, the magnetic stir bar is kept in the center of the acrylic ring. Second, with all connections checked, the micro-channel evaporator carrier is immersed into the water bath and centered on the acrylic ring. Since the membrane is 2.2μm thick and very fragile, dropping the micro-channel evaporator carrier can be cause the membrane to break.

Next, the water bath is placed onto the hotplate, and the four RTD wires are connected to the RTD box. The RTD box outputs are connected to multimeters. The heater electrode wires are directly connected to a third multimeter. The third multimeter measures the voltage change across the heater during the calibration. A thermometer is placed into the water bath to measure the water temperature during the calibration.

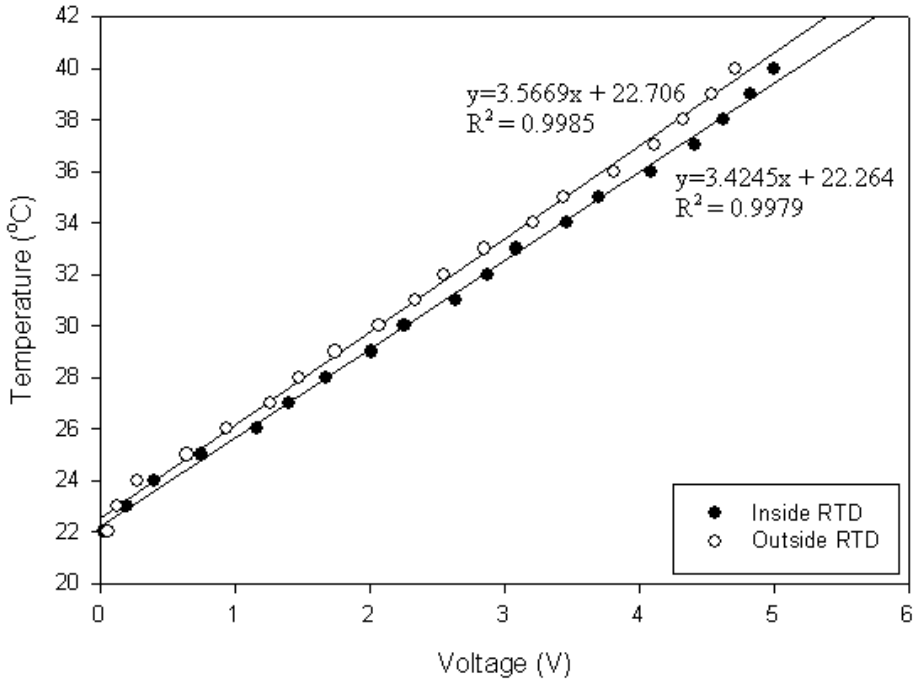


Figure 3.8 Dual RTD calibration curve

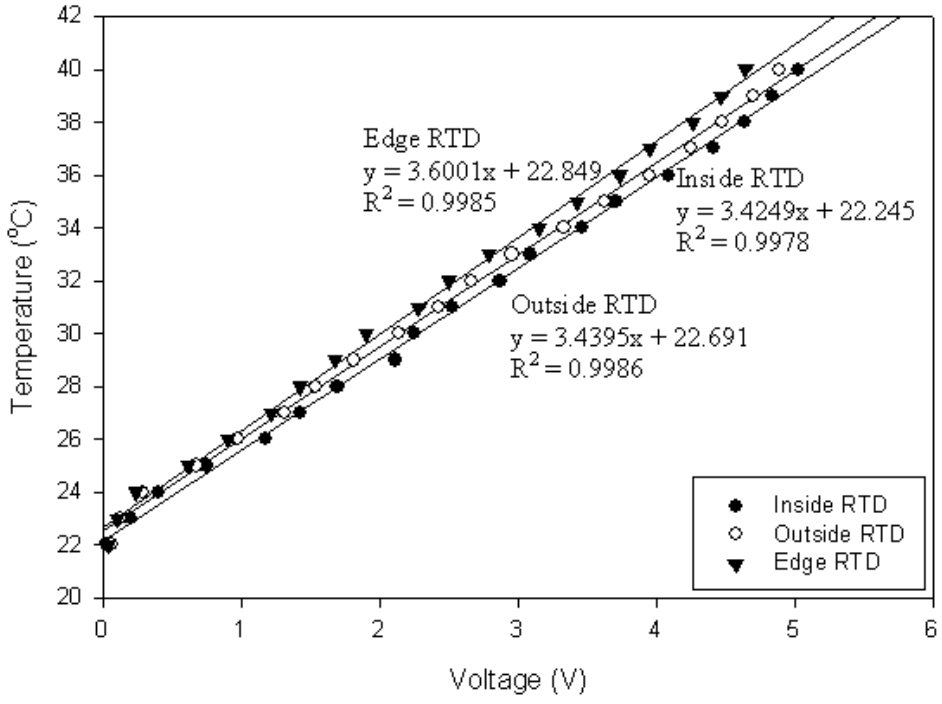


Figure 3.9 Triple RTD calibration curve

With the setup complete, the initial water temperature and voltage values are recorded. The hotplate and stir bar are then turned on. Slow stirring and slow heating is desired to decrease the electrical noise in the measurements and to avoid physical disturbance of the die carrier. Doing so decreases the variance in the calibration and the uncertainties in the RTD measurements. The temperature of the water bath and multimeter display values are checked regularly. Measurements are recorded at every degree of temperature increase, over a range from 20^{o}C to 45^{o}C. The results are plotted as an RTD calibration curve of water bath temperature versus the voltage from the Wheatstone bridge circuit. The measurements are voltage measurements. The least squares linear measurements versus voltage measurements fit is used to correlate the temperature. The desired R^2 value is at least 0.99. Figure 3.8 and Figure 3.9 show typical dual RTD calibration curves and typical triple RTD calibration curves respectively.

3.4 STEADY STATE EVAPORATION TESTS

An experiment starts by introducing the Fluorinert FC77 working fluid at the outer circumference of the radial channels. Capillary forces wick the working fluid from the outer circumference into the center of the membrane. Electrical power is dissipated as heat in the central thin-film resistance heater. Sensible heat then conducts radially outward through the membrane, driving liquid working fluid to evaporate from the radial channels. An energy balance is then experimentally determined for the micro-channel evaporator test die. The major terms of interest in the energy balance are the: (1) electrical power dissipated as heat in the central resistance heater, (2) sensible heat conducted radially outward through the membrane, and (3) latent heat carried away from

the membrane by the evaporation of the working fluid. In the present set of experiments introduced in this dissertation, 30mW to 60mW of electrical power input is dissipated in the thin-film heater at the center of the testing membrane of the micro-channel evaporator. The power dissipated as heat in the thin-film heater is determined by measuring the voltage drop across the platinum resistance hater and the voltage drop across a power resistor in series with the heater. The sensible heat conducted radially outward through the membrane is determined by using the temperatures from the two-concentric annular RTDs on each micro-channel evaporator. Conduction heat transfer out of the membrane is assumed to be radially symmetric so the RTD measurements are used with the cylindrical conduction heat transfer equation to calculate the heat flux conducted across the membrane. Evaporation tests are performed to account for the balance of energy, in the micro-channel evaporator: the electrical energy dissipated as heat in the center of the micro-channel evaporator, the energy conducted as sensible heat across the membrane, and the energy as latent heat carried away from the membrane by evaporation of working fluid. The efficiency of a micro-channel evaporator is then determined from this energy balance. The evaporation test setup consists of the micro-channel evaporator, the carrier assembly, the RTD boxes, multimeters, power resistors, a power supply, Fluorinert refrigerant (FC77) working fluid, a scale, and a timer.

An evaporation test begins by placing the carrier assembly on the scale. The RTDs are attached to the RTD box inputs. The RTD box outputs are connected to multimeters. The multimeters are zeroed at room temperature. The resistance heater is connected to the power supply in series with the power resistor (50W 10.3Ω). The FC77 working fluid is added to the micro-channel carrier, to fill the liquid reservoir.

Figure 3.10 Evaporation experiment test set up

The initial mass of the system is recorded. Capillary forces pump fluid from the outer reservoir, under the inner O-ring and into the micro-channels covering the membrane.

The power supply is turned on to the desired voltage setting. The measurements of the mass of system, RTD outputs, and voltage across the heater and power resistor are recorded in five minute intervals for 30-45 minutes (at least 30 minutes). Figure 3.10 shows a photograph of the evaporation experiment test setup.

The temperatures taken by the two annular RTDs are used to measure temperatures at two radial positions on the membrane. Assuming radial symmetry, the conduction heat transfer radially out of the membrane, Q_s is then:

$$Q_s = \frac{2\pi hk(T_i - T_o)}{\ln(r_o / r_i)} \qquad (3.1)$$

Here r_o and r_i are the radii of the outer and inner RTDs (2.4mm and 1.7mm), k is the thermal conductivity of silicon (k=153W/mK) or silicon nitride (k=15W/mK), h is the thickness of the silicon membrane (h=2μm) or silicon nitride (h=300nm), and (T_i-T_o) is the difference of inner and outer RTD temperatures.

The voltage across the heater and power resistor is used to calculate the power input to the system from Ohm's Law.

$$V = IR \qquad (3.2)$$

$$P = VI \qquad (3.3)$$

In equations 3.2 and 3.3, V is voltage, I is current, R is resistance, P is power. The latent heat carried away by the evaporation of the working fluid is determined gravimetrically. The mass of the working fluid that has evaporated from the micro-channel evaporator die is found from the decrease in mass of the entire set-up as measured by the scale, over the course of an evaporation experiment. The latent heat transfer rate from the micro-channel evaporator is taken to be:

$$Q_l = jh_{fg} \qquad (3.4)$$

where j is the rate of mass transfer by evaporation of working fluid from the micro-channels, and h_{fg} is the latent heat of evaporation of the working fluid, 83.736J/g.

The average power into evaporation is added to the conduction power and balanced with the total heat input to complete the energy balance. The schematic of the energy balance in the acrylic carrier is shown in Figure 3.1.

The energy balance error is taken as the difference between the input power and the outgoing power which includes the heat transfer rate by conduction and the heat transfer rate by evaporation, divided by the total power input.

$$\%Error = \left(\frac{P_{in} - P_{cond} - P_{evap}}{P_{in}} \right) \times 100 \qquad (3.5)$$

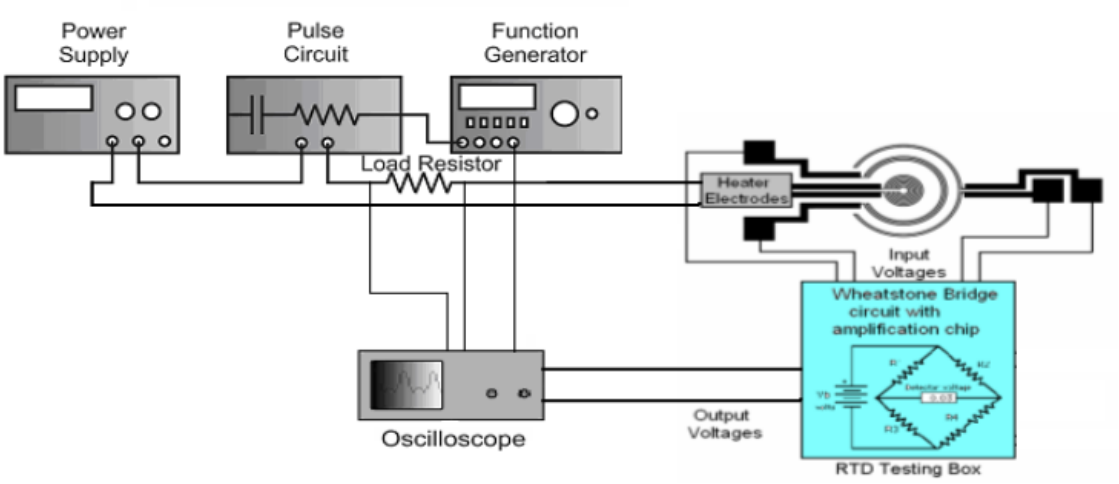
In equation 3.5, P_{in} is the total power input, P_{cond} is the sensible heat transfer rate by conduction, P_{evap} is the latent heat transfer by rate by evaporation.

The micro-channel evaporator efficiency, η_{wick}, is defined to be the ratio of latent heat transfer rate by evaporation to the total power input:

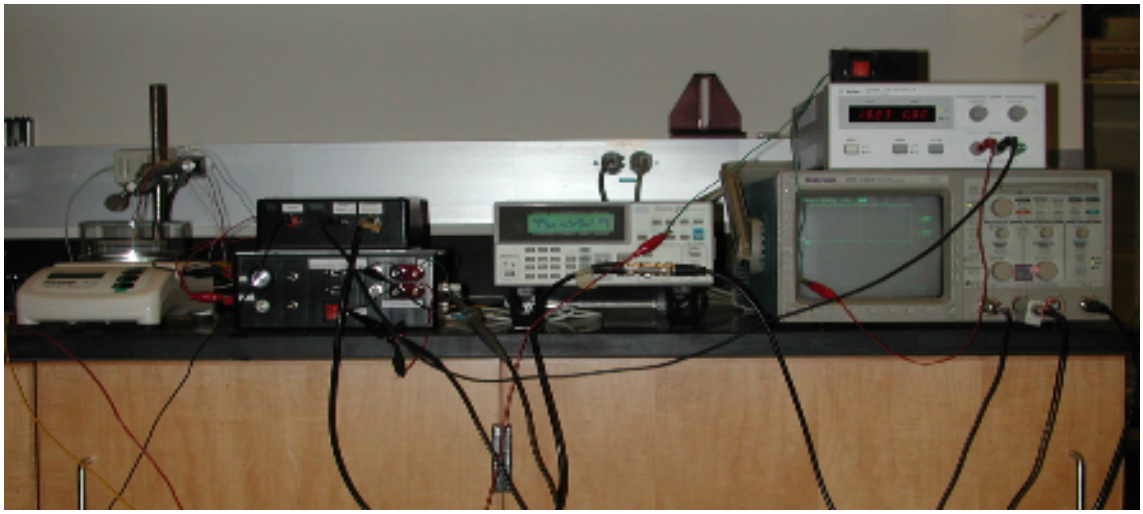
$$\eta_{wick} = \frac{P_{evap}}{P_{in}} \qquad (3.6)$$

3.5 TRANSIENT EVAPORATION TESTS

The transient evaporation experiments are very similar to the steady state experiments. An experiment starts by introducing the Fluorinert FC77 working fluid at the outer circumference of the radial channels. Capillary forces wick the working fluid from the outer circumference into the center of the membrane. Pulsed electrical power is dissipated as heat in the central thin-film resistance heater. Power or heat is pulsed to the testing membrane by applying a periodic voltage across the resistance heater. Sensible heat then conducts radially outward through the membrane, driving liquid working fluid to evaporate from the radial channels. An energy balance is then experimentally determined for the micro-channel evaporator test die. The major terms of interest in the energy balance are the: (1) electrical power dissipated as heat in the central resistance heater, (2) sensible heat conducted radially outward through the membrane, and (3) latent heat carried away from the membrane by the evaporation of the working fluid. In the present set of experiments introduced in this dissertation, 30mW to 60mW of pulsed electrical power input is dissipated in the thin-film heater at the center of the testing membrane of the test die. The power dissipated as heat in the thin-film heater is determined by measuring the voltage drop across the platinum resistance hater and the voltage drop across a power resistor in series with the heater. The sensible heat conducted radially outward through the membrane is determined by using the temperatures from the two-concentric annular RTDs on each micro-channel evaporator. Conduction heat transfer out of the membrane is assumed to be radially symmetric so the RTD measurements are used with the cylindrical conduction heat transfer equation to calculate the heat flux conducted across the membrane.



(a) Schematic of transient experiment set up



(b) Photograph image of transient experiment set up

Figure 3.11 Schematic and photograph of transient experiment set up

Schematic and photograph views of the experiment setup are as shown in Figure 3.11. (a) and (b). Again, a power resistor (50W 10.3Ω) is connected in series with the resistance heater to determine the current flow through the heater. A function generator is used to determine the shape of the wave and control the frequency and duty cycle of the pulses. In the present experiments, a square wave is used. A DC power supply is used to

control the amplitude. In the steady-state experiment a DC power supply is connected directly across the resistance heater and power resistor (50W 10.3Ω). In the transient experiment a pulse circuit is used to regulate the current flow based on the signal supplied from the function generator. An oscilloscope is then used to measure the signals from the RTDs and the periodic electrical input to the resistance heater.

A schematic of the pulse circuit is shown in the Figure 3.12. The pulsed signal is made possible by connecting a pulse circuit in series with the heater and power supply. The function generator is connect to the TTL(+) and GND. A function generator is used to control the pulse parameters: frequency and duty cycle.

A range of frequencies and duty cycles (10Hz to 40Hz of frequency and 50% to 10% duty cycle) are used to test the heat conduction response of the membrane while the evaporation rate is recorded from the scale. In the transient experiment, the time domain must be considered when calculating the heat conduction, Ohm's Law and latent heat transfer rate. Those results are then used to derive the energy balance. In the transient experiment the energy measurement unit is in Joules (J) because the experiment is time dependent whereas the steady state experiment uses power measurement Watts (W). In this work, several different resonant frequencies are tested: 10Hz, 20Hz, and 50Hz. Duty cycles of 50%, 30%, and 10% are used. Power input was hold constant as frequency and duty cycle was changed. For example, 34mW at frequency of 10Hz with 100% duty cycle can be compared with 68mW at 10Hz frequency with 50% duty cycle. In this example the 100% duty cycle with 34mW of power result in a total energy input of 3.4mJ per pulse. Likewise the total energy input for the 50% duty cycle at 64mw is also 3.4mJ per pulse.

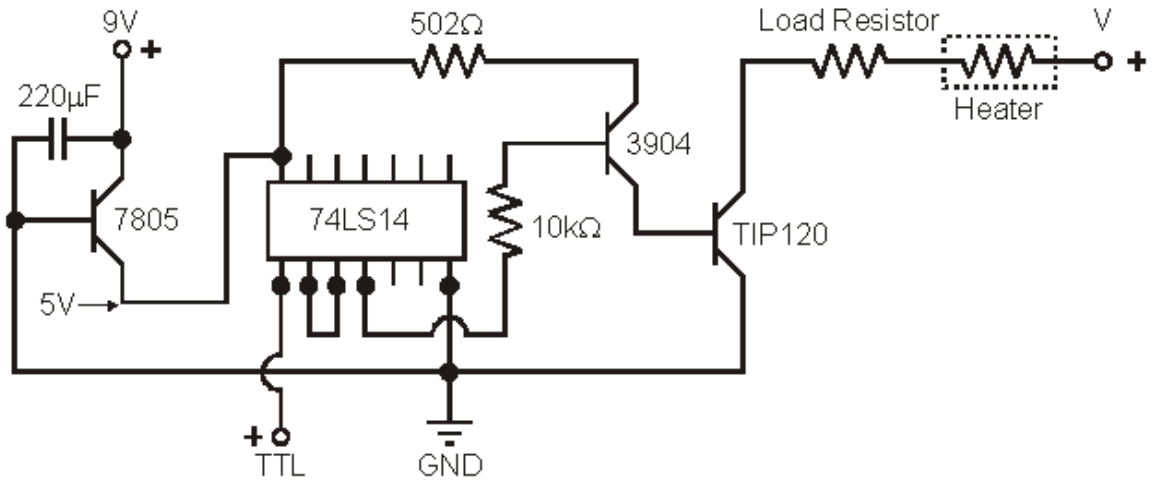


Figure 3.12 Schematic of pulse Circuit, Courtesy S. Whalen

3.6 VISUALIZATION

The position of liquid working fluid within the micro-channel evaporations is visualized two ways. First experiment the movement of the liquid interface for working fluid is tracked in the micro-channel evaporator at different power inputs. Second the meniscus contact angles for the liquid working fluid in the micro-channels is visualized.

In the first visualization experiments, TSI Particle Image Velocimetry (PIV) camera is connected to a Questar QM1-10126-MKIII long distance microscope, as shown in Figure 3.14. The camera is placed and mounted on a sled along with the optical microscope. The micro-channel evaporator carrier holder is assembled on another sled. A 45degree-angled mirror is mounted on the sled to visualize the membrane from beneath. A fiber optic light source is used to illuminate the micro-channel evaporator and liquid working fluid.

The focal length of the microscope is determined before any visualization experiments are done. A 4-point font number printout is taped to a square piece of acrylic and set in the carrier holder. The printout is lit from above with the fiber optic light.

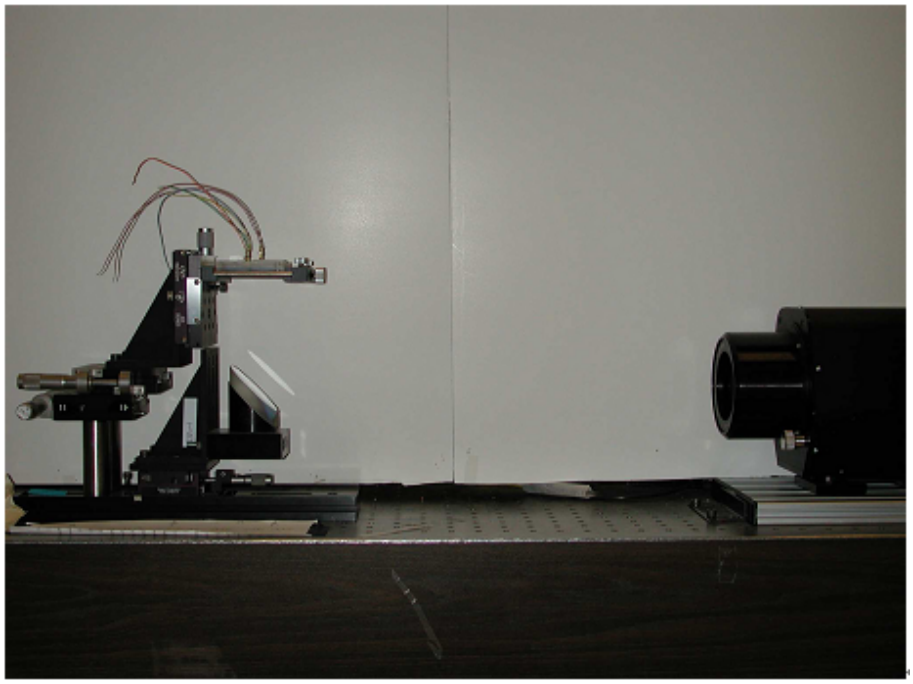


Figure 3.13 Photographic image of visualization experiment set up

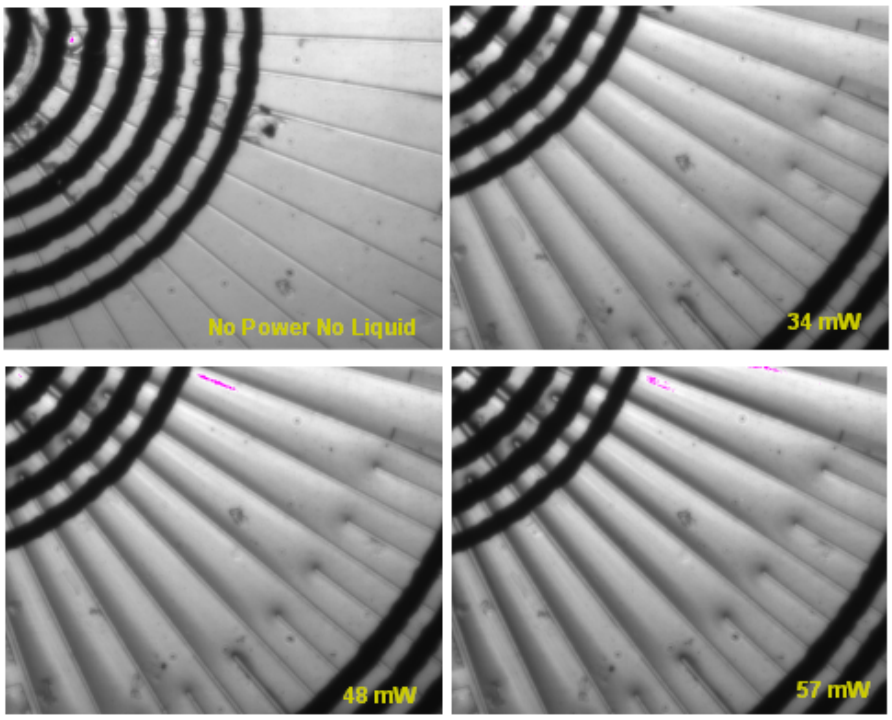


Figure 3.14 Visualization images from the experiment set up shown in Figure 3.13

The camera collects images. The Exposure Mode “Free” and Capture Mode “Continuous” of the Insight data collection software are used. The sled is moved until the number printout is in focus.

Once the camera is focused, the micro-channel evaporator carrier is placed on the carrier holder. The fiber optic light source is placed directly above the membrane. Yellow tissue paper is used to diffuse the intensity of the light source. The resistance heater is connected to the power supply. The micro-channels and reservoir are filled with the working fluid. Several images of the working fluid movement as the power to the micro-channel evaporator are varied (30mW ~ 60mW). Typical images of 40μm high and 70μm wide micro channels are shown in Figure 3.14.

The meniscus formed by the working fluid in the micro-channels is visualized using a digital camera coupled to a microscope. In particular, the liquid meniscus and liquid contact angles are captured. In this second setup a digital camera is coupled to a microscope. No power supply is connected to the heater electrodes. The schematic and photographic images of the second visualization experiment setup are shown in Figure 3.15. A typical image is shown in Figure 3.16. The testing die is mounted on a sled that allows the testing die to stand at a 90degree angle. A Fiber Lite High Intensity Illuminator is used to light the membrane from above.

This microphotography technique is especially useful to visualize the meniscus contact angles in the micro-channels. Pictures are taken of the contact angles and are measured those contact angles using a protractor. The accuracy of measurements made in this way is verified by measuring a silicon-etched pit with a known angle. This silicon-etched angle (54.75degree) is shown in Figure 3.16. The angle is measured to be 54.75^{o}.

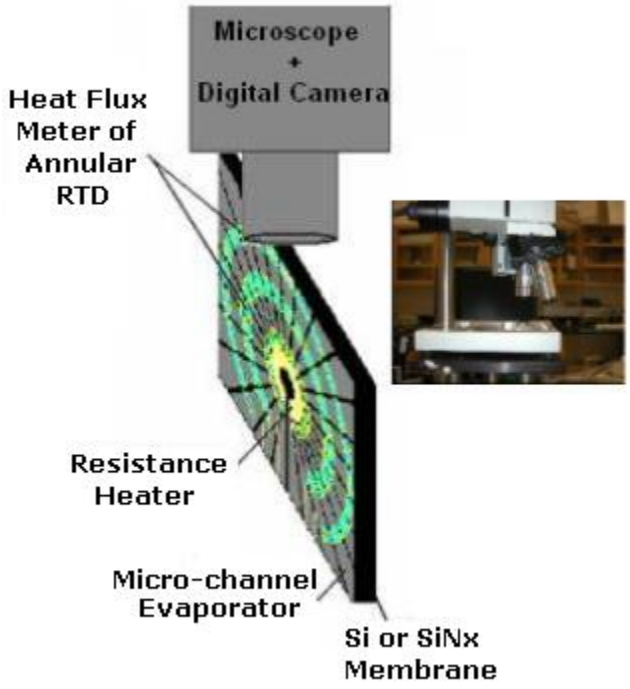


Figure 3.15 Schematic and photographic images of visualization experiment setup

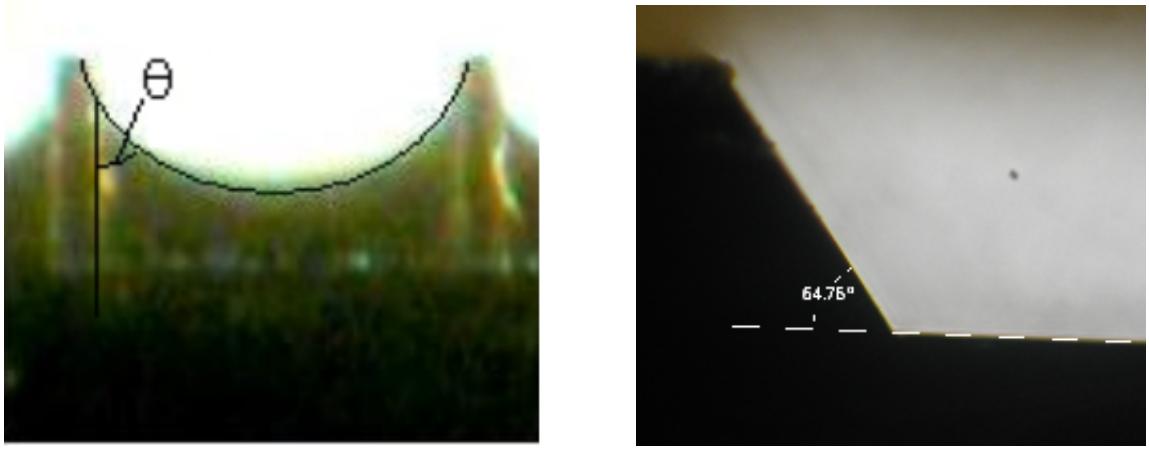


Figure 3.16 A typical meniscus image and the silicon etched angle image

3.7 UNCERTAINTIES

Uncertainties for the principle measurements taken during experiments are now presented. The principle measurements include the power input, the heat transfer rate across the membrane, and the evaporation rate.

3.7.1 UNCERTAINTY OF POWER INPUT

The uncertainty of the electrical power dissipated as heat in the resistance heater at the center of the micro-channel evaporator is considered first. The power dissipated in the resistance heater is measured with two Multimeters, a load resistor, and an oscilloscope. The Fluke 189 multimeter used to measure resistance of the resistor (shown in Figure 3.11 (a)). The voltage drop across the load resistor is measured with a multimeter during the steady state experiments. A Tektronix TDS-224 digital oscilloscope is used to measure the voltage drop across the load resistor during transient state experiments.

The current flow through the load resistor is found from Ohm's Law,

$$\Delta V_{load} = I_{ce} R_{load} \qquad (3.7)$$

where V is voltage drop of the load resistor, I is current of load resistor, and R is the resistance of the load resistor.

The uncertainty of the current flow through the load resistor is first determined to calculate the uncertainty of the power input.

The power input is calculated with Ohm's Law,

$$P = IV \tag{3.8}$$

where P is power, I is current, and V is voltage of heater.

The method of Kline and M^{c}Clintock [69] is used to determine the uncertainty of the current and the power input.

$$W_R = \sqrt{\sum_{i=1}^{n}\left(\frac{\partial F}{\partial x_i} w_{xi}\right)^2} \tag{3.9}$$

where W_R is the uncertainty, F is the function of interest, x_i is the parameter that has an uncertainty of W_{xi} , and n is number of contributing variables [70].

If the mutimeter measures the resistance of the load resistor to be 10Ω, the uncertainty for this measurement is ±0.005Ω. If the voltage drop across the load resistor is measured to be 6V with the multimeter or the oscilloscope, the uncertainty of this voltage drop is ±0.003V. An uncertainty of the current is calculated using the typical value of ΔV_{load}=6V and R_{load}=10Ω by substituting equation 3.7 into equation 3.9.

$$W_{I_{cd}} = \sqrt{\left(\frac{1}{10\Omega}(0.003V)\right)^2 + \left(\frac{6V}{10\Omega^2}(0.005\Omega)\right)^2} = 0.0004A \tag{3.10}$$

A typical value for current, I_{ce}= 0.6A, is found by using equation 3.8 and again with ΔV_{load}=6V and R_{load}=10Ω. The uncertainty of instantaneous power is calculated using typical values of ΔV_{heater}=4V and I_{ce}= 0.6A by substituting equation 3.8 into equation 3.9.

$$W_{I_{cd}} = \sqrt{((0.6A)(0.003V))^2 + ((4V)(0.0004A))^2} = 0.002W \qquad (3.11)$$

The ±2mW uncertainty on P (P= $I_{ce}\Delta V_{heater}$ =2.4W) corresponds to 0.10% of the measurement. The error does not change appreciably during over the range of power inputs applied during the experiments.

3.7.2 UNCERTAINTY OF HEAT TRANSFER RATE ACROSS MEMBRANE

Heat transfer by conduction radially out of the membrane is determine by applied Fourier's Law in cylindrical coordination and using the two radial temperature measurements supplied by the two concentric RTDs. The uncertainty in this heat transfer measurement is calculated as follows. First, the uncertainty of the two RTD measurements is calculated from the calibration data from the RTDs.

The residual (R) temperatures between the thermometer and RTD calibrations are used. The residual temperature difference between the calibration results and the measured temperature is calculated first.

$$R = x_i - x_m \qquad (3.12)$$

The average residual (R_{avg}) and the sum of the squares (SS) are also found in equation 3.13 and 3.14.

$$R_{avg} = (\sum (x_i - x_m))/n \qquad (3.13)$$

$$SS = \sum (x_i - x_m)^2 \qquad (3.14)$$

The standard deviation (σ) is determined by using the number of calibration measurements taken (n).

$$\sigma = \sqrt{\frac{SS}{(n-1)}} \qquad (3.15)$$

Number of data	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Water Bath T(C)	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Inside RTD (V)	0.14	0.44	0.63	0.86	1.18	1.52	1.72	1.87	2.19	2.47	2.74	2.91	3.15	3.31	3.67	3.77
Outside RTD2 (V)	0.08	0.35	0.53	0.77	1.09	1.31	1.53	1.76	2.07	2.33	2.56	2.78	3.03	3.2	3.56	3.68

Table 3.1. Typical values of the RTD calibration data

For example, R_{avg} and σ are 0.6°C and 0.8°C respectively are the values found from Table 3.1 for RTD 2 when calibration relationship is temperature T=4.1442x+26.528.

To find the measurement error ($error_{act}$), a t-value (1.96) for small sample sizes and a 95% confidence interval are used. The upper and lower confidence intervals are found by the equation 3.12.

$$error_{act} = R_{avg} \pm t\frac{\sigma}{\sqrt{n}} \tag{3.16}$$

where x_i is temperature calculation, x_m is measured temperature, and n is the number of measurements.

The upper and lower bound for each RTD is used to calculate for the maximum error possible in the power calculation across the membrane. The uncertainties of the indiviual temperature measurements are added together when calculating the temperature differential across the membrane. The maximum acceptable uncertainty for these measurements is taken as $\pm 0.5\ ^{o}C$.

The RTD masks are printed on the transparency mask at Washington State University, and the printing dimensions are limited to 20μm. Less than 1% error occurs in this process, so this is not considered as a contributor to overall uncertainty.

3.7.3 UNCERTAINTY OF EVAPORATION RATE

The evaporation rate from the micro-channel evaporator is determined gravimetrically. The uncertainty in this measure of evaporation rate is determined. An Acculab scale is used to measure the mass of the micro-channel evaporator carrier assembly and working fluid. As electrical power is dissipated as heat in the resistance heater at the center of the micro-channel evaporator, working fluid evaporates and the mass of carrier assembly decreases. A typical mass evaporation versus time is shown in Figure 3.17.

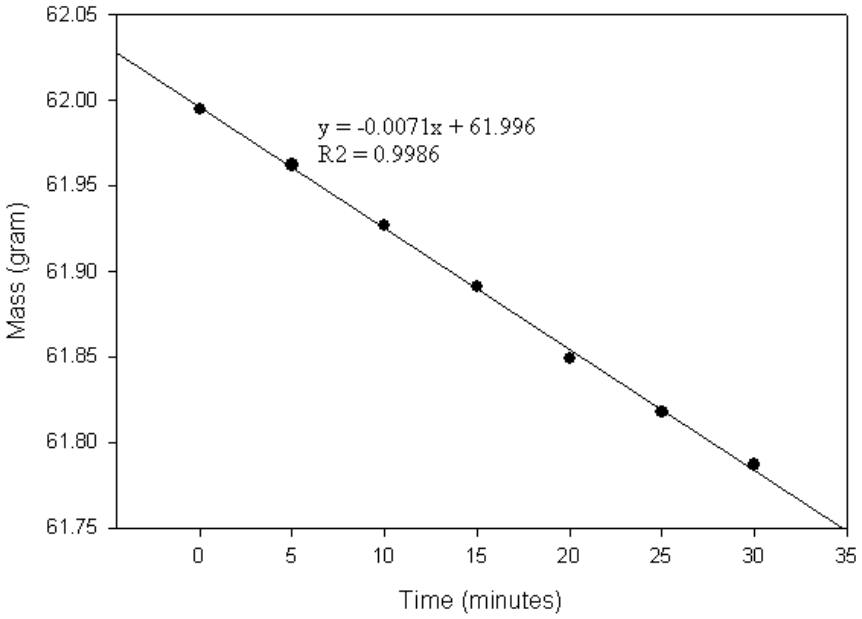


Figure 3.17 A typical evaporation test graph: mass change over time

The uncertainty in the measured evaporation rate is calculated using a linear regression technique. First the mass data and time date are used to find the line fit y = mx + b. The sums (S_{xy}, S_{yy}, S_{xx}) are calculated to find the slope (m).

$$S_{xy} = \sum (x - x_{avg}) \cdot (y - y_{avg}) \qquad (3.17)$$

$$S_{yy} = \sum (y - y_{avg})^2 \qquad (3.18)$$

$$S_{xx} = \sum (x - x_{avg})^2 \qquad (3.19)$$

where *x* is time data, *y* is mass evaporation data, and *avg* is the average value of data.

$$Slope = m = \frac{S_{xy}}{S_{xx}} \qquad (3.20)$$

The intercept (b) is found.

$$Intercept = b = y_{avg} - (m \cdot x_{avg}) \tag{3.21}$$

The results for slope and intercept for typical the case shown in Figure 3.18 are 0.0071 and 61.996. The uncertainty of the slope is calculated using equation 3.22

$$S_r = \sqrt{\frac{S_{yy} - (m^2 \cdot S_{xx})}{(N-2)}} \tag{3.22}$$

where S_r is the uncertainty of the slope, and N is number of data taken.

The uncertainty of the evaporation rate is $S_r = \pm 0.07$mg/min for a typical case shown in Figure 3.18.

The drift of the scale over time is determined. By placing a calibration weight, of 57.8 mg, on the scale. The mass is measured every 15 minutes for two hours. The drift is then the slope of the least squares line fit through the mass versus time plot. The drift of the scale found in this way is ±0.16mg/min. This drift rate is seen to be the dominant error in measurements of evaporation rate.

CHAPTER 4

NUMERICAL METHOD

4.1 NUMERICAL MODEL OVERVIEW

A FORTRAN based numerical model is developed and used to simulate heat and mass transfer from a micro-channel evaporator. The model is developed as a design tool for the development of efficient micro-channel evaporators.

The model in this work predicts the energy balance with in an evaporator. The energy balance is includes the conduction heat transfer across the membrane and latent heat transfer due to the evaporation of working fluid. In addition the model predicts working fluid flow rate and working fluid thickness inside the micro-channels.

The numerical model calculates conduction heat transfer using axisymmetric finite difference time domain (FDTD) integration. Mass transfer due to evaporation from the micro-channels is calculated using conservation of energy at the liquid-vapor interface. Fluid flow through the micro-channel is calculated using force balance in the channel and Yong-Laplace equation. The geometry, discretization, derivation of governing equations and calculations, and boundary conditions are discussed below:

4.2 GEOMETRY

A 2.2μm silicon thickness, 80nm silicon oxide thickness, and 40μm wicking structure thickness are used for the rectangular micro-channel simulation cases.

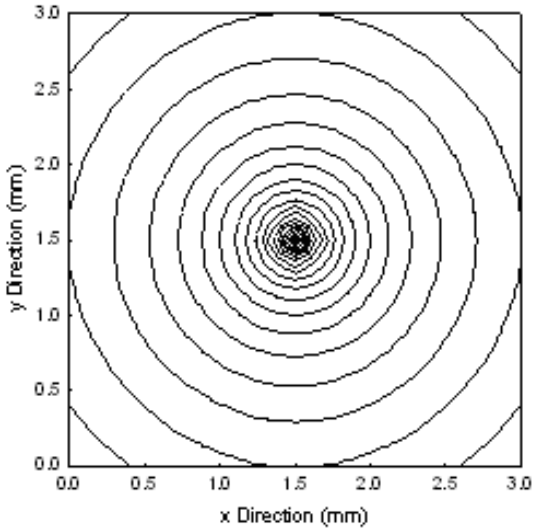


Figure 4.1 Top view of arbitrary temperature contour plot for a rectangular membrane,

Courtesy Scott Whalen

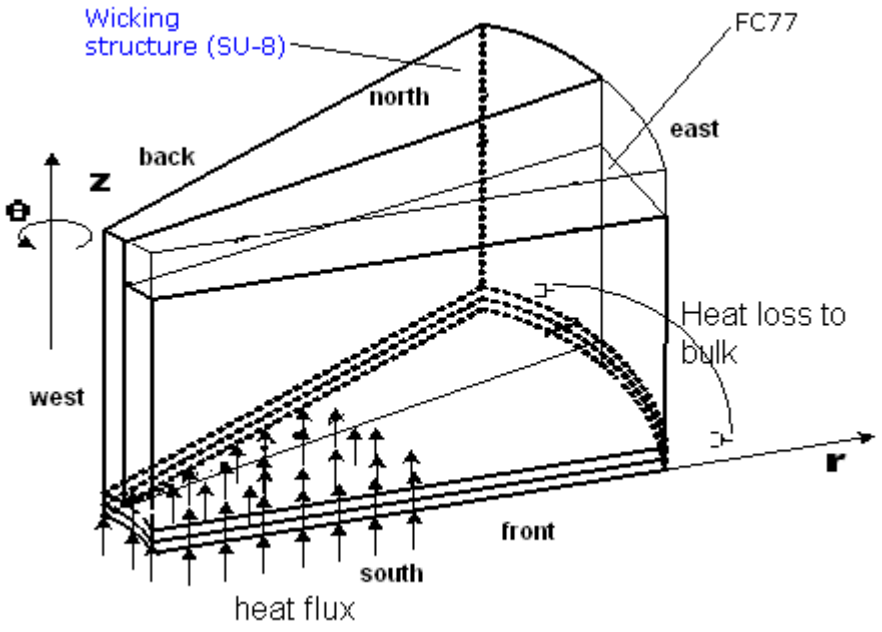


Figure 4.2 General schematic of 3-D axisymmetric geometry

Channel width varied for 35μm, 50μm, and 70μm. The varied liquid thicknesses (14μm, 20μm, 28μm, and 35μm) are tested to find liquid thickness of the experiments.

For the tapered micro-channel cases, a 2.2μm silicon thickness, 80nm silicon oxide thickness, and 40μm wicking structure thickness are used. The 40μm liquid thickness is initially used while outer edge starts at 80μm and narrows to 5μm wide. Then liquid thickness is calculated in the tapered micro-channel cases. The platinum thin film concentric heater and RTDs are not included in the numerical model in order to decrease computational time.

A 3-D axisymmetric geometry is assumed instead of modeling the whole evaporator geometry to decrease the computation time. The 3-D axisymmetric geometry assumption is built based on the previous 2-D numerical work [9]. This 2-D numerical work is developed and verified. Figure 4.1 illustrates a temperature contour plot for a 3mm rectangular membrane where concentric circles indicate lines of arbitrary constant temperature [9]. The heat transfer behaves radially for square membranes except near the far corners as shown in Figure 4.1. The control volume of interest is shaped as a wedge because the heat transfer behaves radially. Figure 4.2 shows the 3-D axisymmetric geometry with the membrane, working fluid, and wicking structure is indicated.

An effective radius for 3-D axisymmetric model is calculated since the model approximates a square membrane, with a circular membrane. The effective radius r_o is taken to be the radius of a circle that has an equivalent area to the surface area for a square membrane. The effective radius r_o is determined as follow Equations 4.1 and 4.2.

$$A_c = \pi r_o^2 = A_s = L^2 \tag{4.1}$$

$$r_o = \sqrt{\frac{L^2}{\pi}} \tag{4.2}$$

where A_c is the circular area, A_s is the surface area for a square membrane, L is the edge length for square membrane, and r_o is the effective radius.

A typical square membrane used in this work is L=5mm. The effective radius r_o=2.84mm.

4.3 DISCRETIZATION

The 3-D axisymmetric geometry is discretized into a grid of elements. Each element is wedge shaped. A non-uniform grid is used. A coarse mesh in the vertical (z) direction and a fine mesh in the horizontal (r) direction are used. This is due to the high aspect ratio of the evaporator, an aspect ratio approximately 1:100.

The meshed model is shown in Figure 4.3. The non-uniform grid is required to save the integration time while keeping the accurate calculation results. A grid independent study is prepared to determine the optimal number of elements needed for an accurate calculation for this numerical integration.

A grid independence study is completed by increasing the number of elements in all three directions until the solutions converge. Typical solutions for temperature at RTD locations are shown for several combinations of the number of elements in Figure 4.4 (a) and (b).

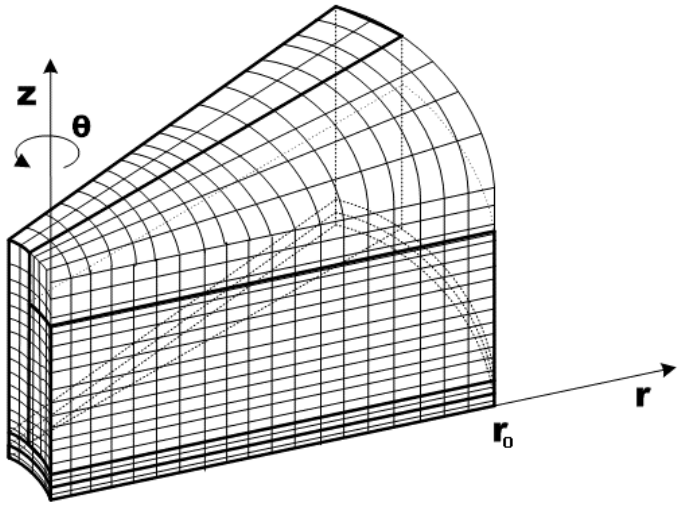
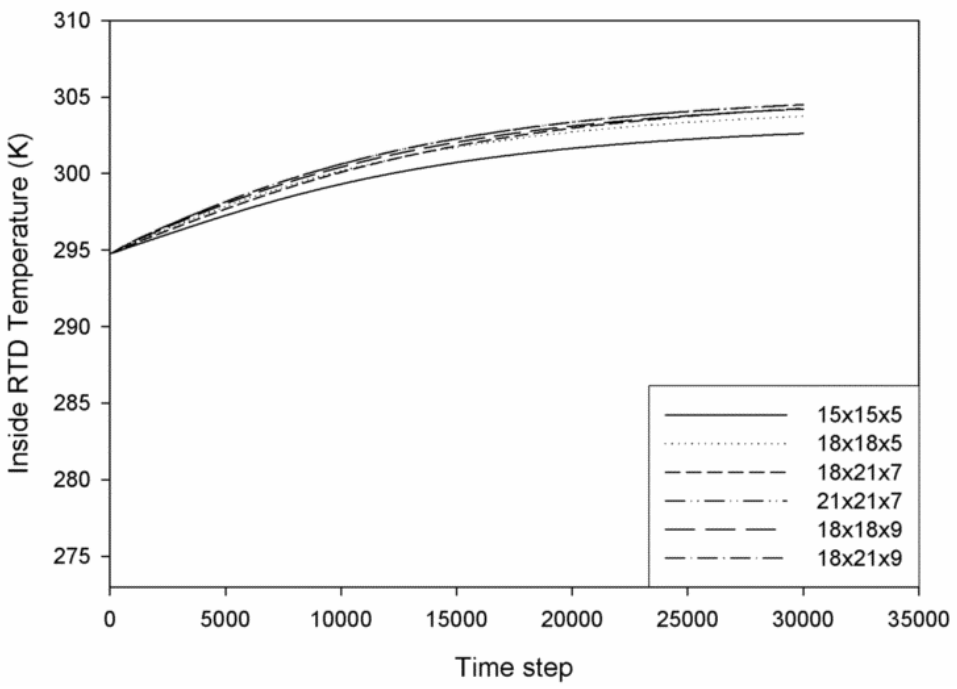
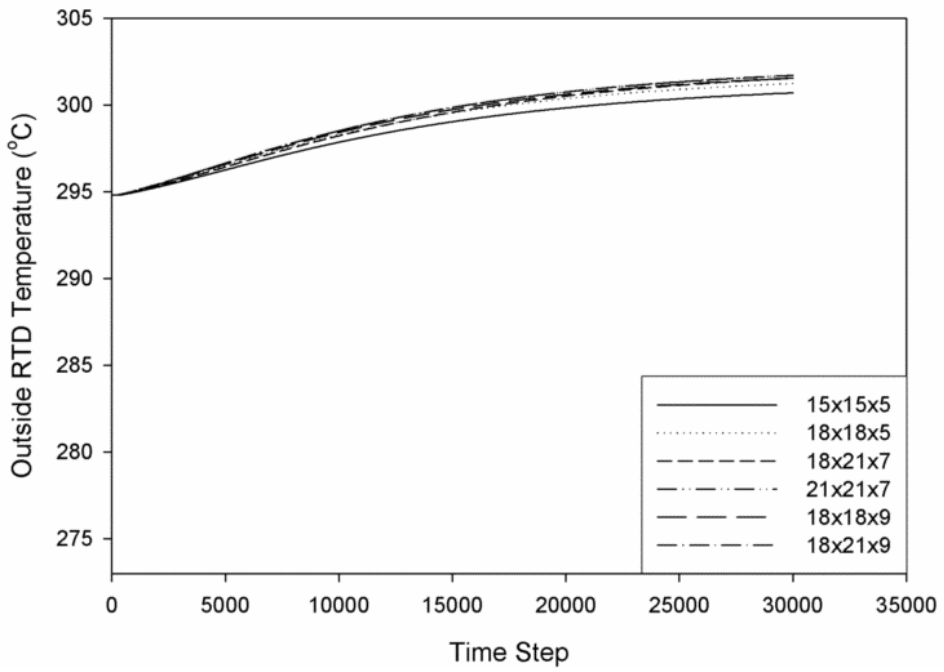


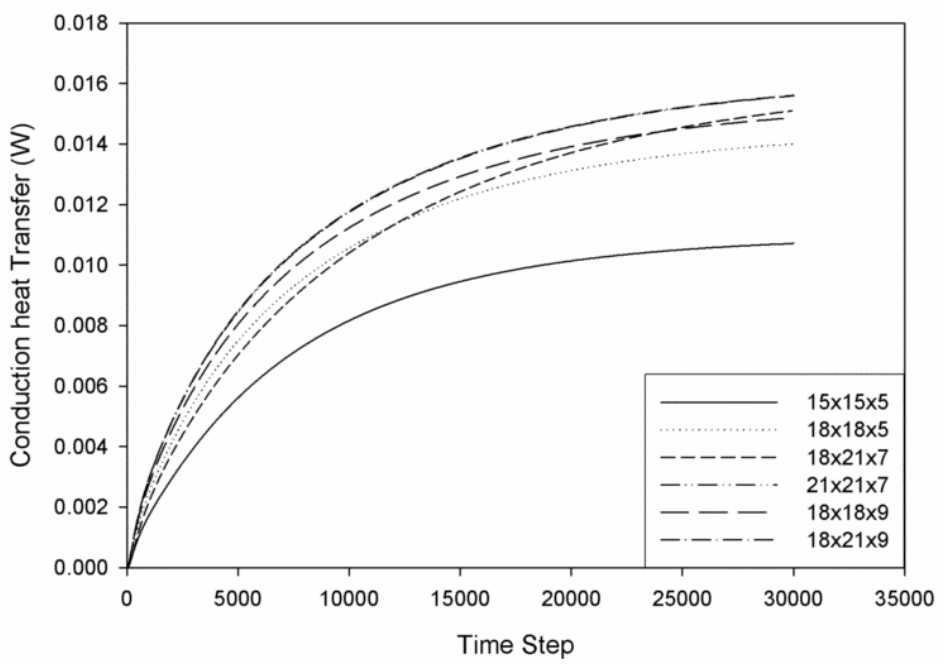
Figure 4.3 Typical schematic of discretization for 3-D axisymmetric model



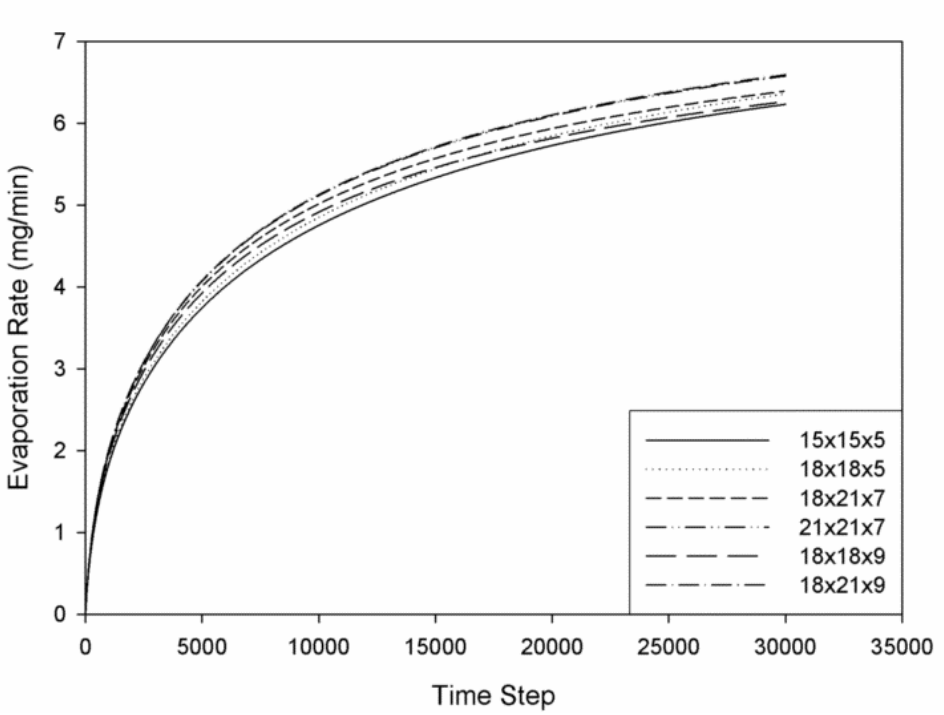
(a) Grid independent study of inside RTD temperature plot



(b) Grid independent study of outside RTD temperature plot



(c) Grid independent study of conduction heat transfer across the membrane



(d) Grid independent study of evaporation rate

Figure 4.4 Grid independent studies of different combinations of number of elements

A typical solutions for conduction heat transfer rates and evaporation rate obtained for the several combinations of number of elements are shown in Figure 4.4 (c) and (d).

The computational results for the grid independent study shows that the numerical calculation becomes grid independent at 21 x 21 x 7 elements in r, z, and Θ directions respectively. Numerical calculation with 21 x 21x 7 elements produced results that are almost identical to the results produced by 18 x 21 x9 elements. The difference between the results produced with 21 x 21 x 7 elements and 18 x21 x9 elements is only 0.06%.

4.4 EQUATION DERIVATION FOR HEAT TRANSFER

The implicit finite difference time domain (FDTD) method is used to calculate heat transfer by conduction and the temperature at each node. The implicit finite difference time domain method is formulated by starting from the governing partial differential equation of the heat diffusion equation. The heat diffusion equation is applied at each node with finite differences employed to approximate differentials. A matrix is established from the resulting set of nodal equation. The following sections show details of how the derivation of the implicit finite time difference time domain equations is carried in the 3-D axisymmetric control volumes.

Finite difference equations to solve for conduction heat transfer and the temperature field in the micro-channel evaporator are find by applying energy conservation to the finite element. First, an energy balance equation is imposed on each element and then parallel resistance network is defined. Finally, a nodal temperature is solved from energy balance and parallel resistance.

4.4.1 ENERGY BALANCE AND PARALLEL RESISTANCE NETWORK

The heat diffusion equation is applied to the element

$$\nabla^2 T = \frac{1}{\alpha}\frac{dT}{dt} \qquad (4.3)$$

where T is temperature, α is thermal diffusivity, and t is time.

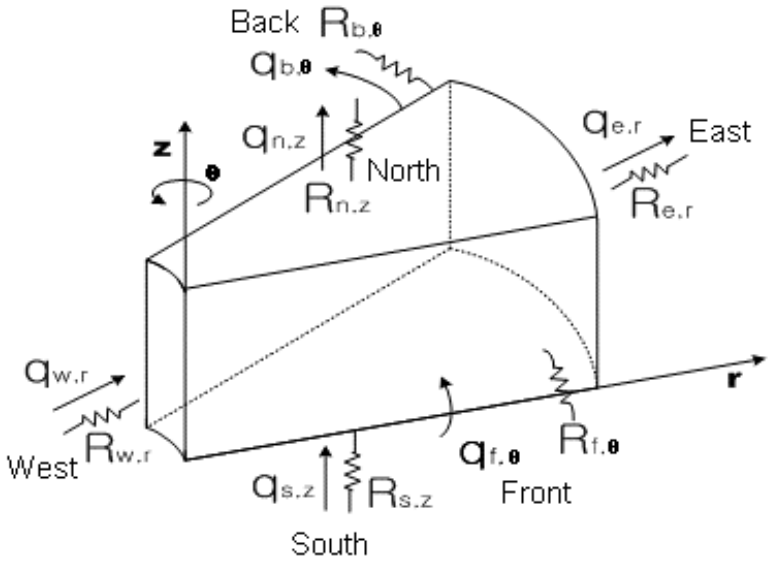


Figure 4.5 An energy balance for a 3-D axisymmetric element

An energy balance for one element is shown in Figure 4.5. A network of parallel resistances is used to approximate conduction heat transfer across the north, south, east, west, front and back faces of element.

The energy balance for each element is

$$E_{in} - E_{out} = E_{stored} \tag{4.4}$$

where

$$E_{in} = q_{w,r} + q_{s,z} + q_{f,\theta} \quad (4.4a) \qquad E_{out} = q_{e,r} + q_{n,z} + q_{b,\theta} \quad (4.4b)$$

$$E_{stored} = q_{st} = (q_{w,r} + q_{s,z} + q_{f,\theta}) - (q_{e,r} + q_{n,z} + q_{b,\theta}) \tag{4.4c}$$

Here the q terms are the heat transfer out of each face of the element as illustrated in Figure 4.5, and q_{st} is the stored energy. Heat conduction is found using Fourier's Law:

$$q = -kA\frac{dT}{dx} \tag{4.5}$$

where k is thermal conductivity, A is across sectional area, T is temperature

Defining the thermal resistances

$$R_{w,r} = \frac{\Delta r}{-k(r_{i,j,k} - \frac{\Delta r}{2})\Delta\theta\Delta z} \tag{4.6a}$$

$$R_{s,z} = \frac{\Delta z}{-k(r_{i,j,k})\Delta\theta\Delta r} \tag{4.6b}$$

$$R_{f,\theta} = \frac{(r_{i,j,k})\Delta\theta}{-k\Delta r\Delta z} \tag{4.6c}$$

$$R_{e,r} = \frac{\Delta r}{-k(r_{i,j,k} + \frac{\Delta r}{2})\Delta\theta\Delta z} \tag{4.6d}$$

$$R_{n,z} = \frac{\Delta z}{-k(r_{i,j,k})\Delta\theta\Delta r} \tag{4.6e}$$

$$R_{b,\theta} = \frac{(r_{i,j,k})\Delta\theta}{-k\Delta r\Delta z} \tag{4.6f}$$

Then Equation 4.5 is become Equation 4.5 (a-f) for each face of element

$$q_{w,r} = -\frac{1}{R_{w,r}}(T^p_{i,j,k} - T^p_{i-1,j,k}) \tag{4.5a}$$

$$q_{s,z} = -\frac{1}{R_{s,z}}(T^p_{i,j,k} - T^p_{i,j,k-1}) \tag{4.5b}$$

$$q_{f,\theta} = -\frac{1}{R_{f,\theta}}(T^p_{i,j,k} - T^p_{i,j-1,k}) \tag{4.5c}$$

$$q_{e,r} = -\frac{1}{R_{e,r}}(T^p_{i+1,j,k} - T^p_{i,j,k}) \tag{4.5d}$$

$$q_{n,z} = -\frac{1}{R_{n,z}}(T^p_{i,j,k+1} - T^p_{i,j,k}) \tag{4.5e}$$

$$q_{b,\theta} = -\frac{1}{R_{w,r}}(T^p_{i,j+1,k} - T^p_{i,j,k}) \tag{4.5f}$$

Equations 4.5 (a~f) are substitute into Equation 4.4c. Energy stored is calculated from Equation 4.4c. Again approximating differentials with finite differences gives:

$$q_{st} = \rho C_p(r_{i,j,k})\Delta\theta\Delta r\Delta z \frac{(T_{i,j,k}^{p} - T_{i,j,k}^{p+1})}{\Delta t} \qquad (4.7)$$

Here subscripts i, j, k are grid point for r, Θ, z direction, R refers an equivalent thermal resistance, ρ is density, C_p is heat capacity, Δr, ΔΘ, Δz are the spatial step size in the each direction as shown in Figure 4.5, Δt is the temporal step, and p+1 is the time step being solved for.

The previews derivation composed of a single element. Section 4.4.1 only represents the energy balance and parallel resistance network for an element. However, the different conditions of elements are presented in the numerical model (Appendix D) and these conditions are the nodes at corners (elements located at corner of control volume), faces with no material interface (elements located at face of control volume with no interface), faces with material interface (elements located at face of control volume with interface), interiors with no material interface (elements located at inside of control volume with no interface), and interiors with material interface (elements located at inside of control volume with interface). Governing equations are different for each type of elements but the solving procedure is identical to section 4.4.1 and following section 4.4.2.

4.4.2 NODAL TEMPERATURE EQUATION

The temperature at each node solved for each time step using the energy balance and parallel resistance network just derived. Again, case for elements composed of a single element is presented in this section. First, equations 4.6 are substituted into the corresponding equation of equations 4.5 to complete the conduction heat transfer equation at each face. The complete equations of equations 4.5 and equation 4.7 are substituted into the Equation 4.4c to complete the energy balance equation and to solve temperatures at the next time step, $T_{i,j,k}^{p+1}$. To simplify the energy balance equation, following terms are defined:

$$A = \frac{\rho C_p(r_{i,j,k})\Delta\theta\Delta r\Delta z}{\Delta t} \quad (4.8a) \qquad B = \frac{k(r_{i,j,k} - \frac{\Delta r}{2})\Delta\theta\Delta z}{\Delta r} \quad (4.8.b)$$

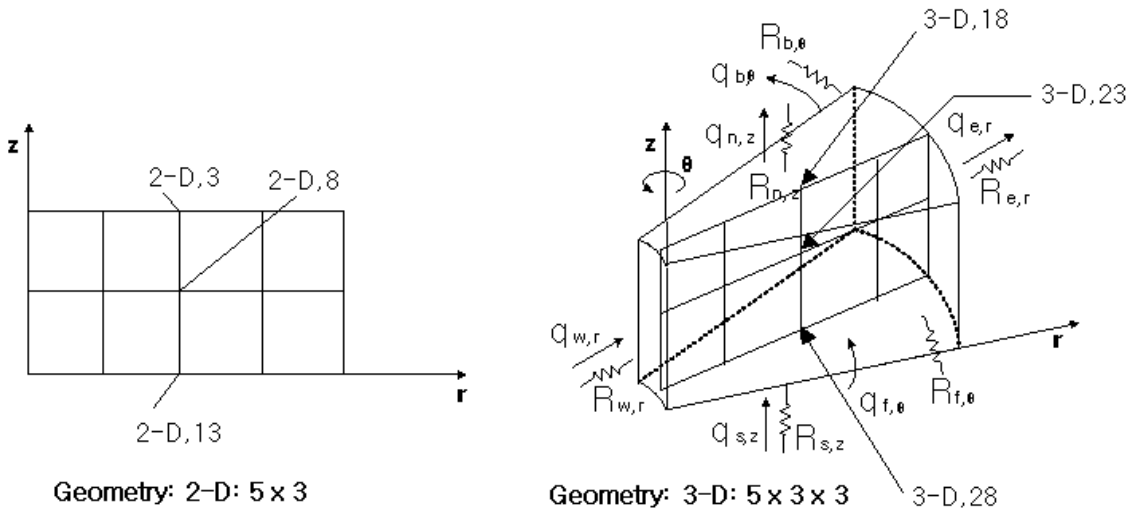
$$C = \frac{k(r_{i,j,k} + \frac{\Delta r}{2})\Delta\theta\Delta z}{\Delta r} \quad (4.8.c) \qquad D = \frac{k(r_{i,j,k})\Delta\theta\Delta r}{\Delta z} \quad (4.8.d)$$

$$E = \frac{k(r_{i,j,k})\Delta\theta\Delta r}{\Delta z} \quad (4.8.e) \qquad F = \frac{k\Delta r\Delta z}{(r_{i,j,k})\Delta\theta} \quad (4.8.f) \qquad G = \frac{k\Delta r\Delta z}{(r_{i,j,k})\Delta\theta} \quad (4.8.g)$$

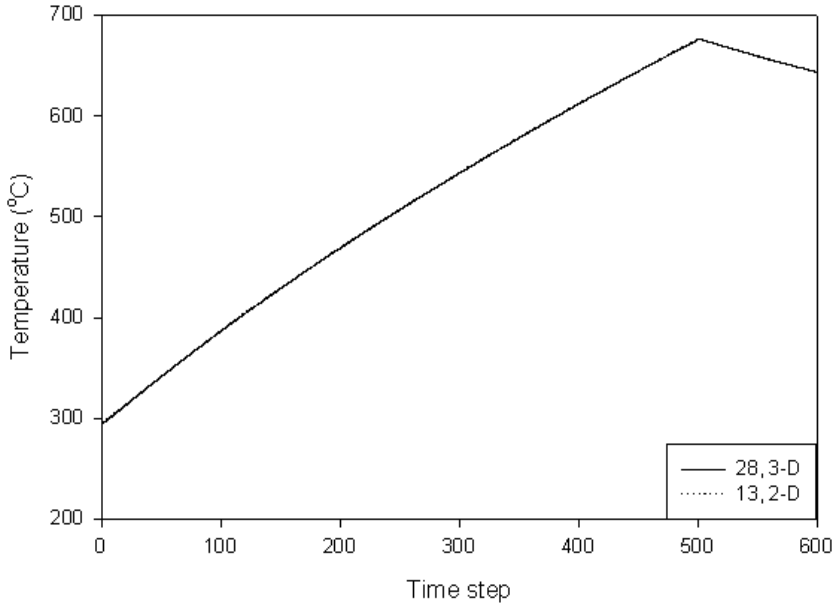
the nodal temperature equation simplifies to:

$$T_{i,j,k}^{p} = \frac{1}{A}[(A + B + C + D + E + F + G)T_{i,j,k}^{p}$$

$$- BT_{i-1,j,k}^{p} - CT_{i+1,j,k}^{p} - DT_{i,j,k-1}^{p} - ET_{i,j,k+1}^{p} - FT_{i,j-1,k}^{p} - GT_{i,j+1,k}^{p}] \quad (4.9)$$



(a) Schematic of 2-D & 3-D



(b) Temperature comparison

Figure 4.6 Schematic of corresponding points for 2-D and 3-D: temperature plot

The nodal temperature equations for elements composed of more than one material are different for each type of elements. These equations are given in appendix B.

The 3-D numerical calculation results are compared with the 2-D numerical model previously work developed at Washington State University [9]. Results for a single uniform silicon material and coarse mesh are calculated using in both of the models and are shown in Figure 4.6 (a) and (b). The 2-D model uses a mesh size of 5 x 3 and the 3-D model uses a mesh size of 5 x 3 x 3 to compare. Thus there are 13 nodes for the 2-D model and 28 nodes for the 3-D model as shown in Figure 4.6 (a) and (b). The comparison shows identical results for both models for the silicon material.

4.5 MASS TRANSFER

A mass transfer model is used account for heat transfer from the liquid-vapor interfaces due to evaporation. The mass transfer model is based on assumption that mass transfer occurs at the liquid-vapor interface, and that phase change at the periphery of the silicon membrane is negligible. Since this area is relatively small compared to the total surface area of the liquid on the membrane it is assumed to have effect on mass transfer rates.

A detailed flow analysis using the force balance in the channel and Yong-Laplace equation [72] are presented in section 4.6. The net vapor flux or mass transfer rate associated with evaporation from the liquid-vapor interface is determined from kinetic theory [71]: To calculate the evaporation rate j, conservation of energy is used at the liquid-vapor interface. The amount of heat transferred through the liquid layer is equal to the heat carried away from the liquid layer by phase change. The energy balance at the interface becomes:

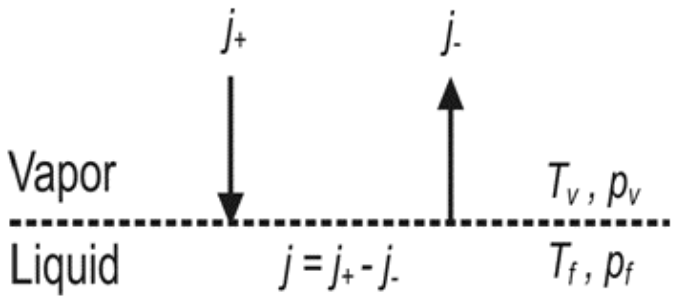


Figure 4.7 Mass flux balance at liquid-vapor interface, Courtesy Scott Whalen

$$q_v - q_l = 0 \tag{4.10}$$

The latent heat of vaporization carried away by evaporation is:

$$q_v = jAh_{fg} \tag{4.11}$$

The sensible heat conduction through the liquid layer is:

$$q_l = -\frac{k_l A}{L}(T_{sl} - T_l) \tag{4.12}$$

Figure 4.7 shows the net mass flux j at the liquid-vapor interface. The net mass flux is the difference between the inlet vapor flux, j_+, from condensation and the outlet vapor flux, j-, from the evaporation.

$$j = \sqrt{\frac{M}{2\pi R}}\left[\frac{p_v}{\sqrt{T_v}} - \frac{p_l}{\sqrt{T_l}}\right] \tag{4.13}$$

where j is the mass flux, M was the molecular weight of the working fluid (0.416 kg/mol for FC77), R is universal gas constant (8.314 J/kmol·K), h_{fg} is the latent heat of vaporization, P and T are the vapor pressure and temperature subscripts v and l referring to the vapor and liquid surface of a homogenous material.

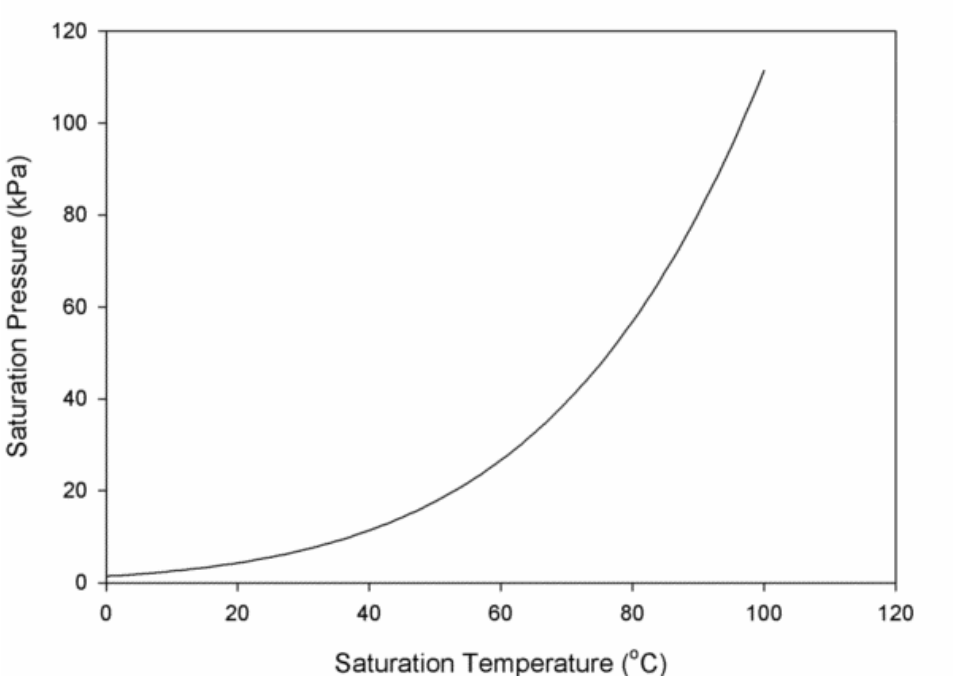


Figure 4.8 Saturation curve for Fluorinert FC77

Equations 4.10 to 4.13 are combined to form heat balance equation:

$$\sqrt{\frac{M}{2\pi R}}\left[\frac{p_v}{\sqrt{T_v}}-\frac{p_l}{\sqrt{T_l}}\right]h_{fg}+\frac{k_l}{L}(T_{sl}-T_l)=0 \qquad (4.14)$$

The only unknown in the equation 4.14 is the vapor temperature T_v. The value of T_v which satisfies Equation 4.14 is determined by iteration. Hence the secant method is used with a convergence criterion of 0.00001°C. Details of the secant method are presented in the appendix D with the complete FORTRAN CODE.

4.6 MENISCUS APPLICATION

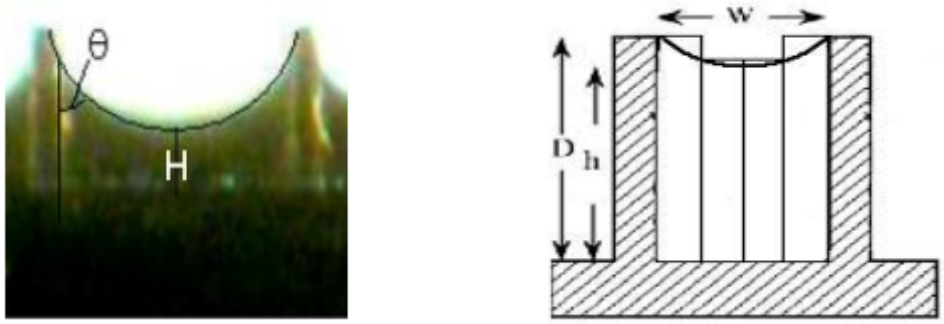


Figure 4.9 Discretization of meniscus in the micro-channel for FDTD model

The meniscus is formed in the micro-channel as shown in Figure 4.9. The meniscus is a curve in the surface of a liquid and is produced in response to the surface of the channel. The meniscus is formed as a circular shape in general and can be described mathematically as shown in Appendix A.

To have more accurate numerical calculation results, the numerical model should consider the meniscus shape. Figure 4.9 shows that the how the meniscus shape is accounted in the numerical model. The coarse mesh is used in the calculation to save computational time.

The liquid thickness at center of channel, H, is found:

$$H = D - \frac{w}{2}\tan\theta \qquad (4.15)$$

where H is working fluid thickness from center of channel, D is height of wick structures, w is width of channel, and θ is contact angle found from visualization test.

The liquid capillary pressure in the channel is found:

$$P_c = \frac{2\gamma \cos\theta}{w} \qquad (4.16)$$

where P_c is capillary pressure, γ is surface tension ($8\text{x}10^{-3}$N/m), θ is contact angle (41^o), and w is micro-channel width.

4.7 FLOW ANALYSYS

To determine the liquid thickness, the force balance equation is derived for the velocity of a liquid flowing in an open micro-channel.

Consider the flow through the micro-channel shown in Figure 4.10. In Figure 4.10, the force balance for steady state from in the x-direction is

$$\sum F_x = Pdydz - Pdydz - dPdydz - 2\tau dxdz - \tau dxdy = 0 \qquad (4.17)$$

where P is pressure, x is the distance along with the channel, y is distance of width of channel, z is the distance of height of the wall.

Simplifying:

$$-dPdydz - 2\tau dxdz - \tau dxdy = 0 \qquad (4.18)$$

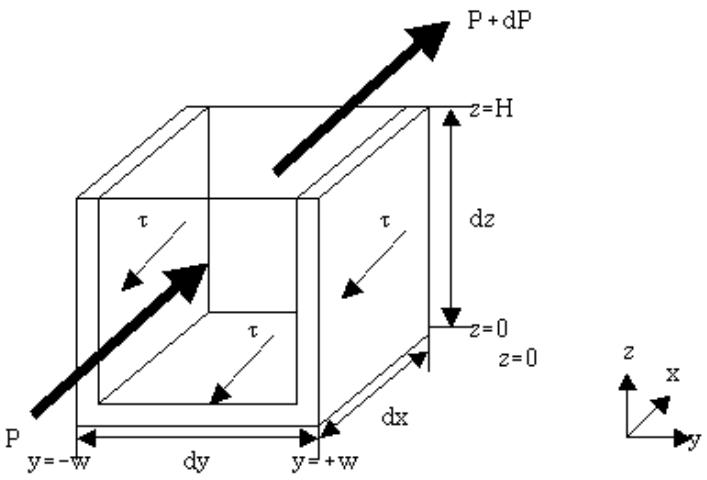


Figure 4.10 Schematic of Three dimensional flows

Divide both sides by dxdydz

$$-\frac{dp}{dx} = 2\frac{d\tau}{dy} + \frac{d\tau}{dz} \tag{4.19}$$

Apply the definition of shear force: $\tau = \mu \frac{du}{dy}$, then equation 4.19 become:

$$-\frac{1}{\mu}\frac{dp}{dx} = 2\frac{d^2u}{dy^2} + \frac{d^2u}{dz^2} \tag{4.20}$$

The boundary conditions are the no slip condition applied at the walls and the assumptions of zero shear stress at the liquid-vapor interface:

3D B.C: (a) $u=0$ where $y=-w$ (b) $u=0$ where $y=+w$

(c) $u=0$ where $z=0$ (d) $du/dz=0$ where $z=H$

We make the reasonable assumption that the liquid velocity found in the 3D model is lower than the liquid velocity in the 2D model. This assumption can be made more reasonable by remembering that the 2D model is represented as two parallel plates with shear forces on both of the sidewalls, and no shear force on the bottom or top.

The velocity of 2D flow is found in Appendix A (u=$-\frac{1}{4\mu}\frac{dP}{dx}(y^2-w^2)$), the velocity of 3D flow can then be found by subtracting an arbitrary velocity from the velocity of the 2D flow. This arbitrary velocity is a function of *y* and *z*.

Therefore, the velocity of a 3D flow can be written as:

$$u(y,z) = u(2D\ velocity) + u'(y,z) \tag{4.21}$$

Equation 4.21 can be rewritten using the velocity of the 2-D flow (u=$-\frac{1}{4\mu}\frac{dP}{dx}(y^2-w^2)$)

$$u(y,z) = -\frac{1}{4\mu}\frac{dP}{dx}(y^2-w^2) + u'(y,z) \tag{4.22}$$

The first and second derivatives of u with respect to y are:

$$\frac{du(y,z)}{dy} = -\frac{1}{2\mu}\frac{dP}{dx}y + \frac{du'(y,z)}{dy} \qquad \frac{d^2u(y,z)}{dy^2} = -\frac{1}{2\mu}\frac{dP}{dx} + \frac{d^2u'(y,z)}{dy^2}$$

(4.23)

The first and second derivatives with respect to z are:

$$\frac{du}{dz} = \frac{du'(y,z)}{dz} \qquad \frac{d^2u}{dz^2} = \frac{d^2u'(y,z)}{dz^2} \tag{4.24}$$

Substituting the second derivatives in Equation 4.23 and Equation 4.24 in to equation 4.20 gives:

$$-\frac{1}{\mu}\frac{dp}{dx} = 2\left(-\frac{1}{2\mu}\frac{dP}{dx} + \frac{d^2u'(y,z)}{dy^2}\right) + \frac{d^2u'(y,z)}{dz^2} \tag{4.25}$$

Thus 4.25 become:

$$2\frac{d^2u'(y,z)}{dy^2} + \frac{d^2u'(y,z)}{dz^2} = 0 \tag{4.26}$$

3-D boundary conditions (no slip condition applied at the walls, the assumptions of zero shear stress at the liquid-vapor interface) are applied to Equation 4.22 to find new boundary conditions for the arbitrary velocity, $u'(y,z)$. The new boundary conditions are:

New 3D B.C: (a) $u'(y,z) = 0$, where y=-w (b) $u'(y,z) = 0$, where y=+w

(c) $\frac{du'(y,z)}{dz} = 0$, where z=H (d) $u'(y,z) = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)$, where z=*0*

The variable $u'(y,z)$ can be founded using separation of variables.

$$u'(y,z) = F(y)G(z) \tag{4.27}$$

Substituting Equation 4.26 to the Equation 4.27 gives

$$2\frac{d^2}{dy^2}F(y)G(z)+\frac{d^2}{dz^2}F(y)G(z)=0 \tag{4.28}$$

Separating variables gives:

$$\frac{2}{F(y)}\frac{d^2F(y)}{dy^2}=-\frac{1}{G(z)}\frac{d^2G(z)}{dz^2}=-\lambda^2 \tag{4.29}$$

From Equation 4.29, first solution for y is

$$\frac{d^2F(y)}{dy^2}+\frac{\lambda^2}{2}F(y)=0 \tag{4.30}$$

and the new 3D boundary conditions (a) $u'(y,z)=0$, where y=-w and (b) $u'(y,z)=0$, where y=+w imply

$$F(\pm w)=0 \tag{4.31}$$

Equation 4.29 gives eigenvalue $\lambda^2=(\frac{n\pi}{2w})^2$ and corresponding nonzero solutions

$$F(y)=\sin(\frac{n\pi}{2w}y) \qquad n=1,2,\cdots \tag{4.32}$$

The ODE for G(z) with eigenvalue $\lambda^2=(\frac{n\pi}{2w})^2$ then becomes

$$\frac{d^2G(y)}{dz^2} - \lambda^2 G(z) = 0 \qquad (4.33)$$

The general solution of equation 4.33 is

$$G(z) = A\cosh\ \lambda z + B\sinh\ \lambda z = A\cosh(\frac{n\pi}{2w}z) + B\sinh(\frac{n\pi}{2w}z)$$

$$n = 1,2,\cdots \qquad (4.34)$$

The new 3D boundary condition (d) $u'(y,z) = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)$, where $z=0$

implies that $G(0) = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)$; that is, $G(0) = A = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)$. This gives

$$G(z) = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)\cosh(\frac{n\pi z}{2w}) + B\sinh(\frac{n\pi z}{2w})$$

$$n = 1,2,\cdots \qquad (4.35)$$

Substituting Equations 4.32 and 4.35 into 4.27 ($u'(y,z) = F(y)G(z)$), gives

$$u'(y,z) = F(y)G(z) = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)\cosh(\frac{n\pi z}{2w})\sin(\frac{n\pi y}{2w}) + B\sinh(\frac{n\pi z}{2w})\sin(\frac{n\pi y}{2w})$$

$$n = 1,2,\cdots \qquad (4.36)$$

The new 3D boundary condition (c) $\frac{du'(y,z)}{dz} = 0$, where $z=H$ implies that $\frac{du'(y,H)}{dz} = 0$,

and this gives

$$\frac{du'(y,H)}{dz} = 0 = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)(\frac{n\pi}{2w}\sinh\frac{n\pi H}{2w}) + B(\frac{n\pi}{2w}\cosh\frac{n\pi H}{2w})$$

$$n = 1,2,\cdots \tag{4.37}$$

From Equation 4.37, B is found to be

$$B = -\frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)\tanh(\frac{n\pi H}{2w}) \qquad n = 1,2,\cdots \tag{4.38}$$

Substituting Equation 4.38 into Equation 4.36, the final form of $u'(y,z)$ is

$$u'(y,z) = F(y)G(z) = \frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)(\cosh(\frac{n\pi z}{2w}) - \tanh(\frac{n\pi H}{2w})\sinh(\frac{n\pi z}{2w}))\sin(\frac{n\pi y}{2w})$$

$$n = 1,2,\cdots \tag{4.39}$$

Substituting Equation 4.39 into Equation 4.22 ($u(y,z) = -\frac{1}{4\mu}\frac{dP}{dx}(y^2 - w^2) + u'(y,z)$), now we have the velocity of 3D flow equation as follows.

$$u(y,z) = -\frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2)(1 - \sum_{n=1}^{\infty}(\cosh(\frac{n\pi z}{2w}) - \tanh(\frac{n\pi H}{2w})\sinh(\frac{n\pi z}{2w}))\sin(\frac{n\pi y}{2w}))$$

$$n = 1,2,\cdots \tag{4.40}$$

By simplifying Equation 4.40 with trigonometric definitions ($e^{x} = \cosh x + \sinh x$, $e^{-x} = \cosh x - \sinh x$) and identities, the final form of the axial speed of flow in the micro-channel is

$$u(y,z) = -\frac{1}{4\mu}\frac{dp}{dx}(y^2 - w^2) \tag{4.41}$$

From the axial speed of flow (Equation 4.41), mass flow rate in the micro-channel can be calculated by multiply the area of flow moving in and density of fluid to axial speed. Then we can calculate how much mass is added to each element for each time step. Again, liquid thickness can be calculated by dividing this mass with density and bottom area of element. The pressure of each element is already determined with Yong-Laplace equation and the constant contact angle (41°), $p = \frac{2\gamma\cos\theta}{w}$, where γ is surface tension of fluid (FC77=8x10^{-3}N/m), θ is contact angle (41°), and w is width of channel. With this pressure *dp* is being calculated.

4.8 BOUNDARY CONDITIONS

The pressure, temperature, and vapor volume of each element cell are determined individually based on the net mass of vapor generated in each cell obtained by solving the ideal gas law as explained in section 4.5 and the boundary conditions.

The boundary conditions applied to the numerical solution determined in this work are described below and shown in Figure 4.11. Five boundary conditions are applied. Each boundary condition imposed is described in turn. (1) The west face, which represents the center of the die, is insulated due to the axisymmetric geometric approximation. (2) The

south face of concentric heater (*D=1.6mm*), which represents the inner portion at the bottom of the silicon membrane, is subjected to a constant heat flux. (3) The east face of the silicon membrane, which represents the periphery of the silicon die, conducts heat to the bulk. (4) The north face, which represents top of the membrane, has a convective boundary condition. (5) The front and back faces are insulated due to the axisymmetric geometry. These boundary conditions are applied to individual nodes or over an entire surface when executing the numerical code.

Two simulations are run to examine the effect of convective boundary conditions and show a difference of only 0.5%. The free (h=100w/mK) and forced gas convective (h=300w/mK) coefficients are found [73]. This h value represents the convective heat transfer on the north face. A convective coefficient less than 300w/mK is actually expected since the north face is covered by an environment air. The average convective (h=250w/mK) boundary conditions can be used since it shows only small difference and the evaporator is covered by air. The east face boundary condition is determined the in previous 2-D numerical work [9].

The east face is bounded by the gasket, and this surface is assumed to be insulated except where the membranes join the bulk silicon die. Newton's law of cooling is used to approximate heat transfer from the membrane to the bulk silicon to save computation time. The effective convective coefficient h_{eff} = 135,000 W/m^2K was determined by Whalen [9]. Whalen fit his 2-D numerical model to experimental data obtained for a device assembled with air as the working fluid. Heat was applied at the indicated location using a single ring resistance heater pulsed for 5ms and delivering 9.2mJ. The result Taken from [9] is shown in Figure 4.12.

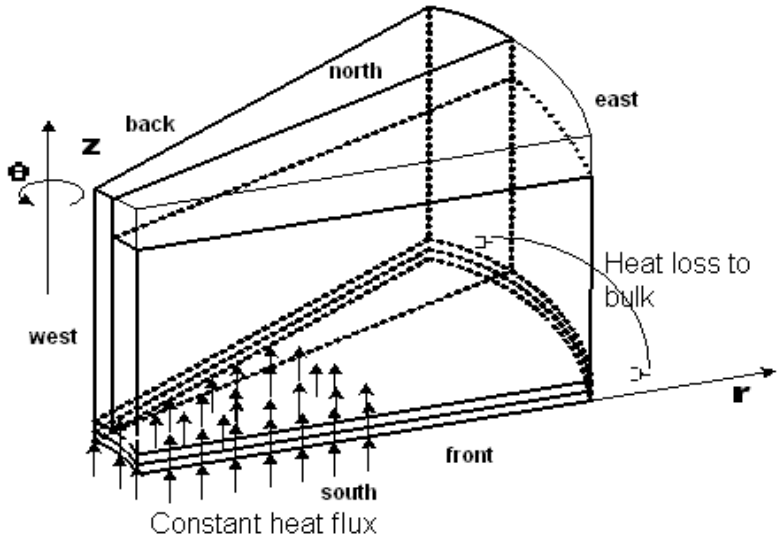


Figure 4.11 Thermal boundary conditions for 3-D model

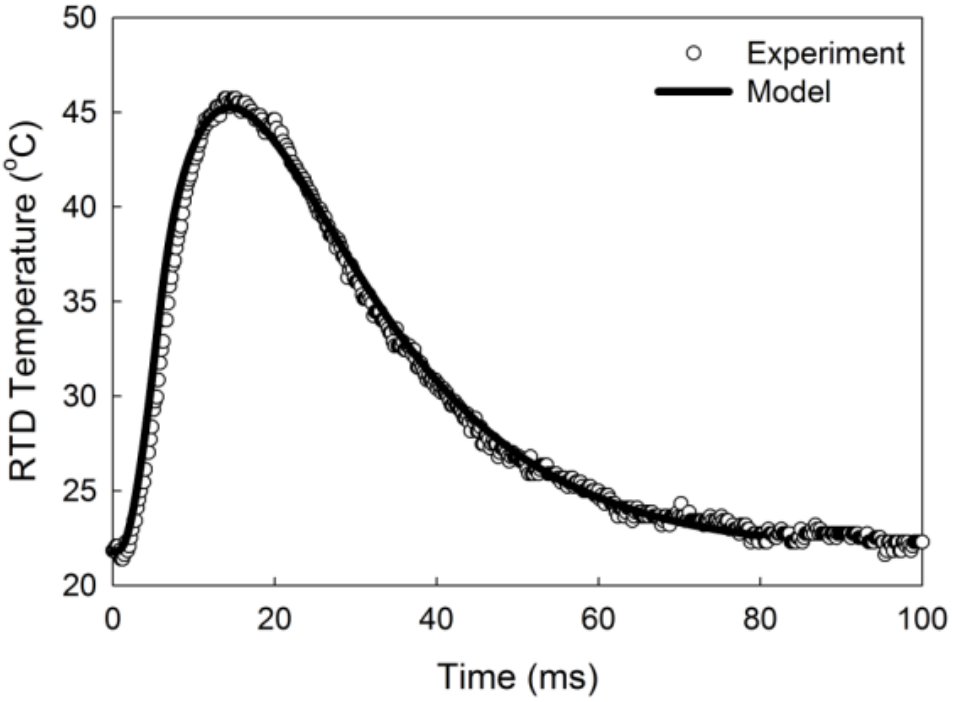


Figure 4.12 Fitting of model to the experiment for determining $h_{eff,}$

Courtesy Scott Whalen

The 3-D numerical model was run employing this effective convective coefficient, h_{eff}. The result was good agreement with 2-D model results. However, this h_{eff} boundary condition was found to be valid only for evaporators with silicon membranes. Numerical simulations run for evaporators with SiNx membranes gave poor results.

In order the numerically simulate SiNx membranes, a new boundary condition for the east face was required. Since the membrane thickness and the bulk thickness are significantly different, the temperature at the edge of square membrane was assumed to be uniform. For this reason a constant surface temperature boundary condition was employed [54]. To verify the assumption of constant temperature at the edge of the membrane triple RTDs, as shown in Figure 2.4, were fabricated to measure the membrane edge temperature. The 3-D numerical results with the fixed temperature boundary conditions were the compared to measured results from the experiment. Good agreement between numerically simulated and experimentally measure results were found. A detailed comparison of the numerical results and experiment results are presented in the Chapter 6.

4.9 FLOWCHART

The flow chart shown in Figure 4.13 illustrates the solution procedure for the numerical work presented in this dissertation. First, the initial conditions of material temperatures and thermodynamic properties are set before computation. The numerical model is discretized to define mesh size and grid spacing. Next the material properties and boundary conditions are defined for each individual element.

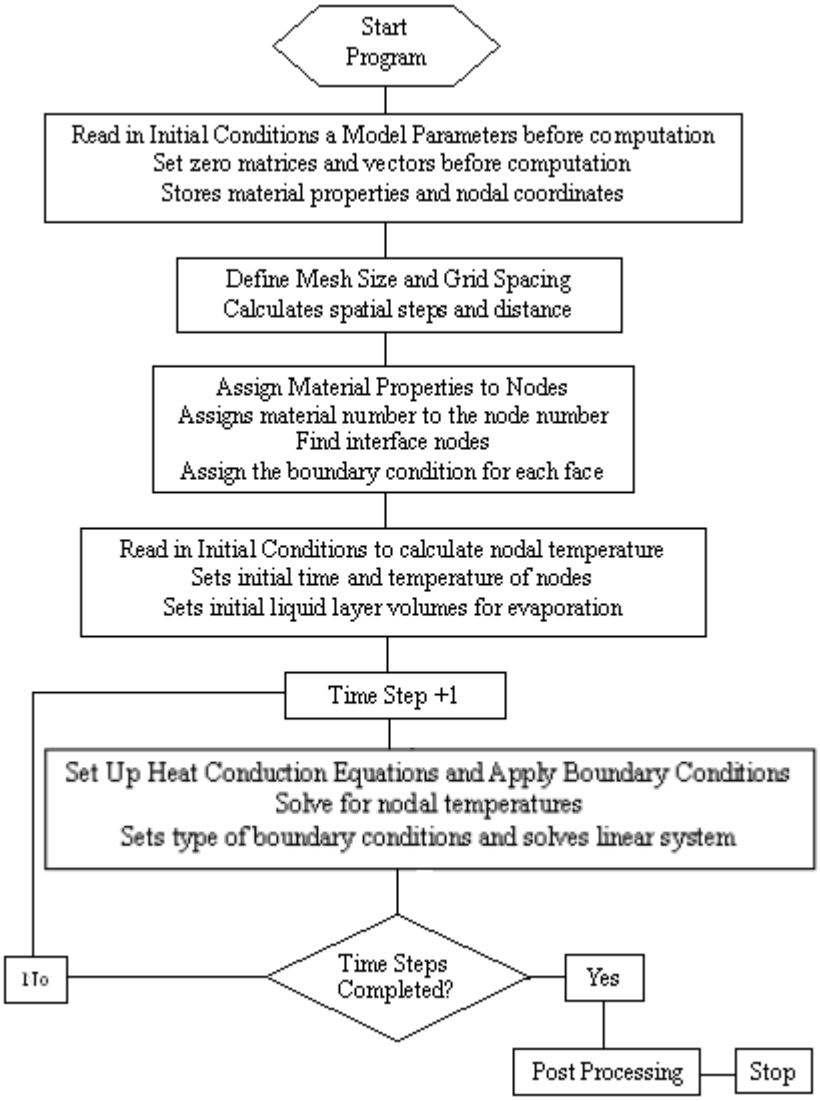


Figure 4.13 3-D numerical model flowchart

The first time step of temperatures are calculated with the implicit finite time domain method and the nodal temperature equation as explained in section 4.4.

The evaporation mass fluxes for each element are then calculated as explained in section 4.5. The mass flow rate of liquid working fluid is then found for the tapered micro-channels case. The program iterates returning to the temperature calculations using the FDTD method, with nodal temperature equation. Evaporation mass fluxes and liquid

mass flow rates are found. Iteration repeats until the specified solution time (Input time to the program) is satisfied.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 EXPERIMENTAL RESULTS OVERVIEW

The results of heat and mass transfer measurements performed on micro-channel evaporators are presented in this chapter. Steady-state evaporation and transient evaporation tests used to characterize the performance of different micro-channel evaporators are presented.

Two geometries of micro-channel evaporators are considered. Constant cross section channels are considered first. Dimensions of constant rectangular cross section micro-channels are described as channel depth x channel width x channel wall thickness. For example 10 x 10 x 10 refers to a micro-channel that is 10μm deep, 10μm wide with each channel formed between 10μm thick walls. Tapered micro-channels are considered next. Dimensions of tapered micro-channels are 40μm deep and their width is 80μm at the outer radius of the micro-channel. Each micro-channel width tapers down to 5μm at the inner radius of the micro-channel. The walls dividing micro-channels also are 80μm wide at their outer radius and taper down to a width of 5μm at their inner radius. Since the wall thickness and channel width are tapered same dimensions, dimensions of tapered micro-channels can be described as 40 x 5~80.

Finally, evaporators with membranes fabricated of two different materials, silicon and silicon-nitride are tested. A series of steady state evaporation tests and transient operation tests are presented in this chapter and appendix C. The list of the test conditions is shown in Table 5.1.

Material	SU8 Height (μm)	SU8 Wall Thickness (μm)	Channel Width (μm)	Average power input (mW)	Conditions Frequency (Hz) Duty cycle (%)
Si	10	10	10	27.1, 30.3, 34.1, 37.2, 41.3	Steady state
Si	40	5	35	33.6, 48.4, 57.5	Steady state
Si	40	5	50	35.5, 49.9, 70.3	Steady state
Si	40	5	70	32.0, 42.5, 53.9	Steady state
Si	40	5~80	5~80	33.6, 43.7, 57.7	Steady state
SiNx	40	5~80	5~80	13, 10, 7	Steady state

(a) Steady state conditions

Material	SU8 Height (μm)	SU8 Wall Thickness (μm)	Channel Width (μm)	Average power input (mW)	Conditions Frequency (Hz) Duty cycle (%)
Si	40	5~80	5~80	34, 44, 57	10Hz 50%, 30%
Si	40	5~80	5~80	34, 44, 57	20Hz 50%, 30%

(b) Transient operation conditions

Table 5.1 Summary of experiment tests

5.2 STEADY STATE EVAPORATION TEST RESULTS

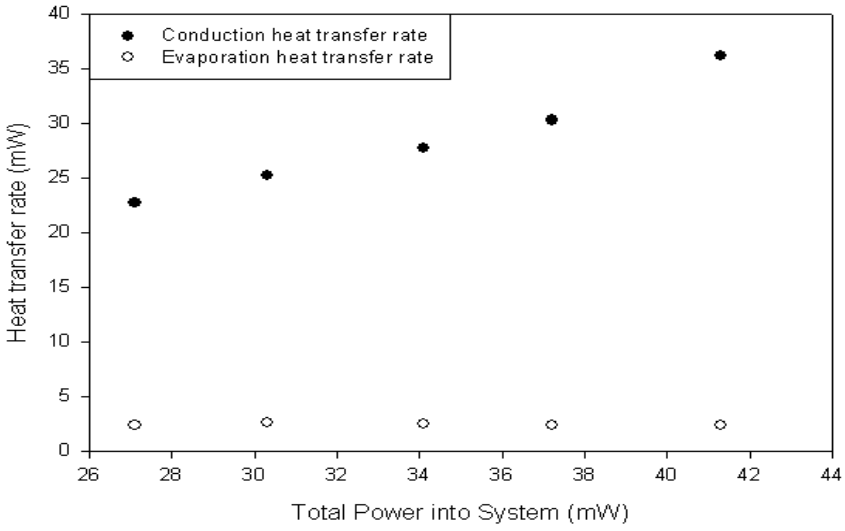


Figure 5.1 Evaporation test results of heat transfer by conduction and evaporation:

10 x 10 x 10 micro-channel evaporator

	Power into Heater	Inside RTD Temp	Outside RTD Temp	Power across Membrane	Energy into Evap	Efficiency
	(mW)	(^{o}C)	(^{o}C)	(mW)	(mW)	(%)
10μm high 10μm wide 10μm thick micro-channels	27.1	36.9	32.7	22.7	2.4	9
	30.3	37.9	33.2	25.2	2.6	8
	34.1	38.8	33.6	27.7	2.5	7
	37.2	38.8	33.6	30.3	2.4	7
	41.3	40.6	34.5	36.2	2.4	6

Table 5.2 Experimental energy balance:

10 x 10 x 10 micro-channel evaporator

A series of steady state evaporation tests are presented. Results from experimental measurements for constant rectangular cross section micro-channels with 10μm depth, 10μm width and 5μm thick channel walls are shown in Figure 5.1 and tabulated in Table 5.2. The membrane material for the evaporator is silicon. Results are given for power inputs of 27mW to 41mW. In Figure 5.1, experimental values of sensible heat transfer by radial conduction (black solid circles) and latent heat transfer by evaporation of working fluid (white circles) are plotted against the electrical power dissipated in the thin-film heater at the center of the evaporator membrane. Results given in Table 5.2 show that over the power range studied, the temperature of inner and outer RTD ranged from 37^{o}C to 41^{o}C and 32^{o}C to 34^{o}C respectively with temperature uncertainty of ±0.5^{o}C. The mass evaporation rate ranged from 28.7μg/sec to 31.7μg/sec with an uncertainty of ±2.5μg/sec. Thus the energy balance indicates that approximately 90% to 95% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while only 5% to 10% of the power is carried away by the latent heat of evaporation.

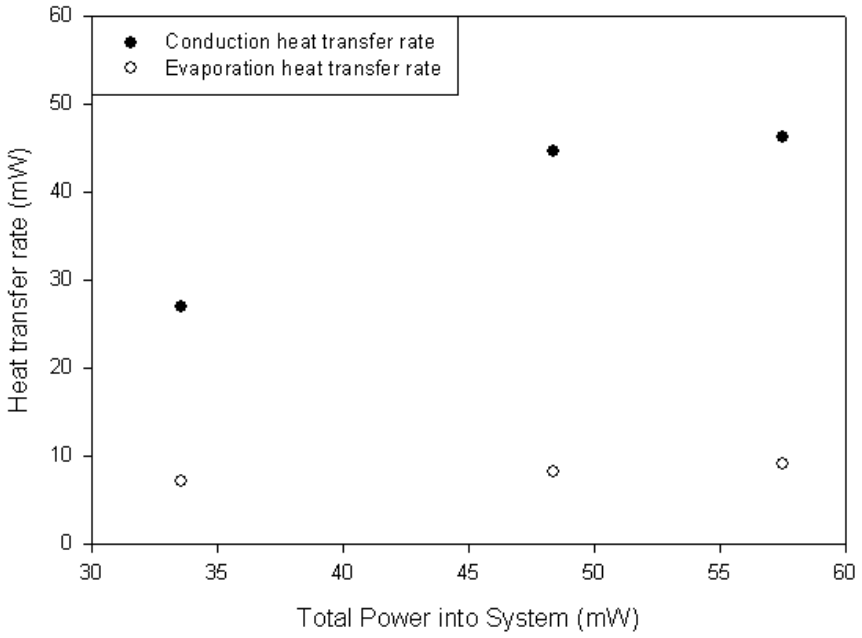


Figure 5.2 Evaporation test results of heat transfer by conduction and evaporation:

40 x 35 x 5 micro-channel evaporator

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	35	33.6	33.1	28.6	27.0	7.1	85	21.0
			48.4	37.5	29.9	44.6	8.2	100	16.3
			57.5	39.0	31.2	46.3	9.1	108	15.7

Table 5.3 Experimental energy balance:

40 x 35 x 5 micro-channel evaporator

Figure 5.2 shows experimental results for a constant rectangular cross section micro-channels with 40μm depth, 35μm width and 5μm thick channel walls with thermal power inputs of 34, 44 and 55mW. Again, the sensible heat transfer by radial conduction is indicated by black solid circles and the latent heat transfer by evaporation of working fluid is indicated by white circles.

On Figure 5.2, the heat transfer rates and evaporation rates are plotted against total power inputs. Results of the evaporation experiments given in Table 5.3 show that over the power range studied, the inner RTD temperature ranged from 33^{o}C to 39^{o}C, while the outer RTD temperature ranged from 29^{o}C to 32^{o}C with temperature uncertainty of ±0.5^{o}C. The evaporation rate ranged from 85μg/sec to 108μg/sec with uncertainty of ±2.5μg/sec. The experimental energy balance indicates that approximately 79% to 84% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while only 16% to 21% of the power is carried away by the latent heat of evaporation.

Conduction heat transfer rates (black solid circles) and evaporation rates (white circles) for 40 x 50 x 5 micro-channel evaporators are plotted in Figure 5.3. The micro-channel evaporator membrane is silicon. Both Figure 5.3 and Table 5.4 show experimental results for thermal power inputs of 35.5, 49.9, and 70.3mW. For this power range the inner RTD temperature ranged from 31^{o}C to 42^{o}C while the outer RTD temperature ranged from 27^{o}C to 31^{o}C with ±0.5^{o}C. The mass evaporation rates ranged from 95μg/sec to 134μg/sec with ±2.5 μg/sec. The experimental energy balance indicates that approximately 77% to 84% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while only 16% to 23% of the power is carried away by the latent heat of evaporation.

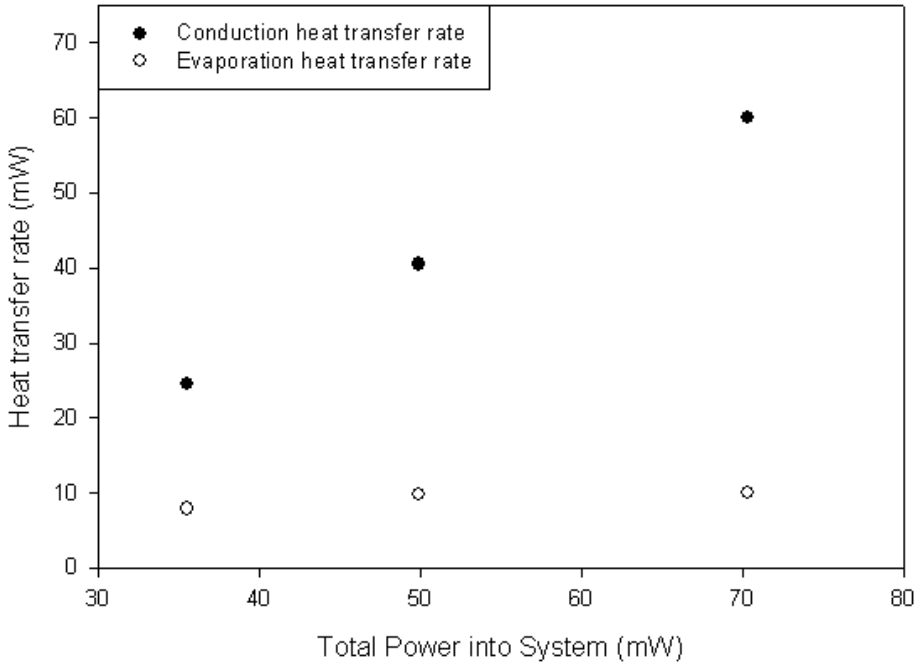


Figure 5.3 Evaporation test results of heat transfer by conduction and evaporation:

40 x 50 x 5 micro-channel evaporator

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	50	35.5	31.5	27.4	24.6	7.9	95.0	22.4
			49.9	36.2	28.9	40.5	9.9	118.3	20.2
			70.3	41.4	30.9	60.1	10.1	133.3	15.8

Table 5.4 Experimental energy balance:

40 x 50 x 5 micro-channel evaporator

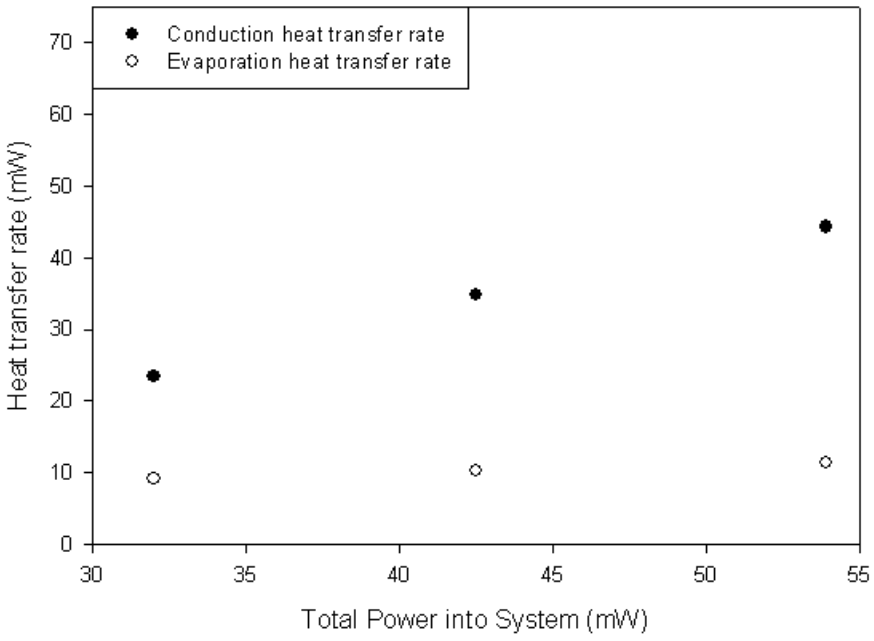


Figure 5.4 Evaporation test results of heat transfer by conduction and evaporation:

40 x 70 x 5 micro-channel evaporator

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	70	32.0	32.1	28.2	23.4	9.1	108.3	28.4
			42.5	35.0	30.0	34.8	10.3	121.7	24.2
			53.9	40.4	33.2	44.3	11.4	136.7	21.2

Table 5.5 Experimental results of energy balance:

40 x 70 x 5 micro-channel evaporator

Results from the evaporation experiments for 40 x 70 x 5 micro-channel evaporator are given in Figure 5.4. Experimental results are given for power inputs of 32mW to 54mW. In Figure 5.4, experimental values of sensible heat transfer by conduction radially out of the membrane are marked with black circles and latent heat transfer by evaporation of working fluid is marked with white circles. Both experimental values of heat transfer by conduction and evaporation are plotted against the electrical power dissipated in the resistance heater at the center of the evaporator membrane. Table 5.5 also shows the evaporation experiment results over the power range. The inner RTD temperatures ranged from 32^{o}C to 41^{o}C while the outer RTD temperatures ranged from 28^{o}C to 34^{o}C with an uncertainty of ±0.5^{o}C. The evaporation rate ranged from 108μg/sec to 137μg/sec with an uncertainty of ±2.5μg/sec. The experimental energy balance indicates that approximately 71% to 89% of the power dissipated in the resistance heater is conducted out through the silicon membrane. Only 21% to 29% of the power is carried away by the latent heat of evaporation. One major conclusion from the results in Figures 5.1 to 5.4 and Tables 5.2 to 5.5 is that evaporator efficiency decreased as power input increased.

Results from evaporation experiments for tapered micro-channel evaporators (40μm deep with 5μm to 80μm wide micro-channel) are given next. Figure 5.5 and Table 5.6 shows the results of heat transfer rate and evaporation rate measurements for the tapered micro-channel evaporator. Results are given for thermal power inputs of 33.6, 43.7, 57.7mW. For this tapered micro-channel evaporator, the inner and outer RTD temperature ranged from 33^{o}C to 42^{o}C and 29^{o}C to 34^{o}C. The evaporation rate ranged from 103μg/sec to 127μg/sec.

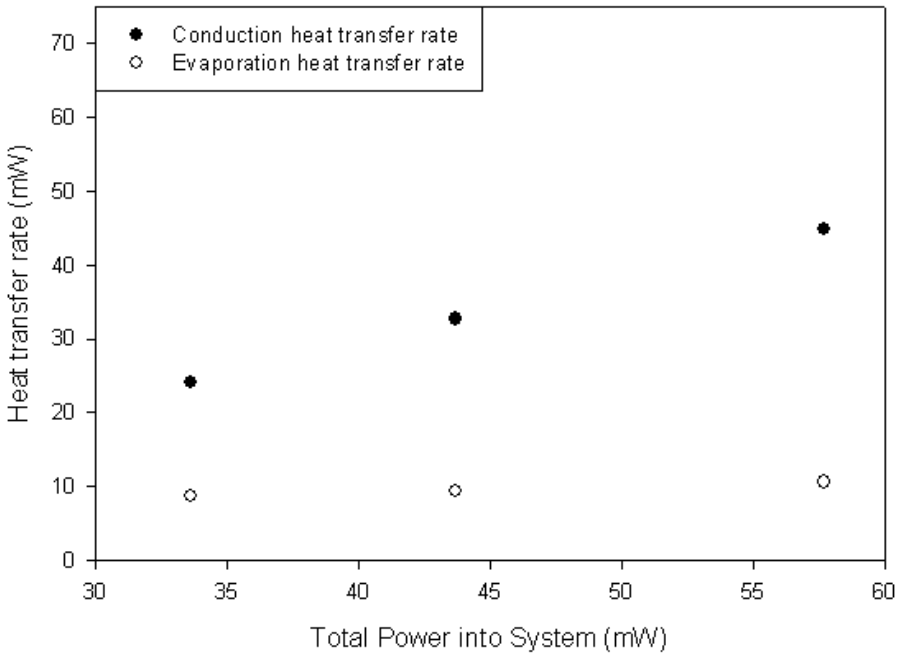


Figure 5.5 Evaporation test results of heat transfer by conduction and evaporation:

Tapered 40 x 5~80 micro-channel evaporator

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	33.6	33.2	28.8	24.1	8.6	103.3	25.6
			43.7	36.9	31.0	32.7	9.3	111.7	21.3
			57.7	41.5	33.5	44.9	10.6	126.7	18.5

Table 5.6 Experimental energy balance:

Tapered 40 x 5~80 micro-channel evaporator

The experimental energy balance indicates that approximately 74% to 81% of the power is conducted out through the silicon evaporator membrane while 21% to 26% of power is carried away by the latent heat of evaporation. Micro-channel evaporator efficiencies decreased as power inputs increased for the tapered micro-channel evaporators just as they did for the rectangular micro-channel evaporators.

5.3 SILICON NITRIDE (SiNx) EVAPORATION TEST RESULTS

Tapered micro-channels was fabricated on 300nm thick Silicon Nitride membranes. Silicon nitride membranes were used for two reasons. First, the thermal conductivity of Silicon Nitride is much lower than Silicon: 15W/mK vs. 148W/mK. Second, Silicon Nitride membranes can be fabricated to be much thinner than silicon membranes: 300nm vs. 2μm. The tapered micro-channels were identical to the tapered micro-channel fabricated on Silicon membranes as before. A single steady state heat transfer experiment was conducted with this SiNx membrane micro-channel evaporator. The results of this experiment are tabulated in Table 5.7. The Table 5.7 shows that for a power input of 13.7mW, the inner RTD temperature is 43^{o}C, and the outer RTD temperature is 28^{o}C. The evaporation rate is 140μg/sec. The experimental energy balance indicates that approximately 14% of the power dissipated in the resistance heater is conducted out through the silicon nitride (SiNx) membrane, while 86% of the power is carried away by the latent heat of evaporation. The efficiency of the micro-channel based on silicon nitride membrane shows a significant improvement over micro-channel evaporators based on Silicon membranes. However, direct comparison was difficult since the power input was different for the two cases.

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	13.7	42.9	27.8	1.24	11.7	139.2	85.4

Table 5.7 Experimental energy balance: Silicon Nitride (SiNx) membrane

Tapered 40 x 5~80 micro-channel evaporator

5.4 SUMMARY OF STEADY STATE EVAPORATION TEST RESULTS

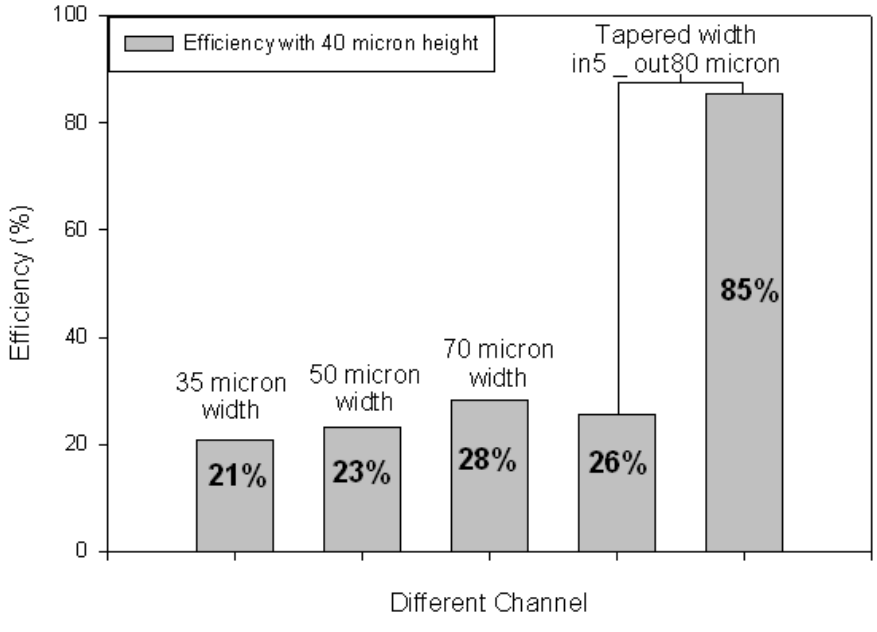


Figure 5.6 Summary of micro-channel evaporator maximum efficiencies

The bar graph shown in Figure 5.6 summarizes the steady state evaporation test results. The efficiencies shown on Figure 5.6 are the maximum efficiencies for each different geometry of micro-channel evaporator. The efficiency results from Figure 5.6 shows no significant difference in maximum evaporator efficiency between the different micro-channel geometries. However, Figure 5.6 shows significant difference in maximum

evaporator efficiency between the different membrane materials (Si and SiNx). The efficiencies are seen to range from 20% to 28% for power input of 34mW for micro-channel evaporator on Silicon membrane. The efficiency is 85% for power input of 13.7mW for micro-channel evaporator on Silicon Nitride membrane.

5.5 TRANSIENT EVAPORATION TEST RESULTS

Next we consider the performance of the micro-channel evaporator changes during the transient operation. The same power inputs (34mW, 44mW, and 57mW) are used to aid in the comparison of transient results with the steady state results. Only one micro-channel geometry, the 40μm deep, the 80μm wide to 5μm wide tapered channels are used in the transient evaporation tests. Recall that previous results indicate that the micro-channel geometry has little effect on the efficiency of the micro-channel evaporators (see Figure 5.6).

Figure 5.7 shows typical temperature histories for a transient evaporation test. Since the transient condition is applied to the testing membrane, the temperature measurements from the RTDs are in a waveform.

Results from the transient evaporation experiment of 10Hz frequency and 50% duty cycles with average thermal power inputs of 34mW, 44mW, and 57mW are shown in Figure 5.8 and Table 5.8. Experimental values of sensible heat transfer by radial conduction and latent heat transfer are plotted against the average electrical power dissipated in the resistance heater at center of the membrane. The black circles and white circles show the experimental values of heat transfer by conduction and evaporation respectively.

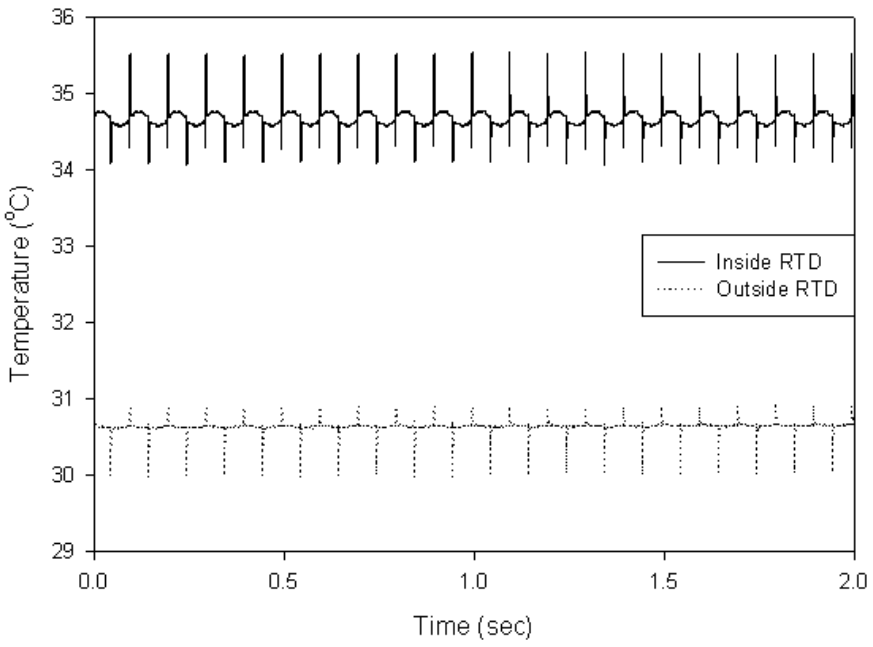


Figure 5.7 Typical temperature histories for transient evaporation test:

10Hz with 50% duty cycle with average power input of 34mW

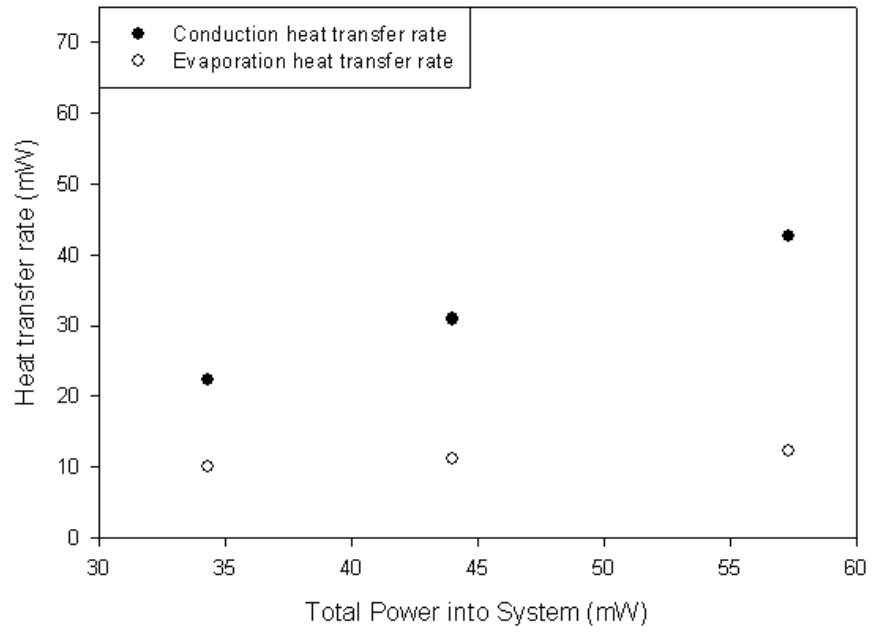


Figure 5.8 Evaporation test results of heat transfer by conduction and evaporation:

Tapered 40 x 5~80 micro-channel evaporator with 10Hz 50% duty cycle

Transient Condition: 10Hz50%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	Tapered 5~80	34.2	3.42	34.8	30.8	22.3	10.0	120.0	29.2
		44.0	4.40	38.5	32.9	30.8	11.2	133.3	24.8
		57.2	5.72	43.5	35.7	42.7	12.3	146.7	21.3

Table 5.8 Experimental energy balance:

Tapered 40 x 5~80 micro-channel evaporator with 10Hz 50% duty cycle

Inner average RTD temperatures range from 34°C to 36°C and outer average RTD temperatures range from 30°C to 36°C with the average power input ranging from 34mW to 57mW. The results of experimental energy balance indicate that approximately 70% to 78% of the power dissipated in the resistance heater is conducted out through the silicon membrane while 22% to 30% of the power is carried away by the latent heat of evaporation. The evaporation rate measured in this experiment is 120μg/sec to 147μg/sec with an uncertainty of ±2.5μg/sec.

Results from the transient evaporation experiment of 10Hz frequency and 30% duty cycles with average thermal power inputs of 34mW, 44mW, and 57mW are shown in Figure 5.9 and Table 5.9. Results from the conduction heat transfer and evaporation heat transfer is denoted with black circles and white circles respectively in Figure 5.9. Results given in Table 5.9 show that over the power range studied (34mW~57mW), the inner average RTD temperature ranged from 37°C to 49°C, the outer average RTD temperature ranged from 33°C to 42°C with temperature uncertainty of ±0.5°C. The evaporation rate ranged from 130μg/sec to 167μg/sec with evaporation rate uncertainty of ± 2.5μg/sec.

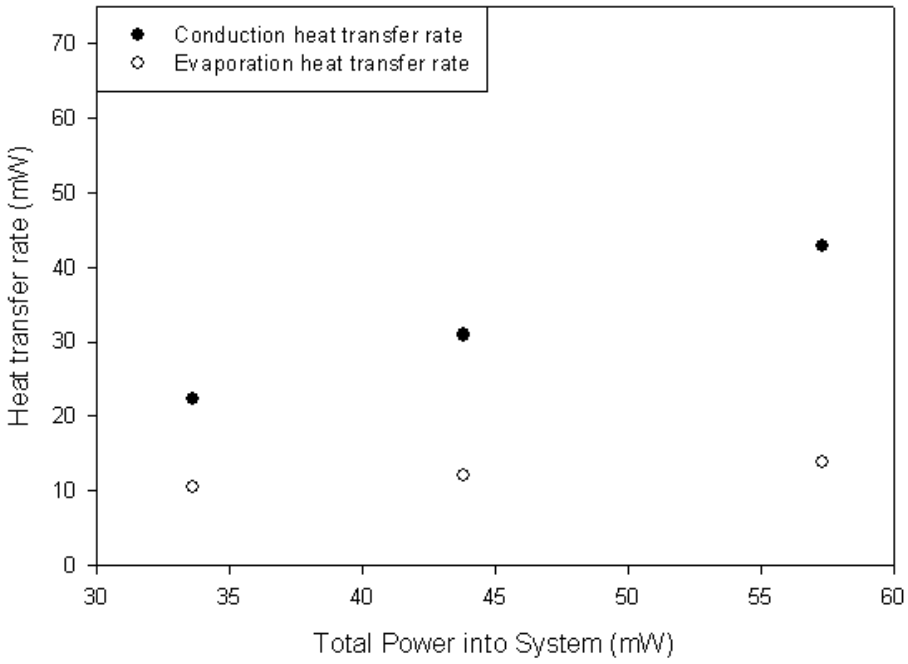


Figure 5.9 Evaporation test results of heat transfer by conduction and evaporation:

Tapered 40 x 5~80 micro-channel evaporator with 10Hz 30% duty cycle

Transient Condition: 10Hz 30%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	Tapered 5~80	33.7	3.37	37.3	33.4	22.5	10.5	130.0	31.9
		43.8	4.38	42.1	36.7	31.0	12.0	148.3	28.2
		57.3	5.73	48.7	41.5	42.8	13.8	166.7	24.6

Table 5.9 Experimental energy balance:

Tapered 40 x 5~80 micro-channel evaporator with 10Hz 30% duty cycle

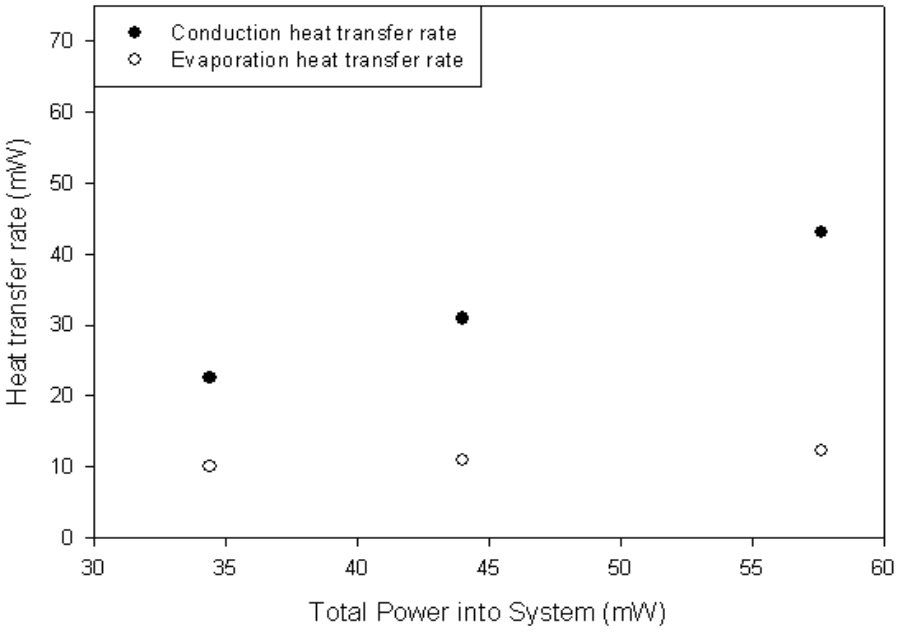


Figure 5.10 Evaporation test results of heat transfer by conduction and evaporation:

Tapered 40 x 5~80 micro-channel evaporator with 20Hz 50% duty cycle

Transient Condition: 20Hz50%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	Tapered 5~80	34.3	1.72	34.8	30.8	22.7	10.0	120.0	29.2
		44.1	2.21	38.5	32.9	30.9	11.0	131.7	24.8
		57.4	2.87	43.5	35.7	43.2	12.3	146.7	21.3

Table 5.10 Experimental energy balance:

Tapered 40 x 5~80 micro-channel evaporator with 20Hz 50% duty cycle

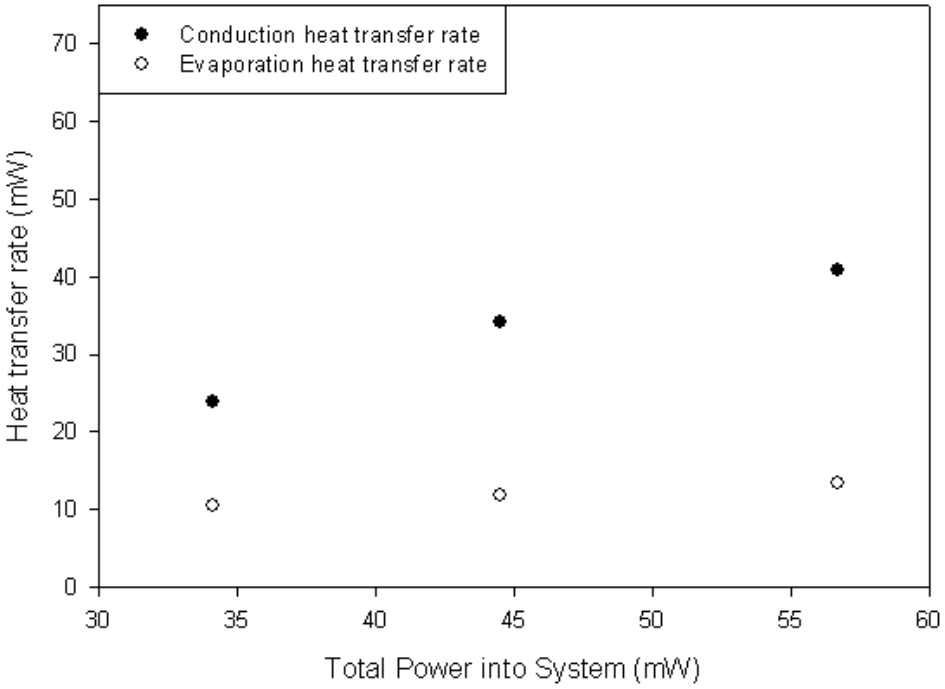


Figure 5.11 Evaporation test results of heat transfer by conduction and evaporation:

Tapered 40 x 5~80 micro-channel evaporator with 20Hz 30% duty cycle

Transient Condition: 20Hz50%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	Tapered 5~80	34.2	1.71	35.2	31.0	23.8	10.5	123.3	30.7
		44.5	2.23	40.6	34.5	34.2	10.8	143.3	26.6
		57.7	2.89	46.9	39.6	40.8	13.3	158.3	23.4

Table 5.11 Experimental energy balance:

Tapered 40 x 5~80 micro-channel evaporator with 20Hz 30% duty cycle

Several attempts were made to conduct transient evaporation test at 10Hz with a 10% duty cycle. However at 10Hz with a 10% duty cycle the transient experiments failed [membrane broke] due to sharp and high thermal effects.

Heat transfer rate results for conduction (black circles) and evaporation (white circles) for 20Hz with a 50% duty cycle are plotted in Figure 5.10. Both Figure 5.10 and Table 5.10 show experimental results for average thermal power inputs 34mW, 44.0mW, and 58mW. With this power range the temperatures resulted in the inner average RTD temperature ranged from 34^{o}C to 44^{o}C and the outer average RTD temperature ranged from 31^{o}C to 36^{o}C with uncertainty of ±0.5^{o}C. Also, the mass evaporation rates are found from 120μg/sec to 147μg/sec with uncertainty of ±2.5μg/sec.

Heat transfer rate results of conduction (black circles) and evaporation (white circles) for 20Hz with a 30% duty cycle are plotted in Figure 5.11. Both Figure 5.11 and Table 5.11 show experimental results for average thermal power inputs of 34mW, 44mW, and 567mW. With this power range the results in the inner average RTD temperatures ranged from approximately 35^{o}C to 47^{o}C and the outer average RTD temperatures ranged from approximately 31^{o}C to 40^{o}C with ±0.5^{o}C. The mass evaporation rates were found to range from 123μg/sec to 158μg/sec with ±2.5μg/sec.

5.6 SUMMARY OF TRANSIENT EVAPORATION TEST RESULTS

The bar graph in Figure 5.12 shows a comparison of efficiencies for transient evaporation test results at each different transient condition. The efficiency results from Figure 5.12 shows no significant difference between the transient conditions. However efficiencies are higher cases with at 10Hz duty cycles than at cases with 20Hz duty cycles.

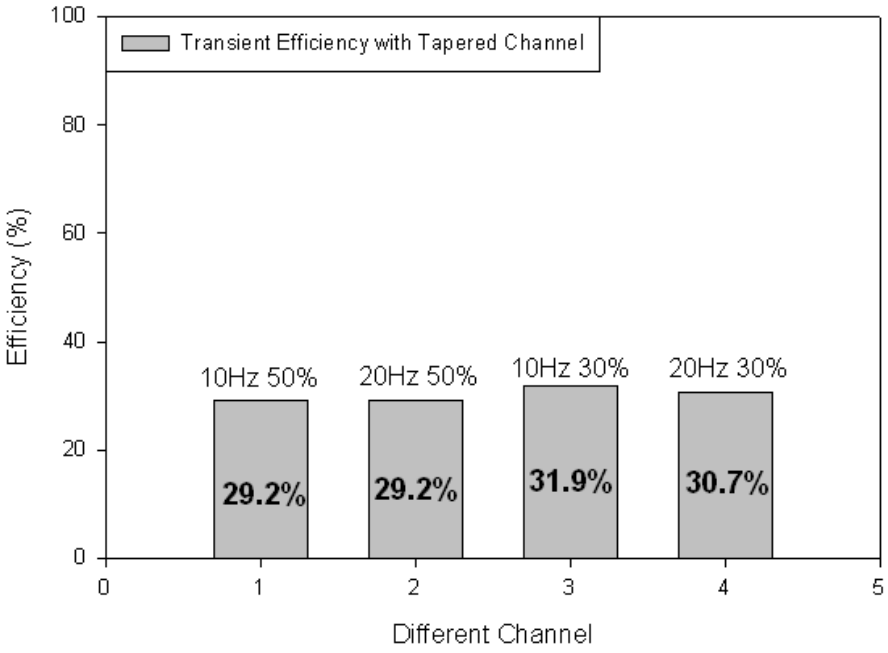


Figure 5.12 Experimental results summary for various transient conditions

Also, Figure 5.12 show that micro-channel evaporator efficiencies decreased as power inputs are increased as in the transient operation test results.

The efficiencies ranged from 29% to 32% at the same power inputs. Transient cases have approximately 10% better efficiency than the steady state cases with same average power inputs. However, the transient state evaporation test micro-channel evaporator efficiencies do not show significant improvement.

5.7 VISUALIZATION TEST RESULTS

The movement of the liquid vapor interface was visualized for constant cross section rectangular micro-channel evaporators at power inputs of 34mW, 44mW and 57mW.

Figures 5.13, 5.14, and 5.15 show the top view of the micro-channel evaporator for 40 x 35 x 5, 40 x 50 x 5, and 40 x 70 x 5 respectively.

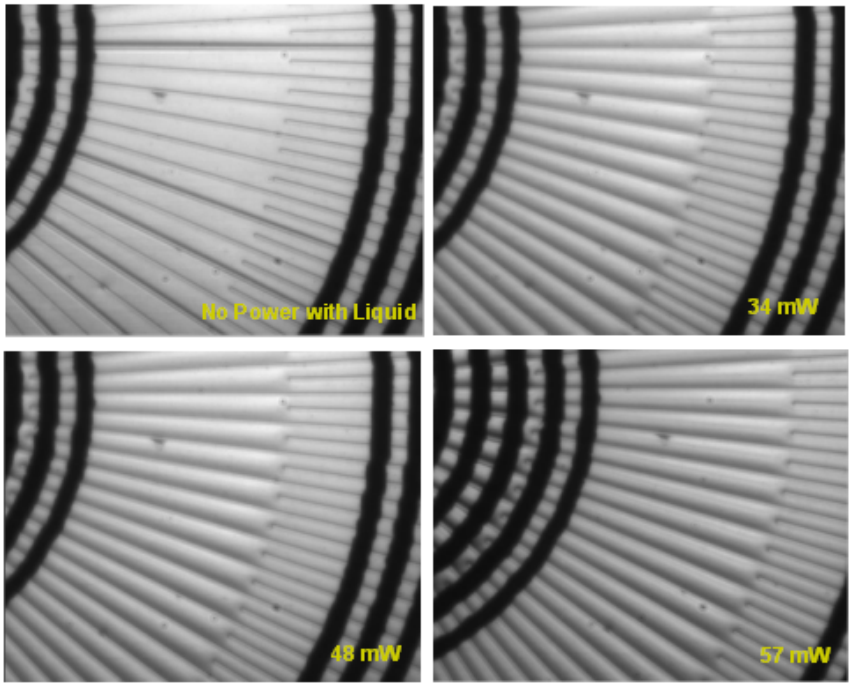


Figure 5.13 Visualization of 40 x 35 x 5 micro-channels with various power inputs

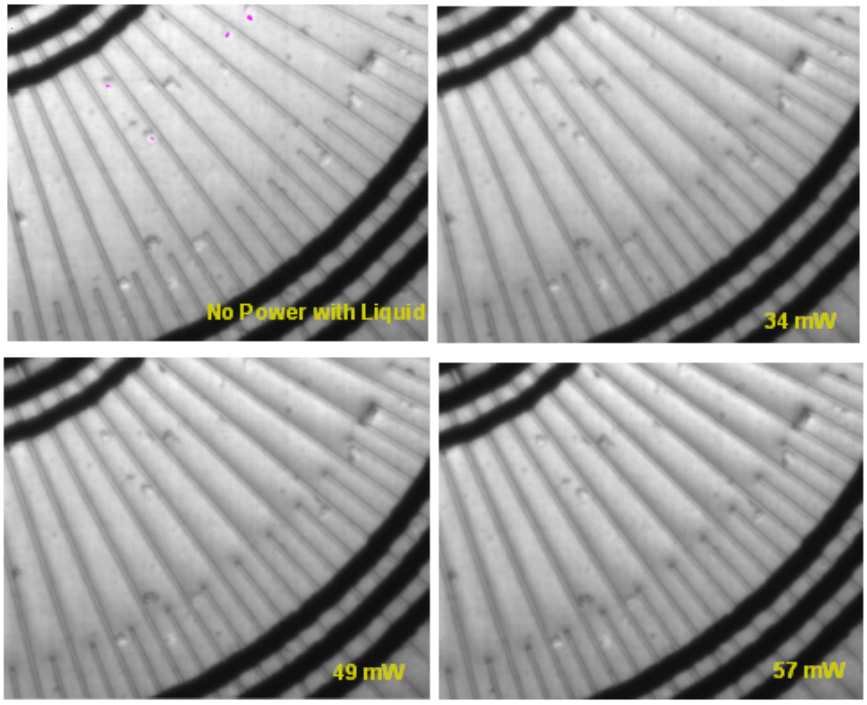


Figure 5.14 Visualization of 40 x 50 x 5 micro-channels with various power inputs

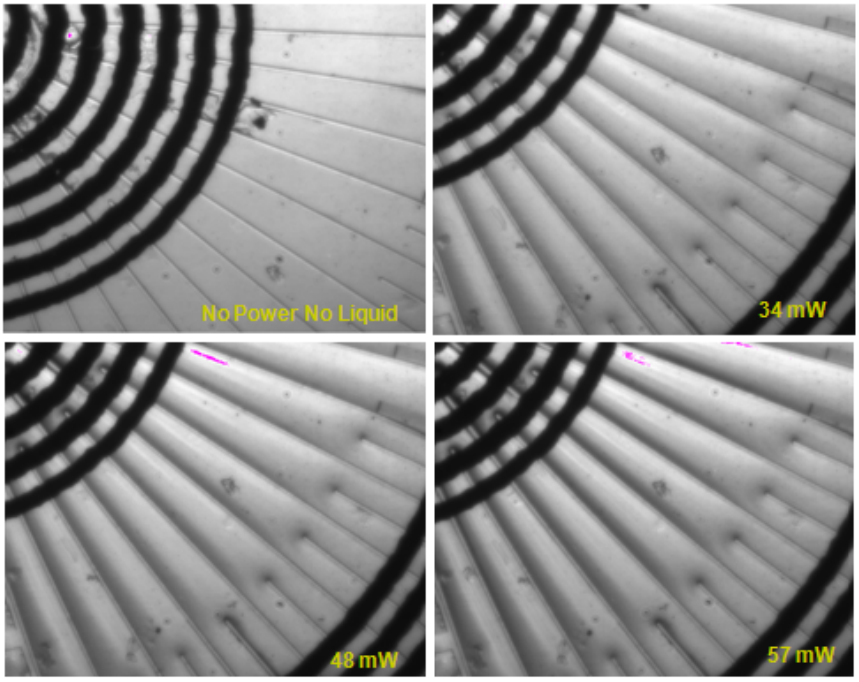


Figure 5.15 Visualization of 40 x 70 x 5 micro-channels with various power inputs

The top-left picture in Figures 5.13, 5.14, and 5.15 show the micro-channels before working fluid is added and before power is input to the resistance heater in the micro-channel evaporator. Each successive picture shows the micro-channels for increasing power. The magnitude of each power input is shown on the pictures.

Figures 5.13, 5.14, and 5.15 show the shadows located close to the wick structures darken as power inputs are increased. In other words, the higher power input makes the working fluid layer thinner and deformation of the meniscus stronger.

Results from a different micro-channel visualization test are shown in Figure 5.16. In Figure 5.16, four micrographs are shown as the working fluid evaporates from 40 x 50 x 5 micro-channels. No power input is applied in these photos. The first micrograph illustrates the empty micro-channels. In next two micrographs, the liquid meniscus is seen to descend in the micro-channels. In the last micrograph, the working fluid is seen to

have retreated to two disconnected corners. In these images, the contact angle Θ of the meniscus remains constant at 41±1°as the level of the working fluid decreases. The radius of curvature of the meniscus remains constant at 33±1μm. Based on this radius of curvature, the capillary pressure in the 50μm wide channels is found to be approximately 700Pa.

Using similar micrographs the contact angle and radius of curvature of the meniscus for the 35μm wide channels are measured to be 41° and 24μm (see Figure 5.17). The capillary pressure in the 35μm wide micro-channels is found to be approximately 1000Pa.

Likewise the contact angle and radius of curvature of meniscus for the 70μm wide channels are measured to be 42° and 47μm (see Figure 5.17).. The capillary pressure in the 70μm wide channels is found to be approximately 500Pa. Micrographs of 35μm and 70μm wide micro-channels are also taken using the same technique as shown in Figure 5.17.

The second visualization test results are tabulated in Table 5.12. The capillary pressure equation is shown in equation 5.1.

$$P_c = \frac{2\gamma \cos\theta}{r} \qquad (5.1)$$

where P_c is capillary pressure, γ is surface tension ($8\text{x}10^{-3}$N/m), θ is contact angle (41^o)

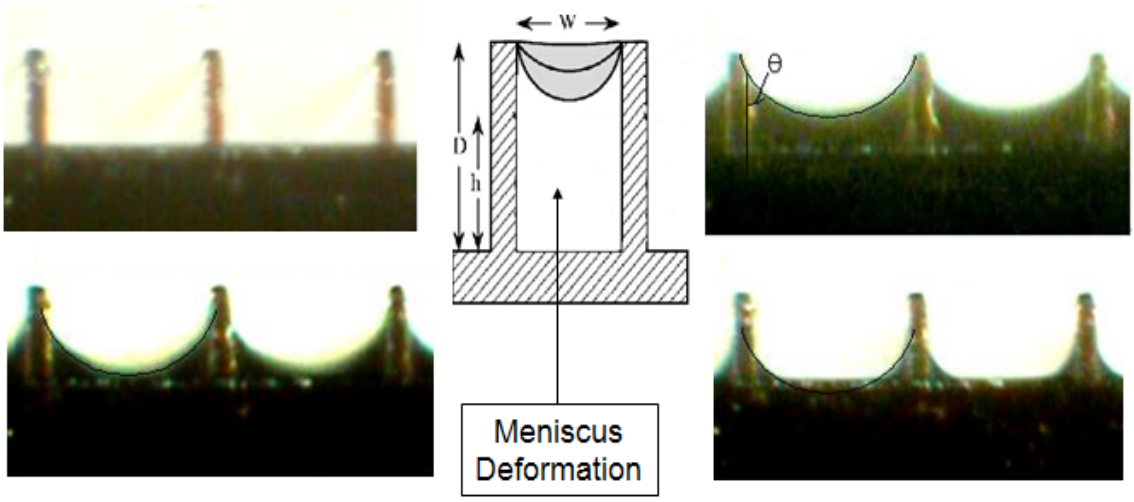


Figure 5.16 Microphotograph of 40 x 50 x5 micro-channels with working fluid, FC77

Figure 5.17 Microphotograph of micro-channels with working fluid, FC77

SU8 Height	Channel Width	Contact angle	Radius of Curvature	Capillary Pressure
(μm)	(μm)	(º)	(μm)	(Pa)
40	35	41	24	1000
40	50	41	33	700
40	70	42	47	500

Table 5.12 Second visualization tests results

CHAPTER 6

DISSCUSION AND COMPARISON OF EXPERIMENAL AND NUMERICAL RESULTS

6.1 COMPARISON OF RESULTS OVERVIEW

Numerical results are presented for steady evaporation and transient evaporation from open top square micro-channels evaporator, and compared to experimental measurements. The micro-channels used in the numerical models match those found in the experimental study. Two geometries are considered: First a geometry that consists of micro-channels of constant rectangular cross section. Each micro-channel is 40μm deep and has constant widths of 35μm, 50μm or 70μm. Micro-channel walls are 5μm thick. The second geometry consists of micro-channels with tapered rectangular cross section. Each micro-channel is 40μm deep. The width of the micro-channels tapers from 80μm at the micro-channels outer radius, down to 5μm at the micro-channels inner radius. The walls of the micro-channels also taper in width from 80μm at their outer radius down to 5μm at their inner radius. The micro-channels walls are assumed to be fabricated of SU8 and fabricated on either a 2μm thick Silicon membrane or a 300nm thick Silicon Nitride membrane.

A table summarizing the numerical simulations is given in Table 6.1 (a) and (b). Table 6.1 (a) shows the numerical simulation conditions of the steady state in this chapter. Table 6.2 (a) shows the numerical simulation conditions of the transient operation.

Material	SU8 Height (μm)	SU8 Wall Thickness (μm)	Channel Width (μm)	Working Fluid Thickness (μm)	Average power input (mW)	Conditions Frequency (Hz) Duty cycle (%)
Si	10	10	10	8	27.1, 30.3, 34.1, 37.2, 41.3	Steady state
Si	40	5	35	14, 20, 28, 35	33.6, 48.4, 57.5	Steady state
Si	40	5	50	14, 20, 28, 35	34.0, 47.2, 62.4	Steady state
Si	40	5	70	20, 30, 36	32.0, 42.5, 53.9	Steady state
Si	40	5~80	5~80	Calculated	34, 44, 57, 82, 107, 124	Steady state
SiNx	40	5~80	5~80	Calculated	13, 10, 7	Steady state

(a) Steady State Conditions

Material	SU8 Height (μm)	SU8 Wall Thickness (μm)	Channel Width (μm)	Working Fluid Thickness (μm)	Average power input (mW)	Conditions Frequency (Hz) Duty cycle (%)
Si	40	5~80	5~80	Calculated	34, 44, 57	10Hz 50%, 30%, 10%
Si	40	5~80	5~80	Calculated	34, 44, 57	20Hz 50%, 30%
Si	40	5~80	5~80	Calculated	34	50Hz 50%
SiNx	40	5~80	5~80	Calculated	11	10Hz 50%

(b) Transient Operation Conditions

Table 6.1 Summary of numerical simulations

6.2 STEADY STATE EVAPORATION TESTS RESULTS (Rectangular Channel)

Steady state evaporation from open top constant cross section micro-channels are considered first. Dimensions of constant rectangular cross section micro-channels are described as channel depth x channel width x channel wall thickness. For example 40 x 35 x 5 refers to a micro-channel that is 40μm deep, 35μm wide with each channel formed between 5μm thick walls as shown in Figure 7.1. Unless otherwise stated all micro-channel evaporators are fabricated of SU8 channel walls fabricated atop 2μm thick silicon membrane.

In these numerical simulations the liquid working fluid thickness in the micro-channels has been specified a priori.

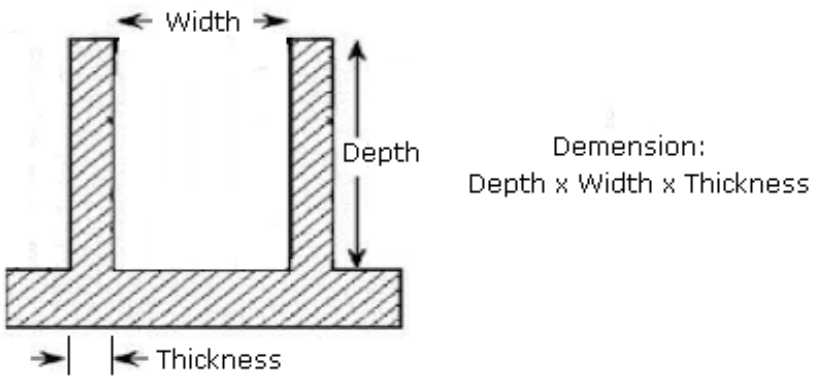
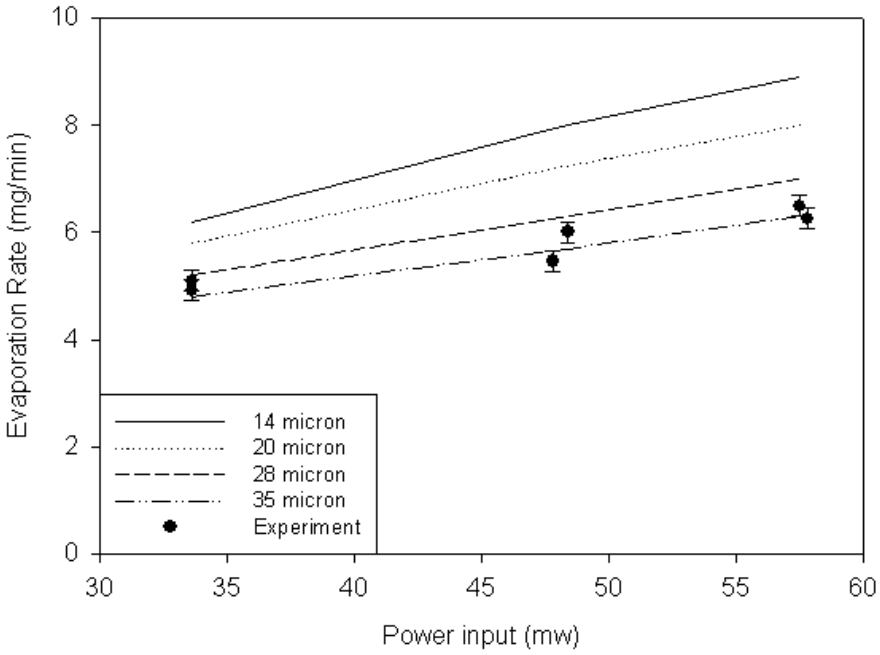


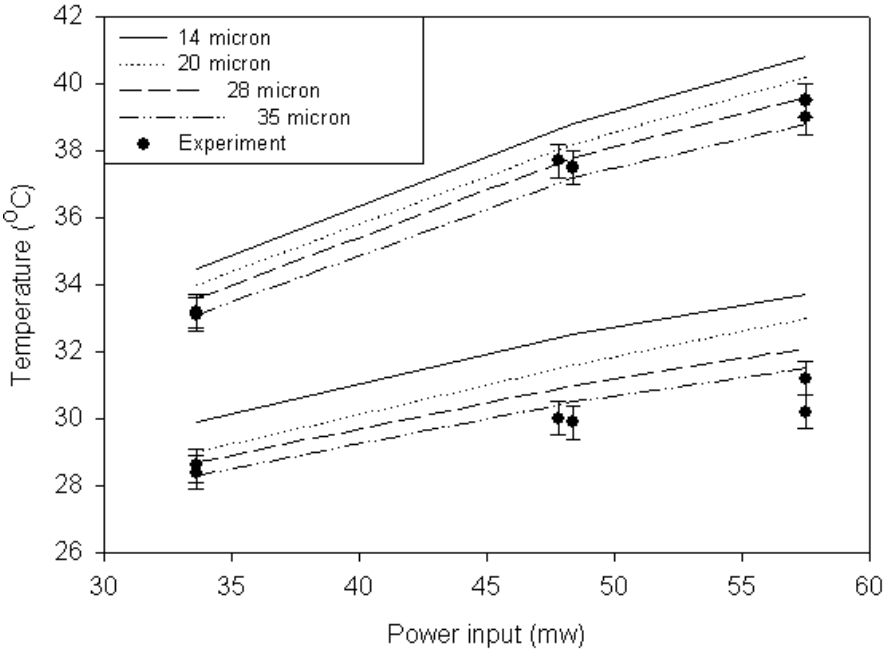
Figure 6.1 Dimensions of Constant Rectangular Cross Section Micro-Channels

	Working Fluid Thickness (µm)	Power into Heater mW)	Inside PRT Temp (°C)	Outside PRT Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Wick Efficiency (%)
Experimental	**Unknown**	**33.6**	**33.1**	**28.6**	**26.9**	**7**	**85.0**	**20.8**
		48.4	**37.5**	**30.0**	**40.5**	**8**	**100.0**	**16.5**
		57.5	**39.0**	**31.2**	**46.3**	**9.1**	**108.3**	**15.8**
Numerical	14	33.6	34.5	29.9	25.0	8.6	103.3	25.6
		48.4	38.8	32.5	36.9	11.5	136.7	22.7
		57.5	40.8	33.7	45.1	12.4	148.3	21.6
Numerical	20	33.6	34.0	29.0	25.5	8.1	96.7	24.1
		48.4	38.2	31.6	37.9	10.5	125.0	21.7
		57.5	40.2	33.0	46.3	11.2	133.3	19.5
Numerical	28	33.6	33.6	28.7	26.3	7.3	86.7	21.7
		48.4	37.8	31.0	39.4	8.8	105.0	18.0
		57.5	39.4	32.1	47.7	9.8	116.7	16.7
Numerical	35	33.6	33.1	28.3	26.9	6.7	80.0	19.9
		48.4	37.2	30.5	40.4	8.0	95.0	16.7
		57.5	38.6	31.2	48.7	8.8	105.0	15.0

Table 6.2 Numerical and Experimental Energy balances for constant rectangular cross section micro-channel 40 x 35 x 5



(a) Numerical and Experimental Evaporation Rates



(b) Numerical and Experimental Temperatures

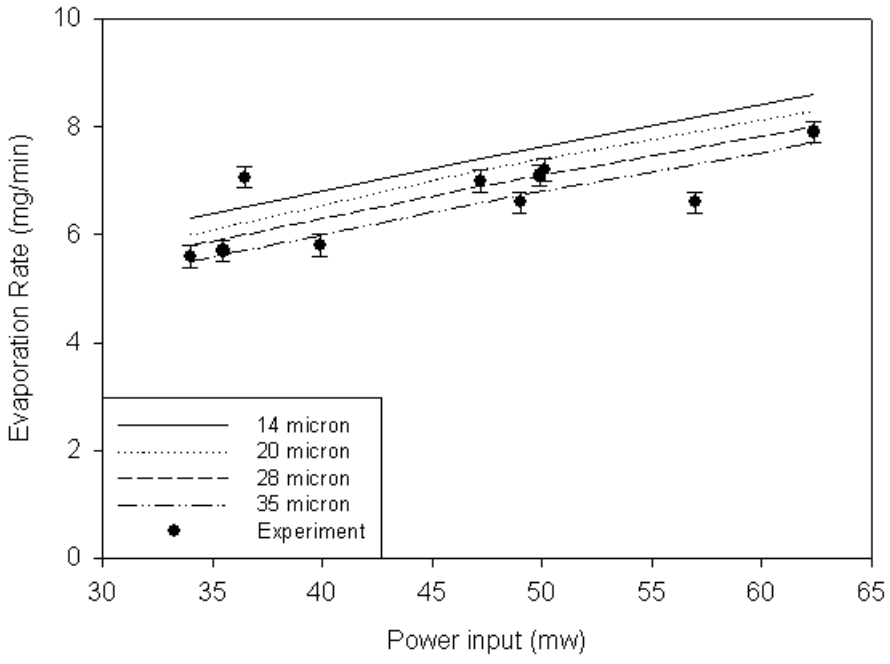
Figure 6.2 Numerical and Experimental Evaporation Rates and Temperatures for constant rectangular cross section micro-channels 40 x 35 x 5

Results from experimental measurements and numerical calculations for constant rectangular cross section micro-channels with 40μm depth, 35μm width and 5μm thick channel walls are shown in Figure 6.2 and tabulated in Table 6.2. In each figure, numerical calculations are indicated with lines, and experimental measurements are indicated with filled circles. Figure 6.2 (a) shows predicted evaporation rates from the micro-channels plotted versus the thermal power input. Predicted evaporation rates are plotted for assumed liquid working fluid thicknesses of 14, 20, 28, and 35μm. Figure 6.2 (b) shows predicted temperatures for the locations of the two annular RTD is plotted versus thermal power input. Once again, predicted temperatures are plotted for assumed liquid working fluid thicknesses of 14, 20, 28, and 35μm.On Figure 6.2 (a) experimental measurements of evaporator rates for three thermal power inputs: 34mW, 48mW, and 57mW are also plotted. On Figure 6.2 (b) experimental measurements of temperatures for the two annular RTDs are plotted for the same three thermal power inputs 34mW, 48mW, and 57mW (see Appendix C). A comparison of the experimental temperatures and evaporation rates with the numerical calculations reveals that the experimental measurements mostly fall between the numerical simulations assuming liquid working fluid thicknesses of 28μm and 35μm. Thus the assumption of a liquid working fluid thickness of 32μm ±4μm leads to a good match between model and measurements.

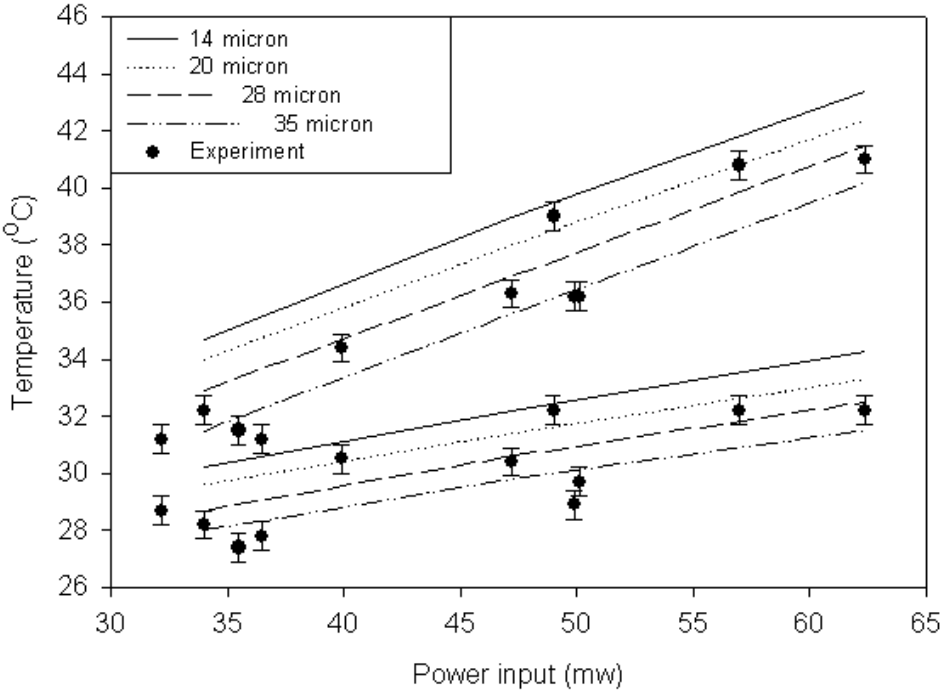
Results from experimental measurements and numerical calculations for constant rectangular cross section micro-channels with 40μm depth, 50μm width and 5μm thick channel walls are shown in Figure 6.3 and tabulated in Table 6.3. In each figure, numerical calculations are indicated with lines, and experimental measurements are indicated with filled circles.

	Working Fluid Thickness (μm)	Power into Heater mW)	Inside PRT Temp (°C)	Outside PRT Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Wick Efficiency (%)
Experimental	**Unknown**	**34.0**	**32.2**	**28.2**	**24.6**	**7.9**	**93.3**	**23.2**
		47.2	**36.3**	**30.4**	**35.4**	**9.8**	**116.7**	**20.6**
		62.4	**41.0**	**32.2**	**51.2**	**11.0**	**131.7**	**17.6**
Numerical	14	34.0	34.7	30.2	25.3	8.7	105.0	25.8
		47.2	39.0	32.2	36.8	10.4	123.3	22.0
		62.4	43.4	34.3	50.5	11.9	143.3	19.1
Numerical	20	34.0	34.0	29.5	25.6	8.4	100.0	24.7
		47.2	38.0	31.3	37.2	10.0	120.0	21.2
		62.4	42.4	33.3	50.8	11.6	138.3	18.6
Numerical	28	34.0	32.9	28.6	26.0	8.0	96.7	23.5
		47.2	36.9	30.6	37.6	9.6	115.0	20.3
		62.4	41.5	32.3	51.2	11.2	133.3	17.9
Numerical	35	34.0	31.5	27.0	26.3	7.7	91.7	22.6
		47.2	35.6	29.2	37.9	9.1	110.0	19.3
		62.4	40.2	31.1	51.4	10.8	128.3	17.3

Table 6.3 Numerical and Experimental Energy balances for constant rectangular cross section micro-channel 40 x 50 x 5



(a) Numerical and Experimental Evaporation Rates



(b) Numerical and Experimental Temperatures

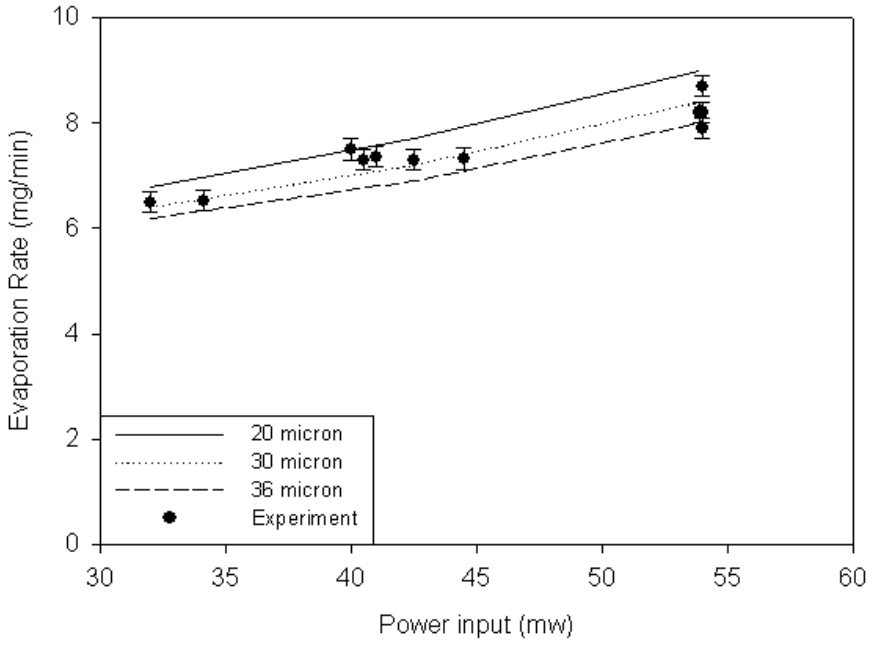
Figure 6.3 Numerical and Experimental Evaporation Rates and Temperatures for constant rectangular cross section micro-channels 40 x 50 x 5

Figure 6.3 (a) shows predicted evaporation rates from the micro-channels plotted versus the thermal power input. Predicted evaporation rates are plotted for assumed liquid working fluid thicknesses of 14, 20, 28, and 35μm. Figure 6.3 (b) shows predicted temperatures for the locations of the two annular RTD is plotted versus thermal power input. Once again, predicted temperatures are plotted for assumed liquid working fluid thicknesses of 14, 20, 28, and 35μm. On Figure 6.3 (a) experimental measurements of evaporator rates for three thermal power inputs: 34mW, 47mW, and 62mW are also plotted. On Figure 6.3 (b) experimental measurements of temperatures for the two annular RTDs are plotted for the same three thermal power inputs 34mW, 47mW, and 62mW (see Appendix C). A comparison of the experimental temperatures and

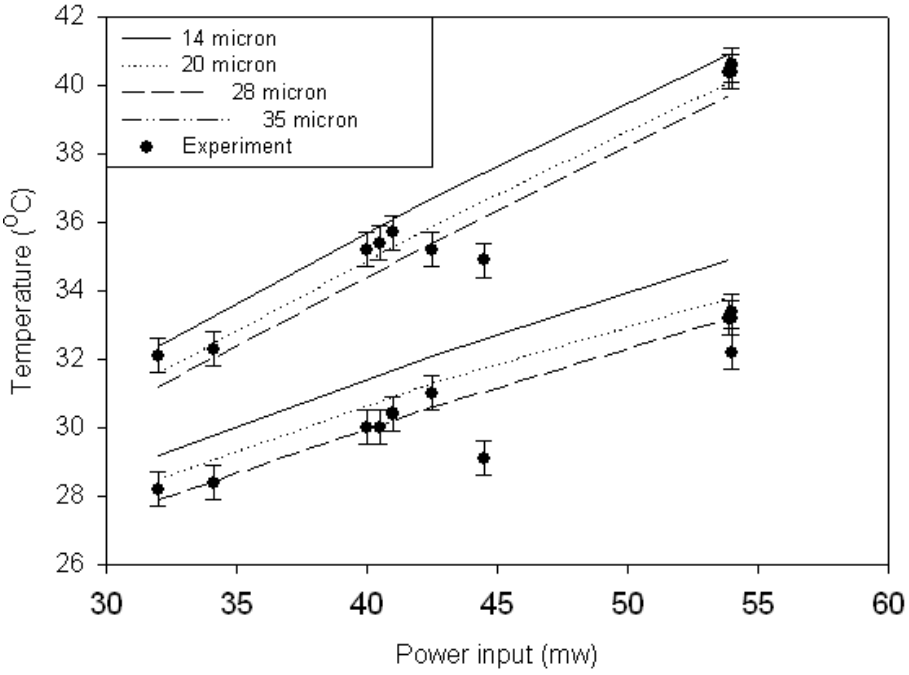
evaporation rates with the numerical calculations reveals that the experimental measurements mostly fall between the numerical simulations assuming liquid working fluid thicknesses of 28μm and 35μm. Thus the assumption of a liquid working fluid thickness of 32μm ±4μm leads to a good match between model and measurements.

Finally, results from experimental measurements and numerical calculations for constant rectangular cross section micro-channels with 40μm depth, 70μm width and 5μm thick channel walls are shown in Figure 6.4 and tabulated in Table 6.4. In each figure, numerical calculations are indicated with solid lines, and experimental measurements are indicated with filled circles. Figure 6.4 (a) shows predicted evaporation rates from the micro-channels plotted versus the thermal power input. Predicted evaporation rates are plotted for assumed liquid working fluid thicknesses of 20, 30, and 36μm. Figure 6.4 (b) shows predicted temperatures for the locations of the two annular RTD is plotted versus thermal power input. Once again, predicted temperatures are plotted for assumed liquid working fluid thicknesses of 20, 30, and 36μm. On Figure 6.4 (a) experimental measurements of evaporator rates for three thermal power inputs: 32mW, 43mW, and 54mW are also plotted. On Figure 6.4 (b) experimental measurements of temperatures for the two annular RTDs are plotted for the same three thermal power inputs 32mW, 43mW, and 54mW (see Appendix C).

A comparison of the experimental temperatures and evaporation rates with the numerical calculations reveals that the experimental measurements mostly fall between the numerical simulations assuming liquid working fluid thicknesses of 30μm and 36μm. Thus the assumption of a liquid working fluid thickness of 33μm ±3μm leads to a good match between model and measurements.



(a) Numerical and Experimental Evaporation Rates



(b) Numerical and Experimental Temperatures

Figure 6.4 Numerical and Experimental Evaporation Rates and Temperatures for constant rectangular cross section micro-channels 40 x 70 x 5

	Working Fluid Thickness (μm)	Power into Heater mW)	Inside PRT Temp (°C)	Outside PRT Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Wick Efficiency (%)
Experimental	**Unknown**	**32.0**	**32.1**	**28.2**	**23.4**	**9.1**	**108.3**	**28.4**
		42.5	**35.2**	**30.0**	**34.8**	**10.3**	**121.7**	**24.2**
		53.9	**40.4**	**33.2**	**44.3**	**11.4**	**136.7**	**21.2**
Numerical	20	32.0	32.4	29.2	22.5	9.5	113.3	29.7
		42.5	35.7	31.1	31.8	10.7	128.3	25.2
		53.9	40.9	34.9	41.1	12.8	153.3	23.7
Numerical	30	32.0	31.6	28.5	23.1	8.9	106.7	27.8
		42.5	34.9	30.3	32.1	9.9	118.3	23.3
		53.9	40.1	33.8	42.2	11.7	140.0	21.7
Numerical	36	32.0	31.2	27.9	23.3	8.7	103.3	27.2
		42.5	34.4	29.6	32.9	9.6	115.0	22.6
		53.9	39.7	33.2	42.7	11.2	133.3	20.8

Table 6.4 Numerical and Experimental Energy balances for constant rectangular cross section micro-channel 40 x 70 x 5

Comparison of the results of the numerical simulations with the results of the evaporation experiments for constant rectangular cross section micro-channels indicates that given an appropriate assumed liquid working fluid thickness in the micro-channel, evaporation rates, sensible heat transfer rates and micro-channel temperatures can be predicted with fair accuracy.

6.3 STEADY STATE EVAPORATION TESTS RESULTS (Tapered Channel)

Steady state evaporation from open top tapered micro-channels are considered next. Dimensions of tapered micro-channels are 40μm deep and their width is 80μm at the outer radius of the micro-channel. Each micro-channel width tapers down to 5μm at the inner radius of the micro-channel. The walls dividing micro-channels also are 80μm wide at their outer radius and taper down to a width of 5μm at their inner radius.

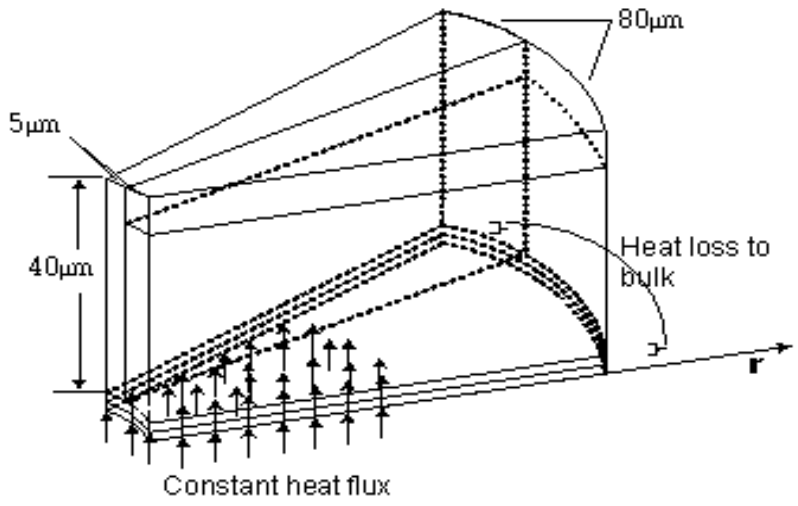
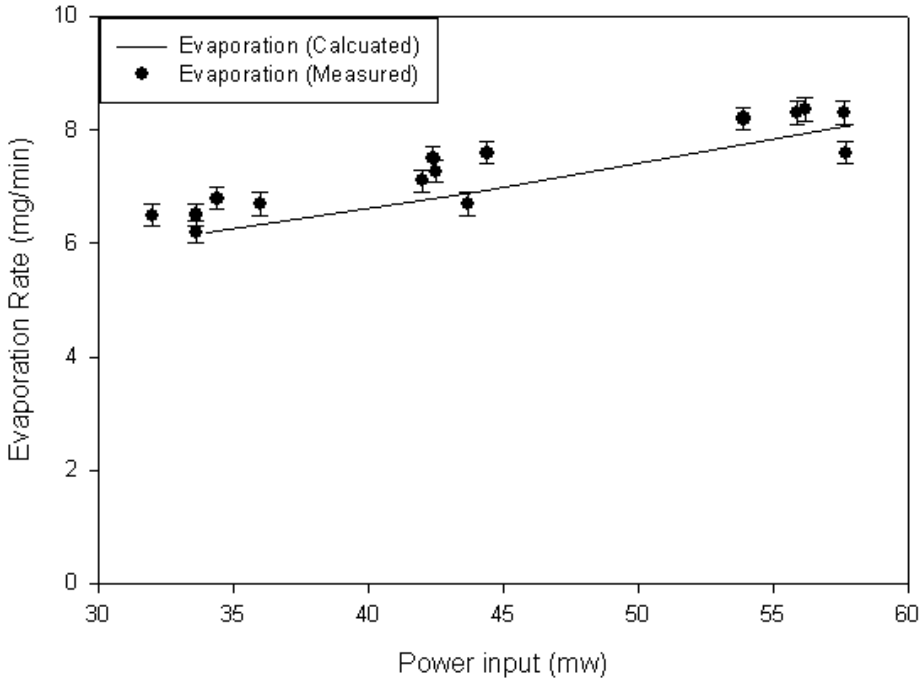


Figure 6.5 Dimensions of Tapered Micro-Channel

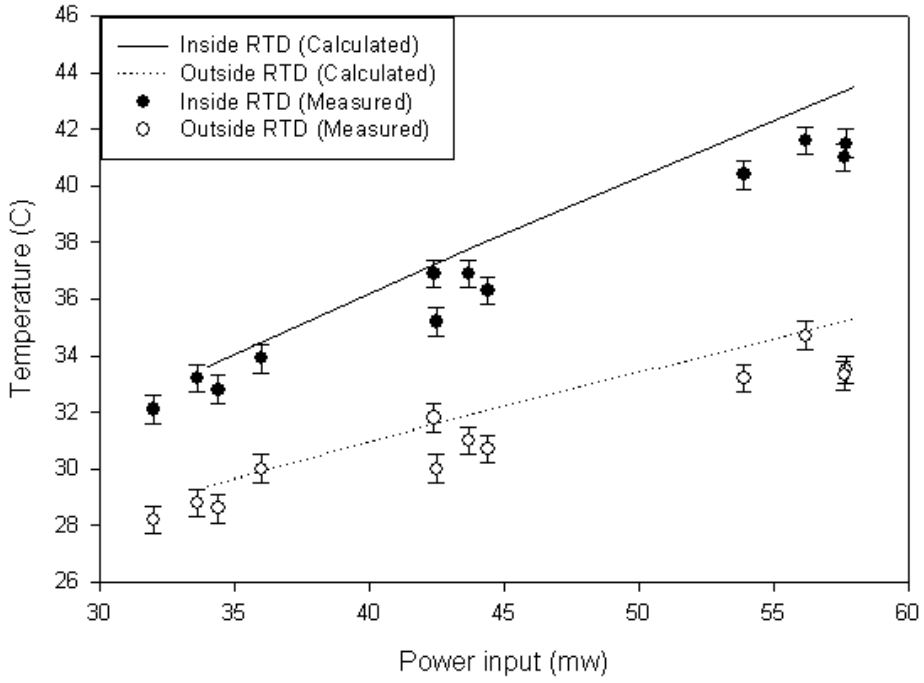
Status	Working Fluid Thickness (μm)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	Unknown	34.0	33.2	28.8	24.1	8.6	103.3	25.6
		44.0	36.9	31.0	32.7	9.3	111.7	21.3
		58.0	41.5	33.5	44.9	10.6	126.7	18.5
Numerical	Calculated	34.0	33.6	29.4	25.4	8.6	103.3	26.3
		44.0	38.1	32.3	34.2	9.8	116.7	22.3
		58.0	43.5	35.3	46.8	11.2	135.6	19.3

Table 6.5 Numerical and Experimental Energy balances for

Tapered micro-channel 40 x 5~80



(a) Numerical and Experimental Evaporation rate



(b) Numerical and Experimental Temperatures

Figure 6.6 Numerical and Experimental Evaporation Rates and Temperatures for Tapered micro-channels 40 x 5~80

As before the micro-channel evaporators are fabricated of SU8 channel walls fabricated on 2μm thick silicon membrane. Since the wall thickness and channel width are tapered same dimensions, dimensions of tapered micro-channels can be described as 40 x 5~80. The dimensions of tapered micro-channels are shown in Figure 6.5. In these numerical simulations, the liquid working fluid thickness in the micro-channels is not specified a priori. The micro-channels are initially assumed to be filled with liquid. At all subsequent times, the liquid working fluid thickness is calculated.

Thus Figures 6.6 through 6.20 shows results from the numerical model that includes calculation of the liquid fluid flow through the open micro-channels based on the force balance equations. Figure 6.6 shows numerical and experimental results for tapered micro-channels 40μm deep with widths tapering from 80μm to 5μm for thermal power inputs of 34, 44, and 58mW (see Appendix C). The numerical results are indicated with lines, and the experimental results are indicated with symbols.

Figure 6.6 (a) shows evaporation rates. Figure 6.6 (b) shows temperatures at the inner (solid line) and outer (dotted line) annular RTDs versus thermal power inputs. Measured evaporation rates (black circles) and measured temperatures for the inner (black circles) and outer (white circles) annular RTDs show good agreement with the numerical results. Table 6.5 shows that for the tapered micro-channel (40μm deep with 5μm to 80μm tapered width) the inner and outer RTD temperature ranged from 33^{o}C to 44^{o}C and 29^{o}C to 35^{o}C. The evaporation rate ranged from 103.3μg/sec to 135.6μg/sec. Both experimental and numerical energy balances indicates that approximately 70% to 80% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while 20% to 30% of power is carried away by the latent heat of evaporation.

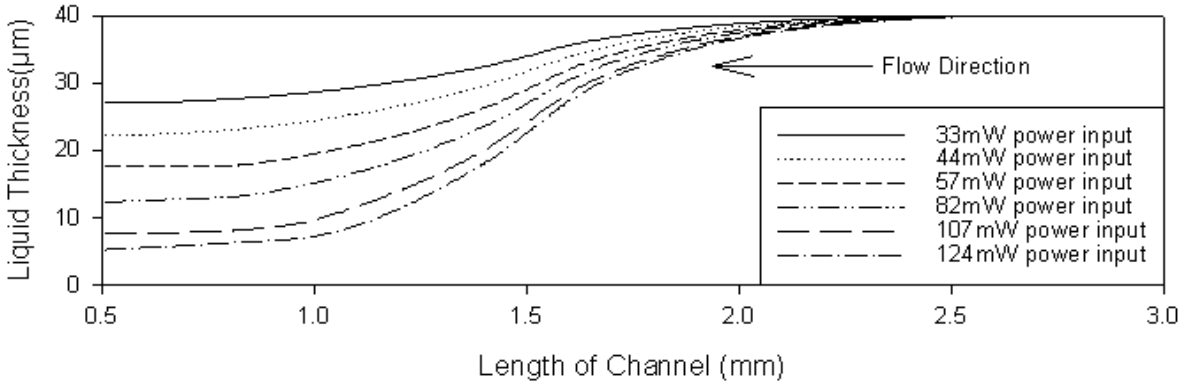


Figure 6.7 Liquid thickness profiles along the length of the channel: Critical heat flux

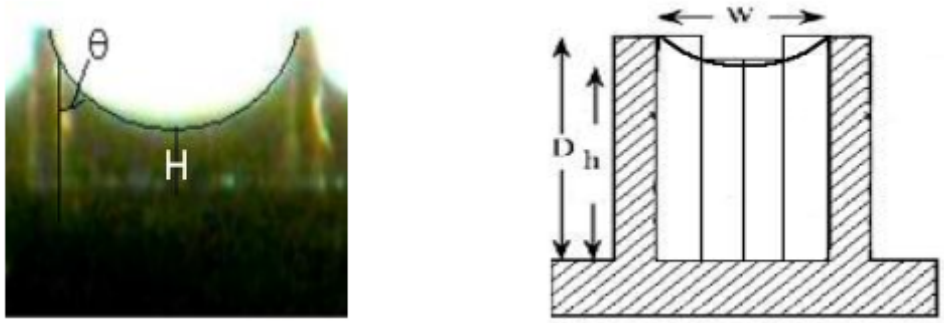


Figure 6.8 Discretization of meniscus in the micro-channel for FDTD model

	Status	Power into Heater mW)	Inside PRT Temp (°C)	Outside PRT Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Wick Efficiency (%)
Numerical	**Matched with experiment**	34	33.6	29.4	25.4	8.6	103.3	26.3
		44	38.1	32.3	34.2	9.8	116.7	22.3
		57	42.8	34.7	46.0	11	133.3	19.3
Numerical	Expected result from calculation	82	48.5	36.6	69.8	12.2	145.0	14.9
		107	55.1	38.6	94.2	12.8	153.3	12.0
		124	59.8	40.0	110.8	13.2	156.7	10.6

Table 6.6 Numerical integration of energy balance for critical heat flux

Tapered 40 x 5~80 micro-channel evaporator

Results also show that micro-channel evaporator efficiencies decreased as power inputs increased in the tapered micro-channels.

Figure 6.7 shows predictions of the liquid thickness profile along the length of the channel calculated by the near the wall of the micro-channel numerical model. The initial liquid thickness is set at 40μm (D in Figure 6.8). The initial liquid thickness near the center of the micro-channel (H in Figure 6.8) is calculated using Equation 6.1.

$$H = D - \frac{w}{2}\tan\theta \qquad (6.1)$$

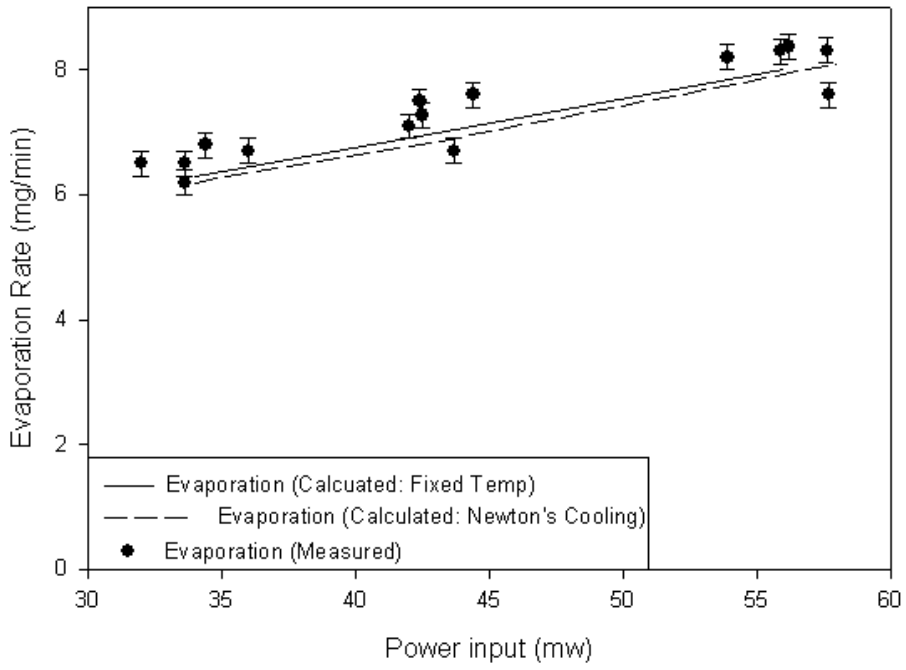
where H is working fluid thickness from center of channel (see Figure 6.7), D is height of wick structures, w is width of channel, and θ is contact angle (41^o).

The plot shows the liquid thickness decreases where the resistance heaters are located. Liquid thickness also decreases as the power input increases.

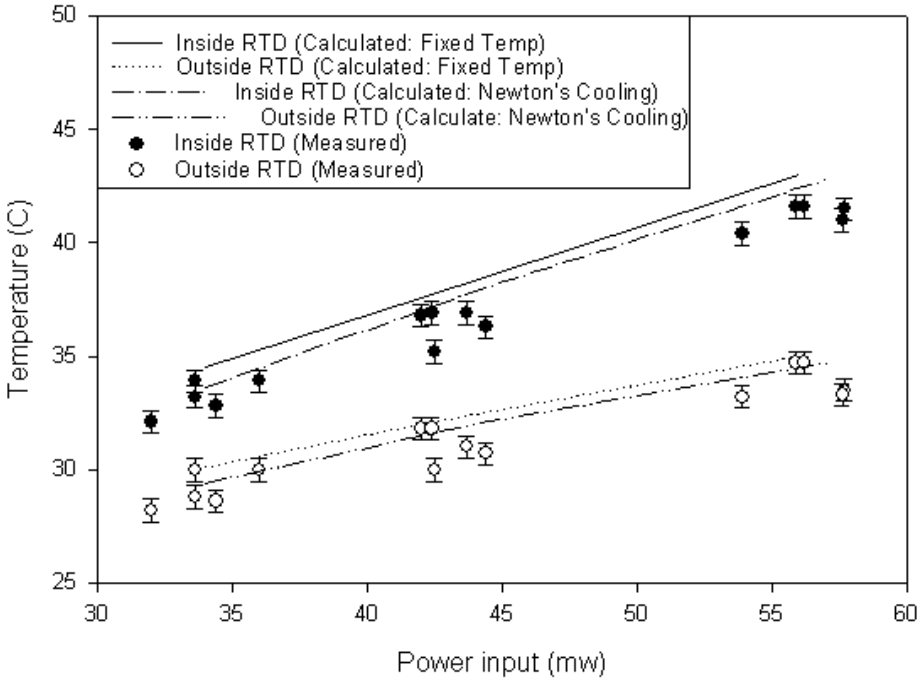
The critical heat flux is found by increasing the power input until the liquid thickness at center of channel reaches zero. The liquid thickness reaches zero when power input reaches 124mW. Table 6.6 shows the calculated radial temperatures, the evaporation rates, the sensible heat transfer rates and the latent heat transfer rates as the liquid thickness at center of the channel goes to zero with increasing the power input. Results also showed micro-channel evaporation efficiencies decrease as power input increases until power input reaches the critical heat flux. If the power input is increased past the critical heat flux value, the calculated temperature will not converge because of the liquid dry out.

6.4 FIXED TEMPERATURE BOUNDARY RESULTS

Two approaches were used to model the boundary conditions at the edge of the membrane for the numerical calculations as explained in Chapter 4. First, a Newton's law of cooling boundary condition was used. The effective convective coefficient was determined by fitting the numerical model to experimental data as explained in Chapter 4. Using a Newton's law of cooling boundary condition with an effective convective coefficient in the numerical model worked well predict the experimental results for micro-channel evaporators on silicon membrane. However, use of a new effective convective coefficient boundary would need to be found for any other kinds of micro-channel evaporators. To effectively predict heat transfer rates for micro-channel evaporators on other membrane material such as silicon nitride, a new boundary condition was preferred. To meet this need a constant temperature boundary condition, employing the temperature at the edge of the square membrane was used. To establish this constant temperature boundary condition, a third RTD was fabricated to measure the membrane edge temperature. This edge temperature was then used as fixed temperature boundary conditions in the numerical calculations. The numerical results with the fixed temperature boundary conditions taken the experimental measurements are plotted alongside to the experimental results in Figure 6.9. Figure 6.9 shows numerical and experimental results for a tapered micro-channel with a depth of 40μm high and width that tapered from 80μm down to 5μm. The membrane supporting the micro-channel was a 2μm thick silicon membrane. Results are shown for thermal power inputs of 34, 44, and 57mW (see Appendix C). The edge temperatures used as the fixed temperature boundary conditions are shown in Table 6.7.



(a) Numerical and Experimental Evaporation rate



(b) Numerical and Experimental Temperature

Figure 6.9 Numerical and Experimental Evaporation Rates and Temperatures for Tapered micro-channels 40 x 5~80with edge temperature boundary condition

Status	Channel Type (μm)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Edge RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	Tapered 5~80	34.0	33.9	30.0	28.1	23.4	9.2	108.3	27.4
		42.0	36.8	31.8	30.3	29.7	9.9	118.3	23.6
		56.0	41.6	34.7	33.2	41.4	11.6	138.3	20.7
Numerical	Tapered 5~80	34.0	34.5	30.1	28.1	24.9	8.9	105.0	26.2
		42.0	38.1	32.5	29.9	31.5	9.7	115.6	23.2
		56.0	42.9	34.8	33.0	45.1	10.8	129.1	19.4

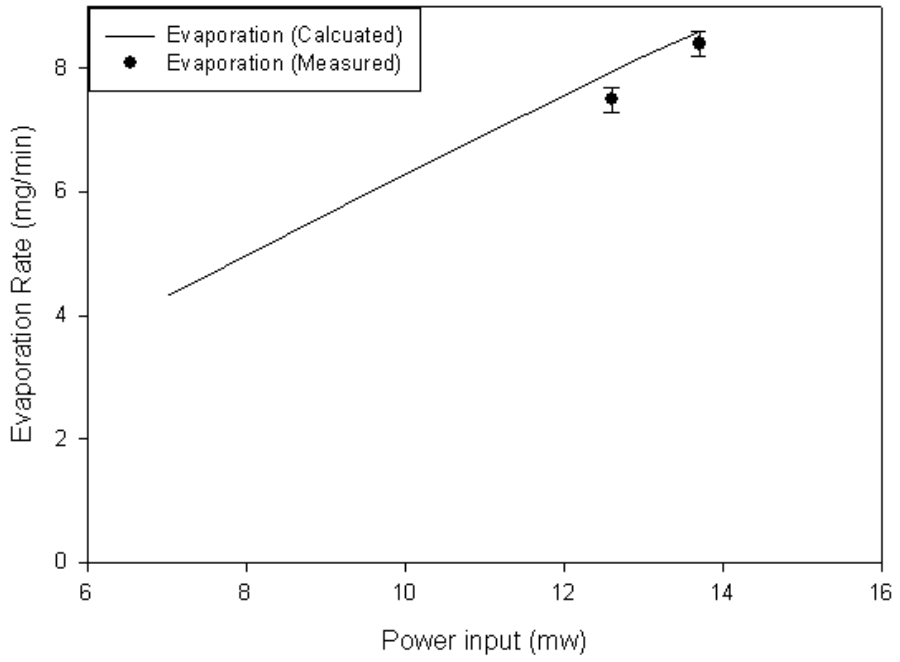
Table 6.7 Numerical and Experimental Energy balances for

Tapered channel 40 x 5~80 with edge temperature boundary condition

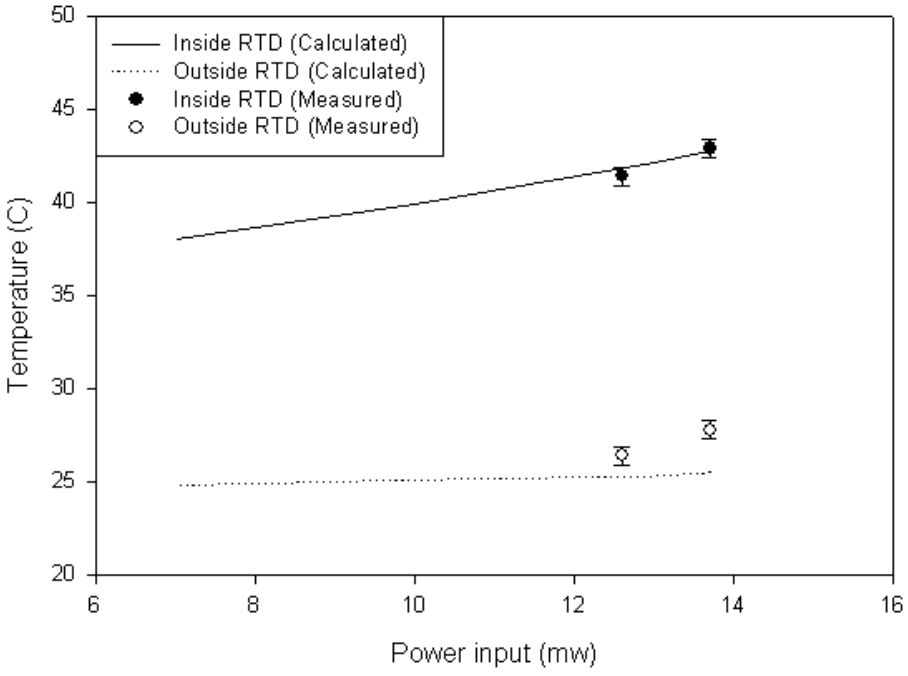
Figure 6.9 (a) shows calculated evaporation rates versus thermal power input. Figure 6.9 (b) shows calculated temperatures at the inner and outer RTDs versus thermal power input. Numerically predicted evaporation rates and temperatures are shown to agree well with measured evaporation rates and inner and outer annular RTD measurements. Like the numerical model that uses the effective convective coefficient, Table 6.7 shows that over the power inputs studied, the numerical model employing a fixed temperature boundary condition is seen to predict temperatures and sensible and latent heat transfer rates as well as the numerical model employing a Newton's law of cooling boundary condition with an effective heat transfer coefficient. As will be seen in the next section the numerical model with the fixed temperature boundary condition can be used to predict results for a micro-channel evaporator built on a silicon nitride membrane just as well.

6.5 STEADY STATE EVAPORATION TESTS RESULTS (Tapered Channel: SiNx)

A second set of tapered micro-channels was fabricated on 300nm thick Silicon Nitride membranes. Silicon Nitride membranes were used for two reasons. First, the thermal conductivity of Silicon Nitride is much lower than Silicon: 15W/mK vs.148 W/mK. Second Silicon Nitride membranes can be fabricated to be much thinner than silicon membranes: 300nm vs. 2μm. The tapered micro-channels were identical to the tapered micro-channels fabricated on Silicon membranes as shown in Figure 6.4. As before the tapered micro-channels were 40μm deep with channel widths that tapered from 80μm at their outer radius to 5μm at their inner radius. Likewise channel wall thickness tapered from 80μm down to 5μm. **N**umerical and experimental results are shown in Figure 6.10, Figure 6.11 and Table 6.8. Results from the evaporation experiments are indicated with symbols and results from the numerical simulations are indicated with lines in Figure 6.10 and Figure 6.11. Figure 6.10 (a) shows comparison of evaporation rates. Figure 6.10 (b) shows comparison of temperatures at the inner and outer RTD versus thermal power inputs, and Figure 6.11 shows the comparison of temperatures along with the length of membrane. In Figure 6.10, experimental results are given for power inputs of 12.6mW and 13.7mW (see Appendix C). The reasons why the low 13mW of power inputs are used in the silicon nitride membrane were previously explained in Chapter 5. Difficulty in satisfying the energy balance in the silicon nitride membrane simulations resulted in 6% and 10% energy balance errors for 13.7mW and 12.6mW power inputs respectively. However, Figure 6.10 shows the numerical calculations and the experimental measurements compare well. The fixed temperature boundary condition used in the Figure 6.10 is the edge temperature measured for the experiment at 13.7mW power input.



(a) Numerical and Experimental Evaporation rate for SiNx membrane



(b) Numerical and Experimental Temperature for SiNx membrane

Figure 6.10 Numerical and Experimental Evaporation rate and temperature: SiNx Tapered channel 40 x 5~80 with edge temperature boundary condition

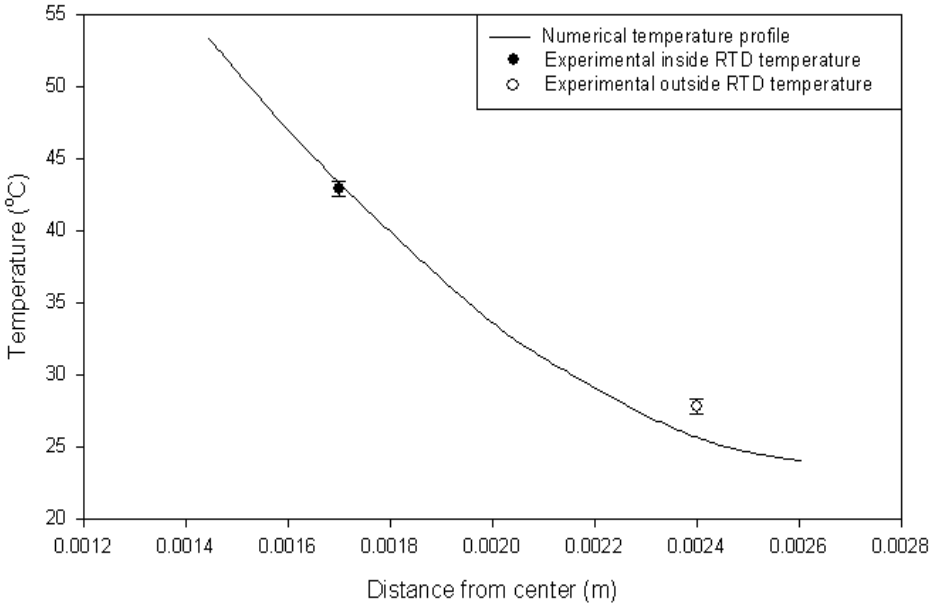


Figure 6.11 Numerical and Experimental Temperature for silicon nitride membrane

Tapered channel 40 x 5~80 with edge temperature boundary condition

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Edge RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (mg/min)	Efficiency (%)
EXPERIMENTAL RESULTS										
40	5~80	5~80	13.7	42.9	27.8	22.8	1.24	11.7	8.4	85.4
			12.6	41.4	26.4	22.8	1.1	10.3	7.5	81.7
NUMERICAL RESULTS										
40	5~80	5~80	13.7	42.9	25.3	22.8	1.6	12.1	8.6	88.5
			13	42.1	25.1	22.8	1.5	11.5	8.2	88.4
			10	39.9	25.0	22.8	1.3	8.7	6.3	87.0
			7	38.0	24.8	22.8	1.0	6.0	4.3	85.7

Table 6.8 Numerical and Experimental Energy balances for SiNx membrane

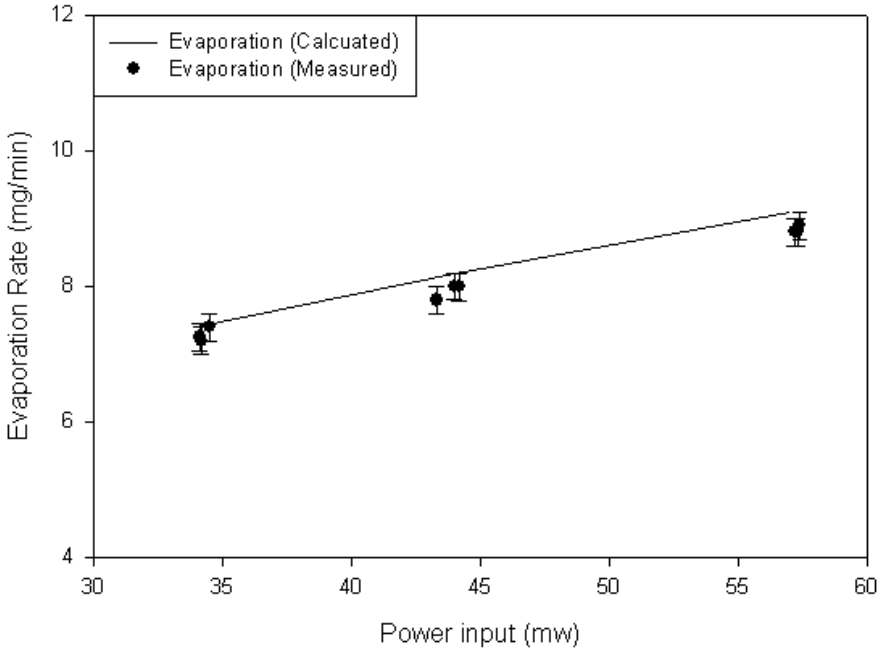
Tapered channel 40 x 5~80 with edge temperature boundary condition

The fixed temperature was measured to be 22.8°C with an uncertainty of ±0.5°C. Figure 6.11 shows temperature measurements for the inner and outer RTD temperatures at a thermal power input of 13.7mW. Figure 6.11 shows that the numerical result for the inner RTD temperature closely match the temperature from the experimental measurements. However the predicted outer RTD temperature is lower than the measured temperature because of model assumptions. The numerical model assures a radially symmetric geometry were the experimental micro-channel evaporator is a square geometry. This mismatch causes greater error near the outside of the square micro-channel evaporator.

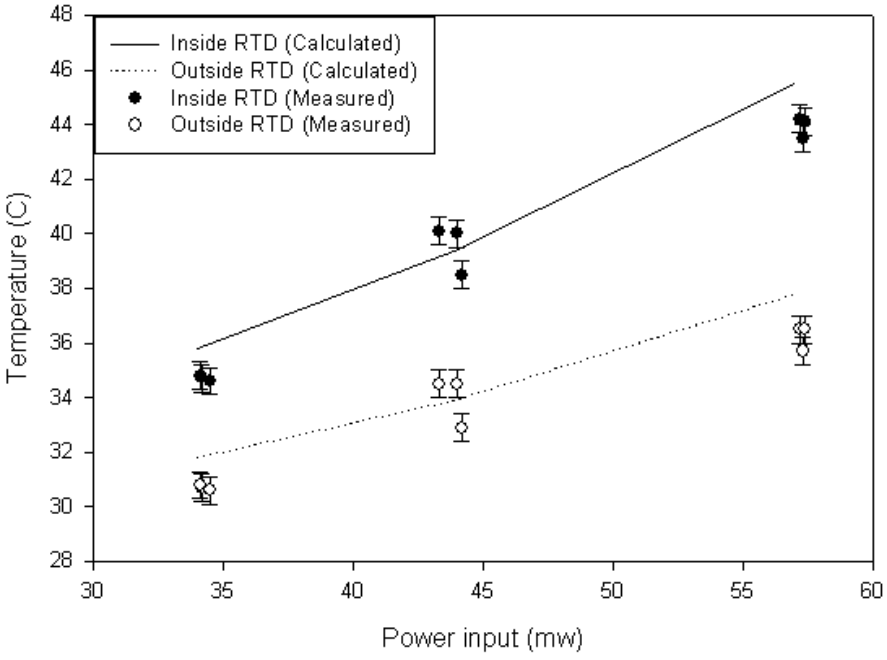
Table 6.8 shows that micro-channel evaporator built in a silicon nitride membrane (with micro-channel 40μm high and tapered 5μm to 80μm wide) has the best wick efficiency at the highest thermal power input of 13.7mW. The results of the numerical simulations given in Table 6.8 show that over the power range studied, the inner and outer RTD temperature ranged from 38°C to 42°C and approximately 25°C respectively. The evaporation rate ranged from 72μg/sec to 137μg/sec. Both experimental and numerical energy balances indicate that only 12% to 15% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while 85% to 88% of the power is carried away by the latent heat of evaporation.

6.6 TRANSIENT TEST RESULTS (Tapered Micro-channel: Si)

The experimental and numerical results for transient operation of the micro-channel evaporator are presented next. The same average power inputs as the steady-state runs of approximately 34mW, 44mW, and 57mW are used (see Appendix C).



(a) Numerical and Experimental Evaporation rate



(b) Numerical and Experimental Temperature

Figure 6.12 Numerical and Experimental Evaporation rate and temperature:

Tapered channel 40 x 5~80, 10Hz frequency 50% duty cycle with various power inputs

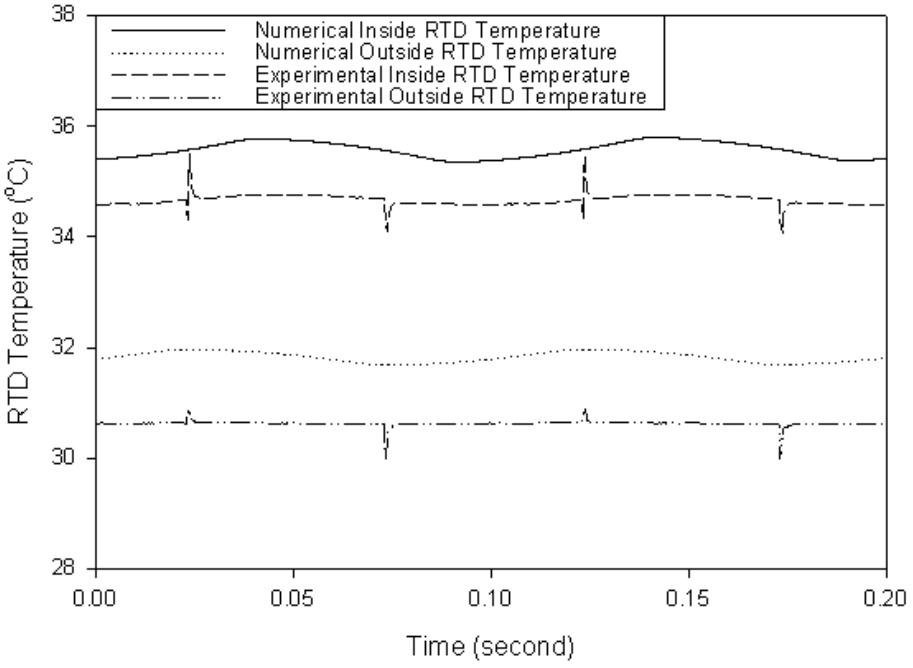


Figure 6.13 Numerical and Experimental Temperature History:

Tapered channel 40 x 5~80, 10Hz frequency 50% duty cycle with 34mW power input

Transient Condition: 10Hz50%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	**Tapered 5~80**	**34.2**	**3.42**	**34.8**	**30.8**	**22.3**	**10.0**	**120.0**	**29.2**
		44.0	**4.40**	**38.5**	**32.9**	**30.8**	**11.2**	**133.3**	**24.8**
		57.2	**5.72**	**43.5**	**35.7**	**42.7**	**12.3**	**146.7**	**21.3**
Numerical	Tapered 5~80	34.0	3.40	35.8	31.8	23.7	10.3	123.3	30.3
		44.0	4.40	39.4	33.9	32.5	11.5	138.3	26.1
		57.0	5.70	45.5	37.8	44.4	12.6	151.7	22.2

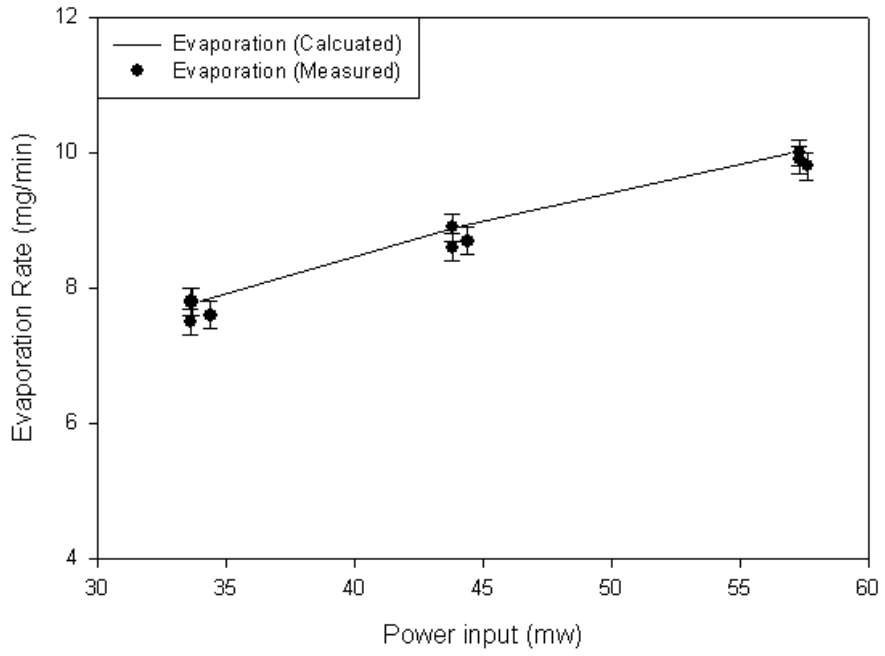
Table 6.9 Experimental and Numerical Energy balances:

Tapered channel 40 x 5~80, 10Hz frequency 50% duty cycle with various power inputs

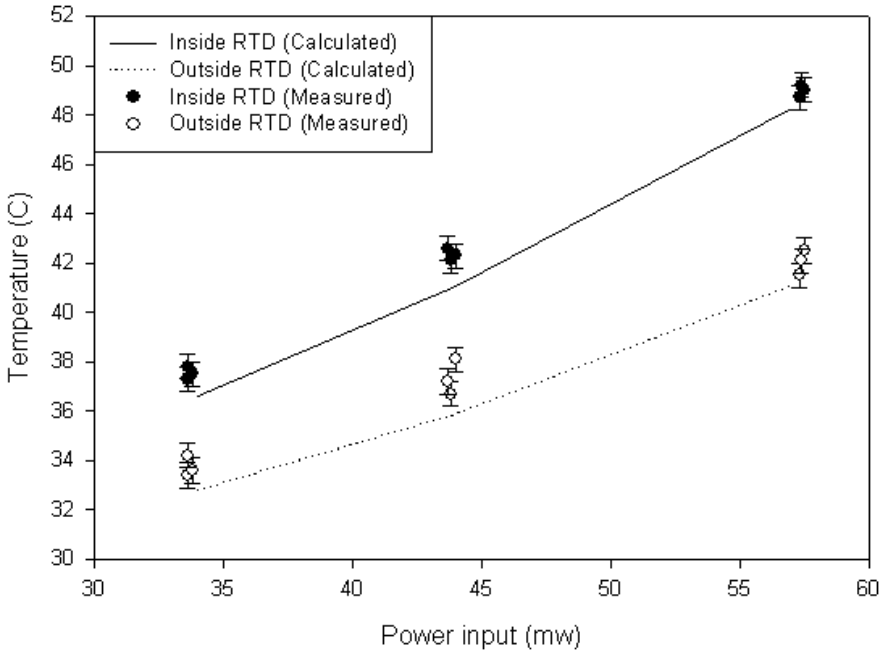
Operation frequencies explored are 10Hz, 20Hz and 50Hz. Duty cycles considered are 50%, 30% and 10%. All experiments and numerical simulations focused on the tapered micro-channel geometry. That is all experiments and numerical simulations were run with micro-channels 40μm deep, with widths that tapered from 80μm at their outer radius to 5μm at their inner radius.

Figure 6.12 shows numerical and experimental results for the tapered micro-channel evaporator operated at a frequency of 10Hz and duty cycle of 50% for thermal power inputs of 34, 44, and 57mW (see Appendix C). Figure 6.12 (a) shows calculated evaporation rates (solid line) versus thermal power input. Figure 6.12 (b) shows calculated temperatures at the inner (solid line) and outer (dotted line) RTD versus thermal power inputs. Both measured evaporation rates (black circles) and measured temperature for the inner (black circles) and outer (white circles) annular RTD show good agreement with the numerical model.

Figure 6.13 shows transient temperature histories for transient evaporation with the conditions of 10Hz 50% duty cycle and 34mW power input. In Figure 6.13, numerical calculations of inside RTD temperature profile is indicated with a solid line, and experimental measurements of inside RTD temperature is indicated with a dotted line. Table 6.9 shows the inner and outer RTD temperature ranged from 34^{o}C to 46^{o}C and 31^{o}C to 38^{o}C. The evaporation rate ranged from 120μg/sec to 152μg/sec. Both experimental and numerical energy balances indicate that approximately 70% to 78% of the power dissipated in the resistance heater is conducted out through the silicon membrane. Only 22% to 30% of the power is carried away by the latent heat of evaporation.



(a) Numerical and Experimental Evaporation



(b) Numerical and Experimental Temperature

Figure 6.14 Numerical and Experimental Evaporation rate and temperature:

Tapered channel 40 x 5~80, 10Hz frequency 30% duty cycle with various power inputs

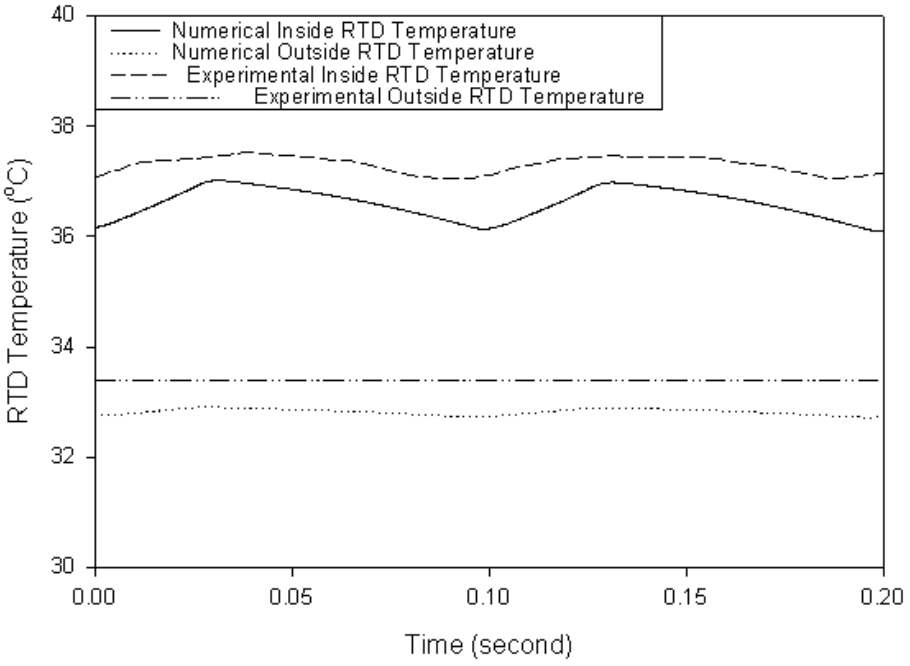


Figure 6.15 Numerical and Experimental Temperature History:

Tapered channel 40 x 5~80, 10Hz frequency 30% duty cycle with 34mW power input

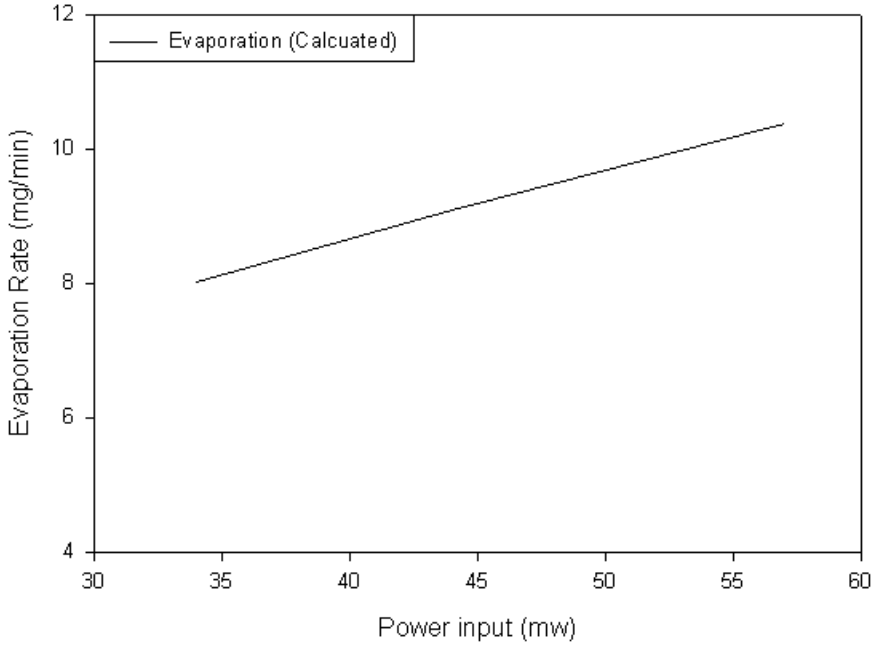
Transient Condition: 10Hz30%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	**Tapered 5~80**	**33.7**	**3.37**	**37.3**	**33.4**	**22.5**	**10.5**	**130.0**	**31.9**
		43.8	**4.38**	**42.1**	**36.7**	**31.0**	**12.0**	**148.3**	**28.2**
		57.3	**5.73**	**48.7**	**41.5**	**42.8**	**13.8**	**166.7**	**24.6**
Numerical	Tapered 5~80	34.0	3.40	36.6	32.8	23.2	10.8	130.0	31.9
		44.0	4.40	41.1	35.9	31.5	12.5	148.3	28.2
		57.0	5.70	48.3	41.1	43.2	13.8	166.7	24.6

Table 6.10 Numerical and Experimental Energy balances:

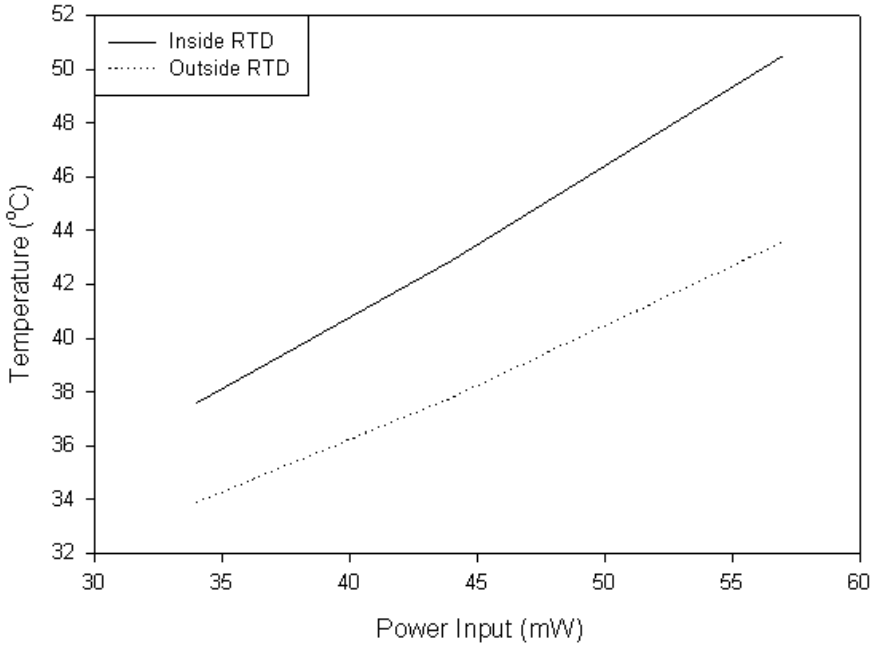
Tapered channel 40 x 5~80, 10Hz frequency 30% duty cycle with various power inputs

Figure 6.14 shows numerical and experimental results for transient operation at a frequency of 10Hz and duty cycle of 30% for thermal power inputs of 34, 44, and 57mW (see Appendix C). Figure 6.14 (a) shows calculated evaporation rates versus thermal power input. Figure 6.14 (b) shows calculated temperatures at the inner and outer RTD versus thermal power input. Measured evaporation rates and measured temperature for the inner and outer annular RTDs show good agreement with the numerical model. Figure 6.15 shows transient temperature histories for transient evaporation with the conditions of 10Hz 30% duty cycle and 34mW power input. In Figure 6.15, numerical calculations of inside RTD temperature profile is indicated with a solid line, and experimental measurements of inside RTD temperature is indicated with a dotted line. Table 6.10 shows the inner and outer RTD temperature ranged from 37^{o}C to 49^{o}C and 33^{o}C to 42^{o}C. Evaporation rates ranged from 130μg/sec to 167μg/sec. Both the experimental and numerical energy balances indicate that approximately 68% to 75% of the power dissipated in the resistance heater is conducted out through the silicon membrane. Only 25% to 32% of the power is carried away by the latent heat of evaporation.

Figure 6.16 shows numerical results for transient operation at a frequency of 10Hz and duty cycle of 10% for thermal power inputs of 34, 44, and 57mW. No experimental results are presented for a frequency of 10Hz and duty cycle of 10%. Figure 6.16 (a) shows calculated evaporation rates versus thermal power input. Figure 6.16 (b) shows calculated temperatures at the inner and outer RTD versus thermal power input. In Figure 6.16, a numerical calculation of inside RTD temperature profile is indicated with a solid line. Table 6.11 shows the inner and outer RTD temperature ranged from 38^{o}C to 51^{o}C and 34^{o}C to 44^{o}C. Evaporation rates ranged from 130μg/sec to 167μg/sec.



(c) Numerical Evaporation



(d) Numerical Temperature

Figure 6.16 Numerical Evaporation rate and temperature:

Tapered channel 40 x 5~80, 10Hz frequency 10% duty cycle with various power inputs

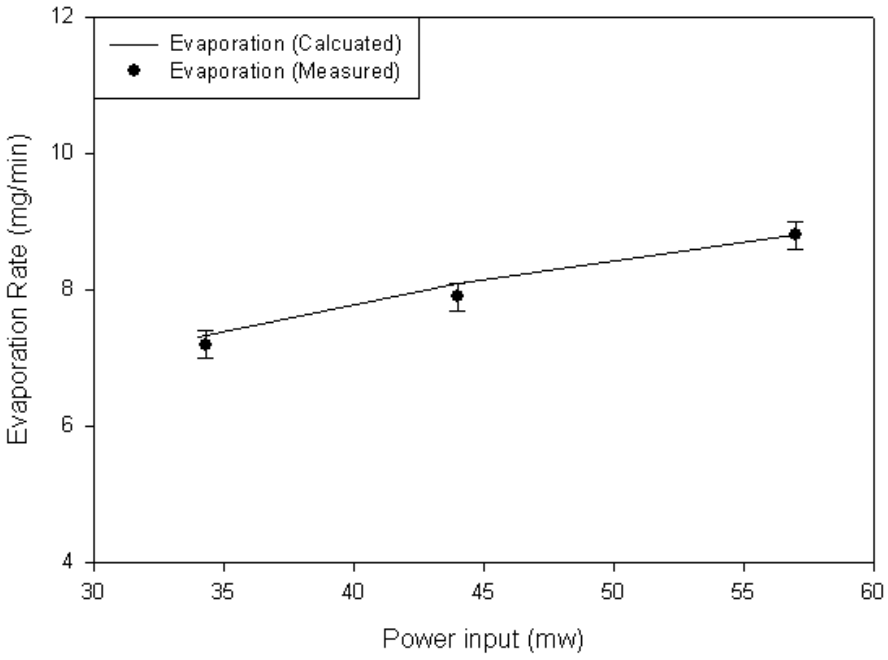
Transient Condition: 10Hz 10%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Wick Efficiency (%)
Numerical	Tapered 5~80	34.0	3.40	34.6	33.9	22.8	11.2	133.7	32.9
		44.0	4.40	42.9	37.8	31.3	12.7	151.7	28.9
		57.0	5.70	50.5	43.6	42.5	14.5	173.2	25.4

Table 6.11 Numerical Energy balances:

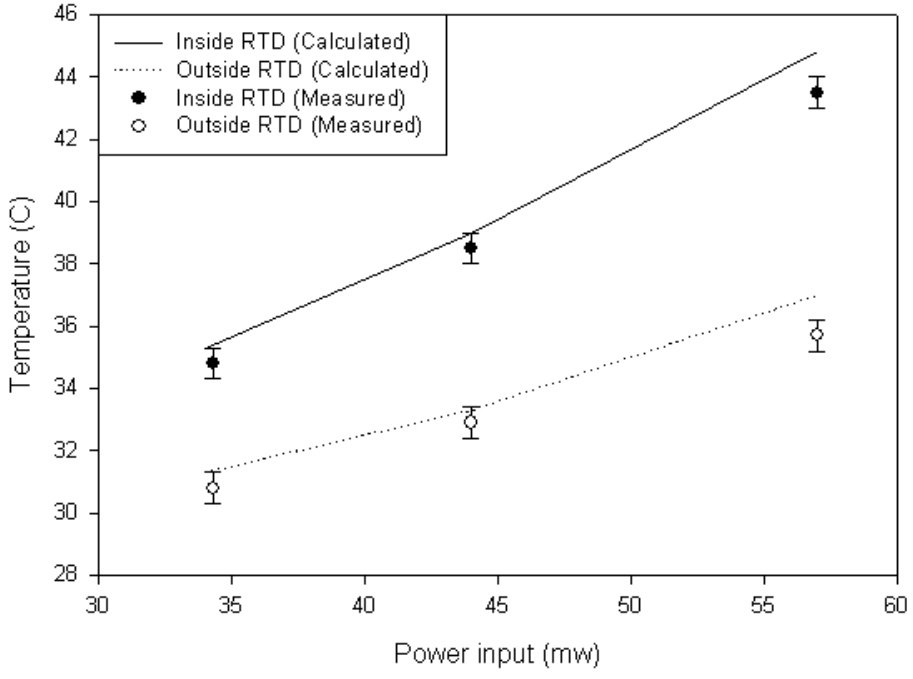
Tapered channel 40 x 5~80, 10Hz frequency 10% duty cycle with various power inputs

The numerical energy balances indicate that approximately 67% to 74% of the power dissipated in the resistance heater is conducted out through the silicon membrane. Only 26% to 33% of the power is carried away by the latent heat of evaporation.

Figure 6.17 shows numerical and experimental results for the tapered micro-channel evaporator operated at a frequency of 20Hz and a duty cycle of 50% for thermal power inputs of 34, 44, and 57mW (see Appendix C). Again, the numerical calculations are indicated with lines, and the experimental results are indicated with symbols. Figure 6.17 (a) shows calculated evaporation rates versus thermal power input. Figure 6.17 (b) shows calculated temperatures at the inner and outer RTDs versus thermal power input. Measured evaporation rates and measured temperatures for the inner and outer annular RTD show good agreement with the numerical model. Figure 6.18 shows transient temperature histories for transient evaporation with the conditions of 20Hz 50% duty cycle and 34mW power input. In Figure 6.18, numerical calculations of inside RTD temperature profile is indicated with a solid line, and experimental measurements of inside RTD temperature is indicated with a dotted line.



(a) Numerical and Experimental Evaporation



(b) Numerical and Experimental Temperature

Figure 6.17 Numerical and Experimental Evaporation rate and Temperature:

Tapered channel 40 x 5~80, 20Hz frequency 50% duty cycle with various power inputs

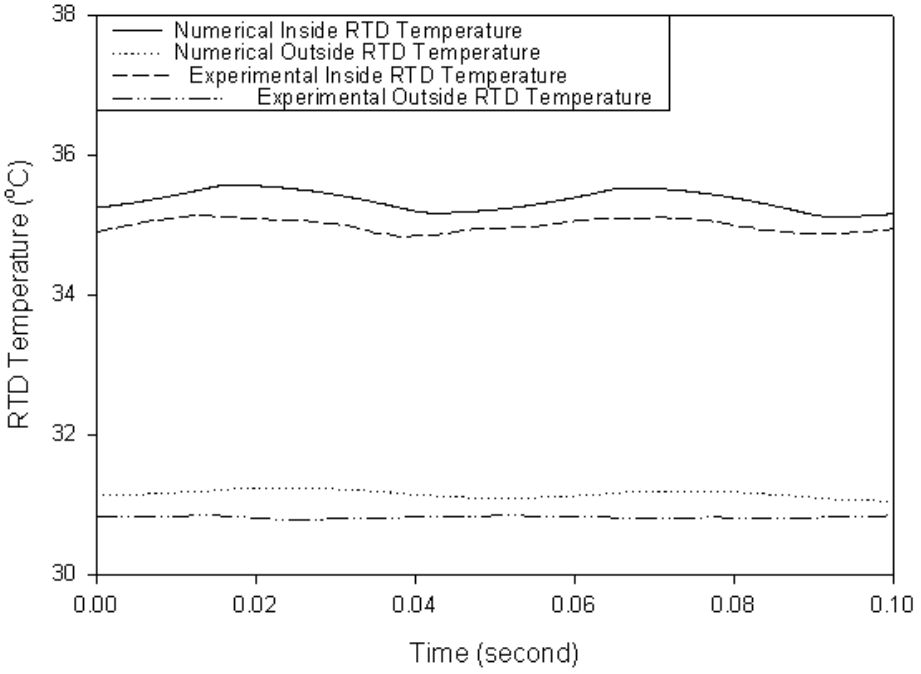


Figure 6.18 Numerical and Experimental Temperature History:

Tapered channel 40 x 5~80, 20Hz frequency 50% duty cycle with 34mW power input

Transient Condition: 20Hz50%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Efficiency (%)
Experimental	**Tapered 5~80**	**34.3**	**1.72**	**34.8**	**30.8**	**22.7**	**10.0**	**120.0**	**29.2**
		44.1	**2.21**	**38.5**	**32.9**	**30.9**	**11.0**	**131.7**	**24.8**
		57.4	**2.87**	**43.5**	**35.7**	**43.2**	**12.3**	**146.7**	**21.3**
Numerical	Tapered 5~80	34.0	1.70	35.3	31.2	23.8	10.2	121.7	29.9
		44.0	2.20	39.0	33.3	32.7	11.3	135.0	25.6
		57.0	2.85	44.8	37.0	44.7	12.3	146.7	21.5

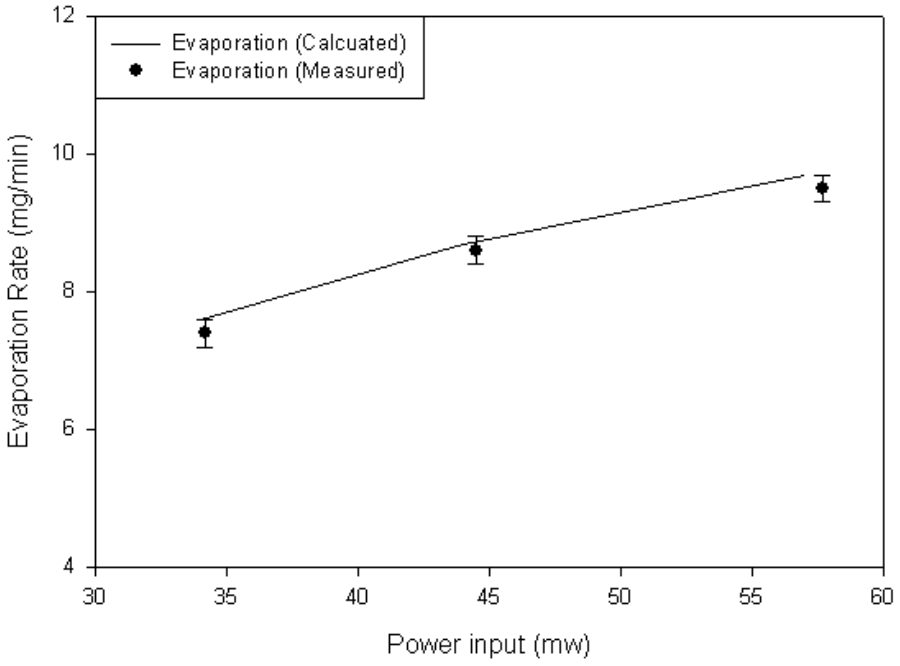
Table 6.12 Numerical and Experimental Energy balances:

Tapered channel 40 x 5~80, 20Hz frequency 50% duty cycle with various power inputs

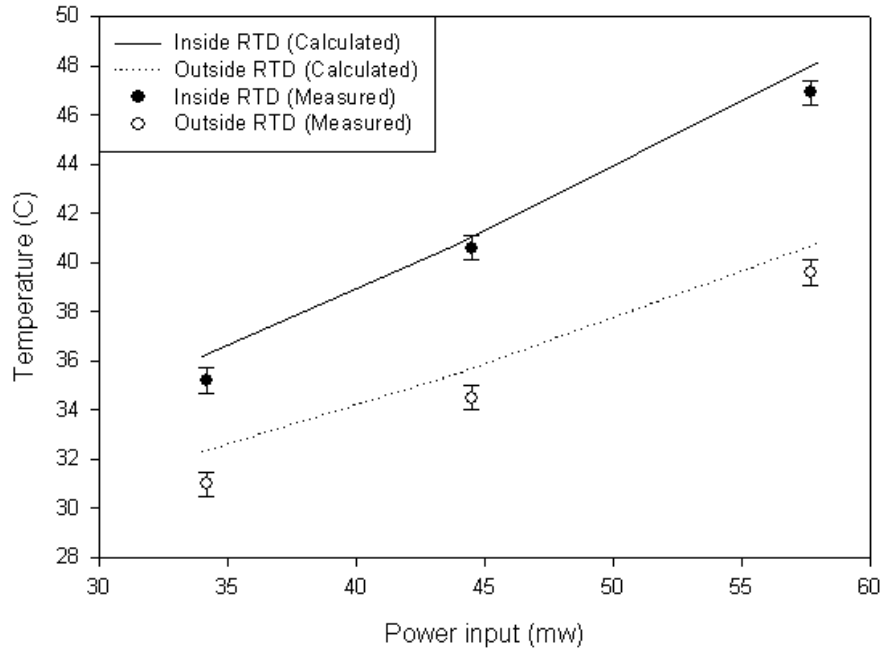
Table 6.12 shows the inner and outer RTD temperature ranged from 35°C to 45°C and 31°C to 40°C. Evaporation rates ranged from 122μg/sec to 147μg/sec. Both experimental and numerical energy balance indicate that approximately 70% to 78% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while 22% to 30% of power is carried away by the latent heat of evaporation.

Figure 6.19 shows numerical and experimental results for transient operation at a frequency of 20Hz and a duty cycle of 30% for thermal power inputs of 34, 44, and 57mW (see Appendix C). Figure 6.19 (a) shows calculated evaporation rates versus thermal power input. Figure 6.19 (b) shows calculated temperatures at the inner and outer RTD versus thermal power input. Measured evaporation rates and measured temperature for the inner and outer annular RTDs show good agreement with the numerical model. Figure 6.20 shows transient temperature histories for transient evaporation with the conditions of 20Hz 30% duty cycle and 34mW power input. In Figure 6.20, numerical calculations of inside RTD temperature profile is indicated with a solid line, and experimental measurements of inside RTD temperature is indicated with a dotted line. Table 6.13 shows the inner and outer RTDs temperature ranged from 35°C to 48°C and 32°C to 41°C. The evaporation rate ranged from 127μg/sec to 168μg/sec. Both experimental and numerical energy balance indicates that approximately 69% to 76% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while 24% to 31% of power is carried away by the latent heat of evaporation.

Figure 6.21 shows numerical results for transient operation at a frequency of 20Hz and duty cycle of 10% for thermal power inputs of 34, 44, and 57mW. No experimental results are presented for a frequency of 20Hz and duty cycle of 10%. Figure 6.21 (a) shows calculated evaporation rates versus thermal power input.



(a) Numerical and Experimental Evaporation



(b) Numerical and Experimental Temperature

Figure 6.19 Numerical and Experimental Evaporation rate and Temperature:

Tapered channel 40 x 5~80, 20Hz frequency 30% duty cycle with various power inputs

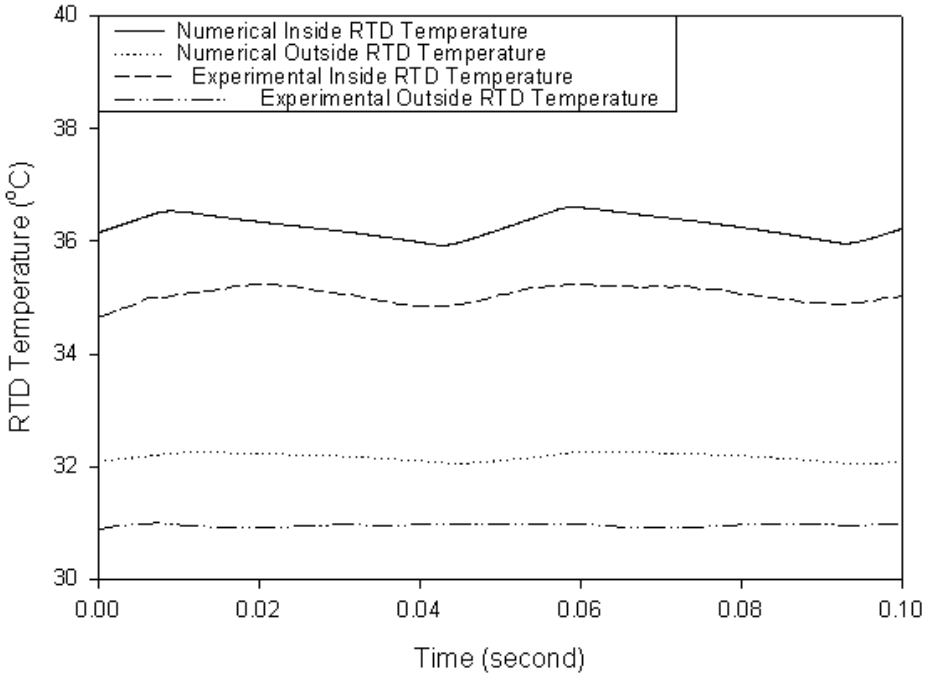


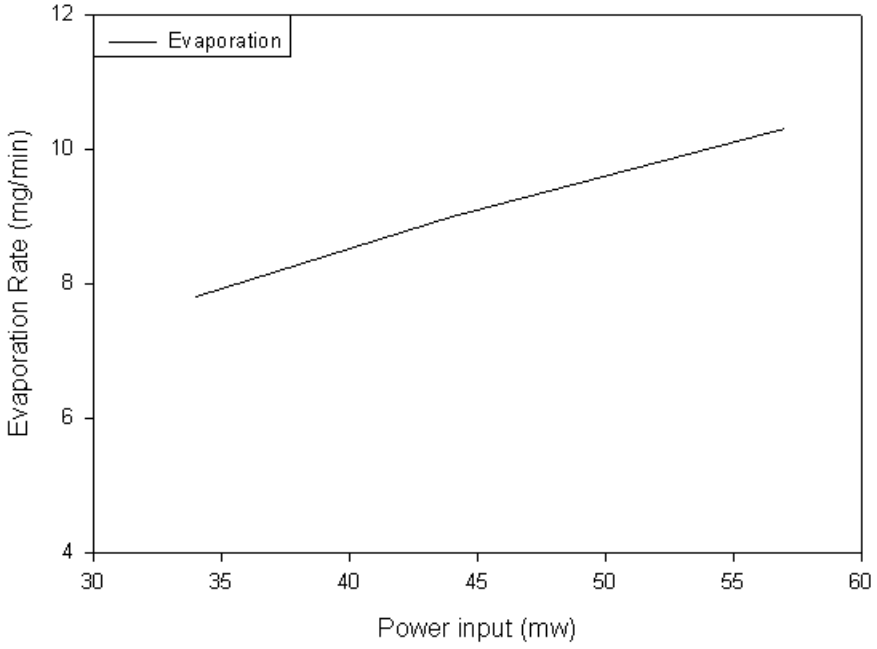
Figure 6.20 Numerical and Experimental Temperature History:

Tapered channel 40 x 5~80, 20Hz frequency 30% duty cycle with 34mW power input

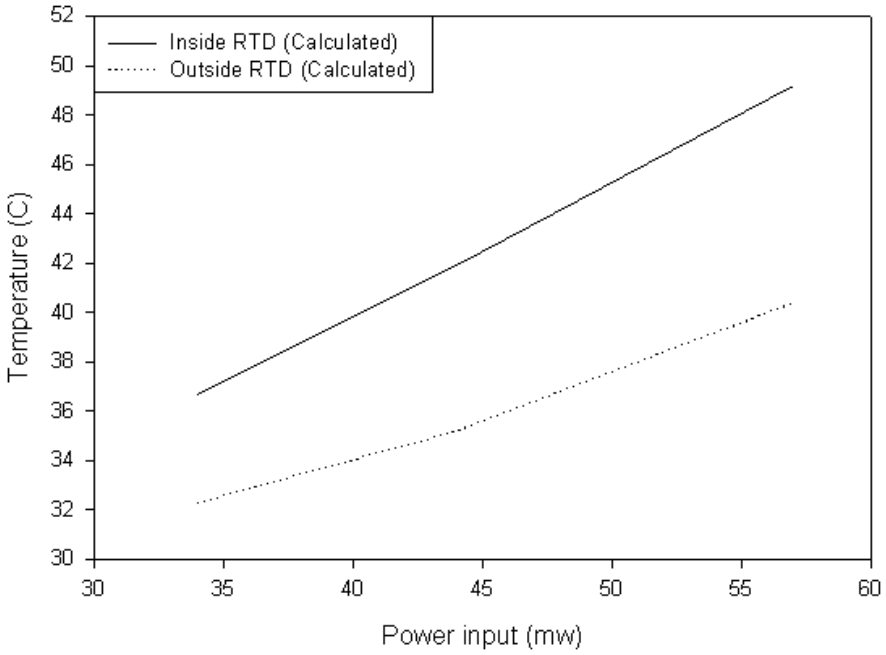
Transient Condition: 20Hz50%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Wick Efficiency (%)
Experimental	**Tapered 5~80**	**34.2**	**1.71**	**35.2**	**31.0**	**23.8**	**10.5**	**123.3**	**30.7**
		44.5	**2.23**	**40.6**	**34.5**	**34.2**	**10.8**	**143.3**	**26.6**
		57.7	**2.89**	**46.9**	**39.6**	**40.8**	**13.3**	**158.3**	**23.4**
Numerical	Tapered 5~80	34.0	1.70	36.2	32.3	23.3	10.7	126.7	31.3
		44.0	2.20	40.5	35.2	31.8	12.2	145.0	27.6
		58.0	2.90	48.4	41.0	44.2	13.7	164.5	23.6

Table 6.13 Numerical and Experimental Energy balances:

Tapered channel 40 x 5~80, 20Hz frequency 30% duty cycle with various power inputs



(a) Numerical Evaporation



(b) Numerical Temperature

Figure 6.21 Numerical Evaporation rate and Temperature:

Tapered channel 40 x 5~80, 20Hz frequency 10% duty cycle with various power inputs

Transient Condition: 10Hz 10%	Channel Type (μm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (μg/sec)	Wick Efficiency (%)
Numerical	Tapered 5~80	34.0	3.40	36.7	32.3	23.1	10.9	130.2	32.2
		44.0	4.40	41.9	35.2	31.5	12.5	149.3	28.4
		57.0	5.70	49.2	40.4	42.7	14.3	170.8	25.0

Table 6.14 Numerical Energy balances:

Tapered channel 40 x 5~80, 20Hz frequency 10% duty cycle with various power inputs

Figure 6.21 (b) shows calculated temperatures at the inner and outer RTD versus thermal power input. In Figure 6.21, a numerical calculation of inside RTD temperature profile is indicated with a solid line. Table 6.14 shows the inner and outer RTD temperature ranged from 37°C to 50°C and 33°C to 41°C. Evaporation rates ranged from 130μg/sec to 167μg/sec. The numerical energy balances indicate that approximately 67% to 75% of the power dissipated in the resistance heater is conducted out through the silicon membrane. Only 25% to 33% of the power is carried away by the latent heat of evaporation.

In all cases the transient results show that the efficiencies of the micro-channel evaporators decreased as thermal power input to the micro-channel evaporator increased, just as for the steady state results. The transient results also show that efficiencies of the evaporator increased as duty cycle decreased and as operating frequency increased.

6.7 TRANSIENT TEST RESULTS (Tapered Micro-channel: SiNx)

A second set of tapered micro-channels was fabricated on 300nm thick Silicon Nitride membranes. Silicon Nitride membranes were used for two reasons. First, the thermal conductivity of Silicon Nitride is much lower than Silicon: 15W/mK vs.148 W/mK. Second Silicon Nitride membranes can be fabricated to be much thinner than silicon membranes: 300nm vs. 2μm. The tapered micro-channels were identical to the tapered micro-channels fabricated on Silicon membranes as shown in Figure 6.4. As before the tapered micro-channels were 40μm deep with channel widths that tapered from 80μm at their outer radius to 5μm at their inner radius. Likewise channel wall thickness tapered from 80μm down to 5μm.

Numerical results presented in this chapter are for the silicon nitride membrane with wicking wall dimensions of 40μm deep and tapered channel 5μm to 80μm wide. Figure 6.22 shows transient temperature histories for the transient operation at a frequency 10Hz and a duty cycle 50% for power input of 11mW power input with the fixed boundary condition of 22.8^{o}C. In Figure 6.22, numerical calculations of inside RTD temperature profile is indicated with a solid line, and numerical calculations of outside RTD temperature is indicated with a dotted line. Unlike the transient simulation with silicon membrane, no waveform was observed from the transient simulation with silicon nitride membrane.

Table 6.15 shows the inner and outer RTD temperature was 40.6^{o}C and 25.2^{o}C. The evaporation rate was 117μg/sec. Numerical energy balances indicate that approximately 12% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while 88% the power is carried away by the latent heat of evaporation.

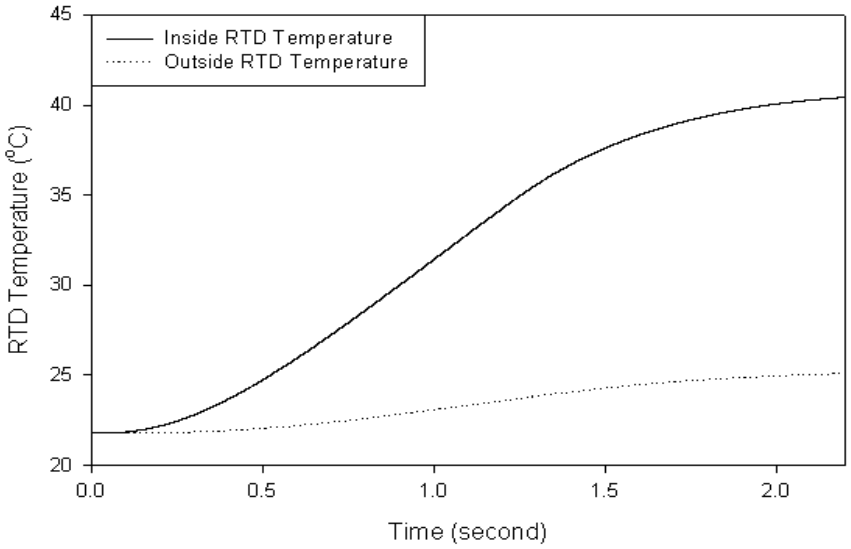


Figure 6.22 Numerical Temperature History:

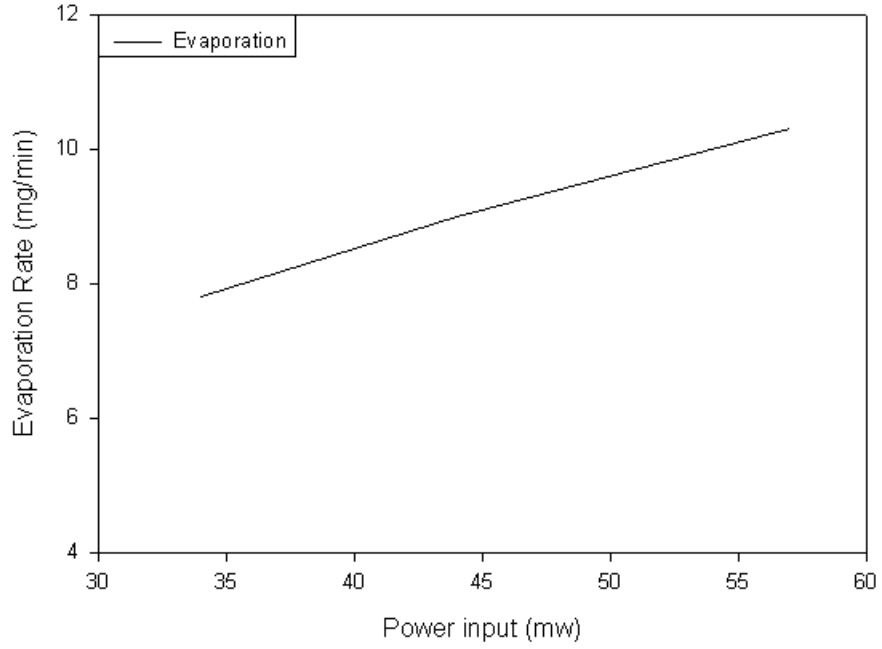
Tapered channel 40 x 5~80, 10Hz frequency 50% duty cycle with 11mW power input

Transient Condition: 10Hz 50 %	Channel Type (µm)	Power into Heater (mW)	Power into Heater per pulse (mJ)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Wick Efficiency (%)
Numerical	Tapered 5~80	11	1.1	40.6	25.2	1.3	9.7	116.7	88

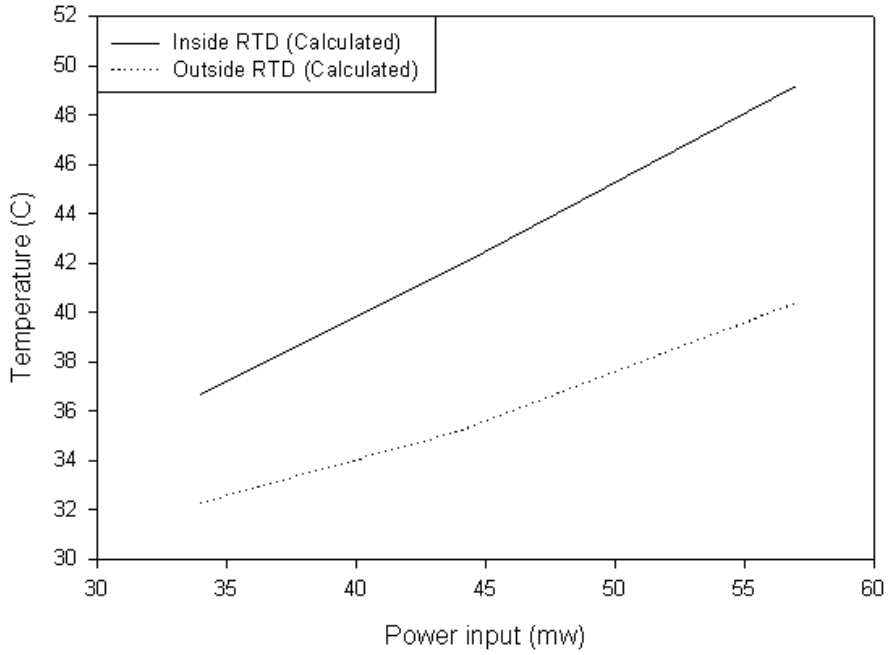
Table 6.15 Numerical Energy balances:

Tapered channel 40 x 5~80, 10Hz frequency 50% duty cycle with 11mW power inputs

6.8 EFFECT OF HEAT INPUT AREA



(a) Numerical Evaporation



(b) Numerical Temperature

Figure 6.23 Numerical Evaporation rate and Temperature:

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	34	32.1	28.6	21.5	12.5	148.3	37.0
			44	35.8	30.4	29.8	14.2	170.0	32.3
			57	40.8	33.4	41.6	15.4	186.7	27.0

Table 6.16 Numerical Energy balances

Steady state evaporation from open top tapered micro-channels is simulated for thermal power input of 34, 44, and 58mW. The thermal input is applied to entire bottom surface of micro-channel evaporator. Dimensions of tapered micro-channels are 40µm deep and their width is 80µm at the outer radius of the micro-channel. Each micro-channel width tapers down to 5µm at the inner radius of the micro-channel. The walls dividing micro-channels also are 80µm wide at their outer radius and taper down to a width of 5µm at their inner radius. The numerical results are indicated with lines.

Figure 6.23 shows numerical results for tapered micro-channels with thermal power inputs of. 34, 44, and 57mW. Figure 6.23 (a) shows evaporation rates. Figure 6.23 (b) shows temperatures at the inner (solid line) and outer (dotted line) annular RTDs versus thermal power inputs. Table 6.16 shows that for the tapered micro-channel (40µm deep with 5µm to 80µm tapered width) the inner and outer RTD temperature ranged from 33^{o}C to 44^{o}C and 29^{o}C to 35^{o}C. The evaporation rate ranged from 103.3µg/sec to 135.6µg/sec. Both experimental and numerical energy balances indicates that approximately 70% to 80% of the power dissipated in the resistance heater is conducted out through the silicon membrane, while 20% to 30% of power is carried away by the latent heat of evaporation.

CHAPTER 7

CONCLUSIONS

7.1 CONCLUSIONS

The first goal of the present study is to find the most efficient wicking structure by determining the overall efficiencies of the micro-channel evaporator with varying wick dimensions. The second goal of this study is to build an effective design tool for the development of efficient micro-channel evaporator.

Channels with dimensions from ten to one hundred microns are used in this study. To find the overall most effective wick geometry and dimensions, the steady state evaporation test and transient evaporation test are performed with silicon-based membranes and silicon-nitride-based membranes. Evaporative heat transfer rates and evaporation rates from various SU8 micro-channels have been experimentally measured. A numerical model of sensible heat transfer and evaporation from the micro-channel evaporator has been developed.

Results from the experimental measurements and numerical calculations shows no significant difference between the different dimensions and designs of the micro-wicking channels for the steady state evaporation condition with silicon-based membranes. For constant cross section rectangular channels, the liquid thicknesses of micro-channels are estimated by comparison of the experimental measurements and numerical calculations from various liquid thicknesses. Such a comparison of the numerical calculations and experimental measurements yields consistent estimates of liquid thickness for the micro-channels within about ±5μm. For tapered (wicking wall dimensions of 40μm deep and tapered channel 5μm to 80μm wide) channels, the force balance equations are introduced

and included in the numerical model to eliminate estimating the liquid thickness. The updated numerical model and experimental results are compared to validate the model. A comparison of the numerical calculations and experimental measurements shows good agreement. Silicon based membrane steady state evaporation shows that both the experimental and numerical energy balances indicates that approximately 70%~80% of the power dissipated in the resistance heater is by conduction through the membrane, while 20%~30% of power is carried away by the latent heat of evaporation. Results also show wick efficiencies are decreased as power inputs are increased for all silicon-based membrane steady state evaporation test cases.

Since much work is completed for the steady state evaporation test, the transient tests are performed with silicon-based membranes to see how the performance of the evaporator changes at transient operations. A number of resonant operation conditions and a number of duty cycles are used to perform both experimental tests and numerical calculations. A comparison of the numerical calculations and experimental measurements shows good agreement. The results are compared with the steady state evaporation results, and the results from transient operations show no significant improvement in the wick efficiencies. Again, results show wick efficiencies are decreased as power inputs are increased for all silicon-based membrane steady state evaporation cases. In the same resonant operation condition, the results show that wick efficiencies are increased as duty cycles are decreased.

Silicon-nitride based membranes are used in the evaporation test since the silicon-based membrane evaporation test does not show effective wick efficiencies. Changing the membrane material from the high conductivity material to the low conductivity material

shows significant improvement in the wick efficiencies because the low conductivity material significantly decreases the heat loss from the radial conduction heat transfer. The experimental measurements and the numerical predictions presented in this dissertation shows that the latent heat by evaporation of the heat added to the concentric heater at the center of the membrane ranged from roughly 18%-26% for silicon based membranes and 82%-86% for silicon nitride based membranes.

REFERENCES

[1] P., Bergstrom, j. Ji, Y., Liu, M., Kaviany, K. Wise, Thermally driven phase-change microactuation. Journal of Microelectromechanical System, 4 (1995) no.1, pp.10-17

[2] C., Rich, K., Wise, A high flow thermopneumatic microvalve with improved efficiency and integrated state sensing. Journal of Micro-electromechanical System, 12 (2003) no.2, pp.201-208

[3] Y.P. Peles, L.P. Yarin, G. Hetsroni, Thermo-hydrodynamic characteristics of two-phase flow in a heated capillary. International journal of Multiphase Flow, 26 (2000) 1063-1093

[4] C.J. Kroeker, H.M. Soliman, S.J. Ormiston, Three-dimensional thermal analysis of heat sinks with circular cooling micro-channels. International Journal of Heat and Mass Transfer, 47 (2004) pp.4733-4744

[5] C.B. Sobhan, S.V. Garimaella, Micro-scale Thermo-physical Engineering. 5 (2001) pp.293-311

[6] Y. Zhuang, C.F. Ma, and M. Qin, Experimental study on local heat transfer with liquid impingement flow in two-dimensional micro-channels. International Journal of Heat and Mass Transfer, 40 (1997) no.18, pp.4055-4059

[7] H.R. Chen, C. Gau, B.T. Dai, M.S. Tsai, A monolithic fabrication process for a micro-flow heat tranfer channel suspended over an air layer with arrays of micro-sensors and heaters. Sensors and Actuators A, 108 (2003) pp.81-85

[8] J. Lee, I. Mudawar, Two-phase flow in high heat flux micro-channel heat sink for refrigeration cooling application: Part II-heat transfer characteristics. International Journal of Heat and Mass Transfer, 48 (2005) pp.941-955

[9] S. Whalen (2004), Cycle work from a MEMS Heat Engine and Characterization of the Liquid-Vapor Phase Change Actuation Mechanism. Doctorial Dissertation. Washington State University. Pullman, WA.

[10] S. Whalen, M. Thompson, D. Bahr, C. Richards and R. Richards, Design, fabrication and testing of the P^3 micro heat engine. Sensors and Actuators, A: Physical, Vol. 104 (2003) No. 3 pp. 290-298

[11] I. Tiselj, G. Hetsroni, B. Mavko, A. Mosyak, E. Pogrebnyak, Z. Segal, Effect of axial conduction on the heat transfer of micro-channels. International of Heat and Mass Transfer, 47 (2004) 2551-2565

[12] W. Owhaib, B. Palm, Experimental investigation of single-phase convective heat transfer in circular micro-channels. Experimental Thermal and Fluid Science, 28 (2004) pp.105-110

[13] C.W. Liu, C. Gau, B.T. Dai, Design and fabrication of a micro flow heated channel with measurements of the inside micro-scale flow and heat transfer process. Biosensors and Bioelectronics, 20 (2004) pp.91-101

[14] Z.Q. Chen, P. Cheng, and T.S. Zhao, An experimental analysis of two-phase flow and boiling heat transfer in bi-dispersed porous channels. Int. Comm. Heat Mass Transfer, 27 (2000) Vol. 3 pp. 293-302

[15] J.H. Daniel, B. Kursor, R.B. Apte, R.A. Street, A. Goredema, J. McCallum, D.C Boils, P.M. Kazmaier, Large area MEMS fabrication with thick SU8 photoresist applied to an X-ray image sensor. Micromachining Fabrication Process Technology VI, 4174 (2000) pp.40-48

[16] J.A. van Kan, I. Rijta, K. Ansari, AA. Bettiol, F. Watt, Nickel and copper electroplating of proton beam micromachined SU-8 resist. Microsystems Technology, 8 (2002) pp.383-386

[17] D.W. Johnson, A. Jeffries, D.W. Minsek, R.J. Hurditch, Improving the Process Capability of SU8, Part II. Internet Resource:
http://www.microchem.com/resources/su8_process_capability_paper_1.pdf

[18] R. Feng, R.J. Farris, Influence of processing conditions on the thermal and mechanical properties of SU8 negative photoresist coatings. Journal of Micromechanics and Microengineering, 13 (2003) pp.80-88

[19] NanoTM SU8. Micro-Chem. Internet Resource:
http://www.microchem.com/products/pdf/SU8_2-25.pdf

[20] T.M. Harms, M. Kazmierczak, F.M. Gerner, A. Holke, H.T. Henderson, J. Pilchowski, K. Baker, Experimental investigation of heat transfer and pressure drop through deep micro-channels in a (110) silicon substrate. Proc. ASME Heat Transfer Division, vol. 1, ASME HTD, vol. 351 (1997) pp. 347-357

[21] S.G. Kandlikar, S. Garimella, D. Li, S. Colin, M.R. King, Heat transfer and fluid flow in mini-channels and micro-channels. Elsevier Ltd. First edition 2006

[22] J. Li, G.P. Peterson, P. Cheng, Three-dimensional analysis of heat transfer in a micro-heat sink with single-phase flow. International Journal of Heat and Mass Transfer, 47 (2004) 4215-4231

[23] D.K. Bailey, T..A. Ameel, R.O. Warrington, T.I. Savoie, Single-phase forced convection heat transfer in micro-geometries: a review. ASME IECEC paper (1995) ES-396 pp. 301-310

[24] L. Zhang, J.M. Koo, L. Jiang, M. Asheghi, K.E. Goodson, J.G. Santiago, and T.W. Kenny, Measurements and Modeling of two-phase flow in micro-channels with nearly constant heat flux boundary conditions. Journal of Electromechanical Systems, 11 (2002) 1 pp.12-19

[25] M.B. Browers, I. Mudawar, High flux boiling in low flow rate, low pressure drop mini-channel and micro-channel heat sink. International Journal of Heat and Mass Transfer 37 (1994) 321-332

[26] B.X. Wang and X.F. Peng, Experimental investigation on liquid forced convection heat transfer through micro-channels. International Journal of Heat and Mass Transfer, vol. 37 (1994) pp. 73-82

[27] X.F. Peng, G.P. Peterson, Convective heat transfer and flow friction for water flow in micro-channel structures. International Journal of Heat and Mass Transfer, vol. 39 no. 12 (1996) pp. 2599-2608

[28] M.B. Browers, I. Mudawar, Two-phase electronic cooling using mini-channel and micro-channel heat sink. Part II. Flow rate and pressure drop constraints. Journal of Electron packaging, ASME 116 (1994) 298-305

[29] M.M. Rahman, Measurements of heat transfer in micro-channel heat sinks. International communication in Heat and Mass Transfer, 27 (2000) 495-506

[30] K.K. Ambatipudi, M.M. Rahman, Analysis of conjugate heat transfer in micro-channel heat sinks/ Numerical Heat Transfer. Part A. Applications 37 (2000) 711-731

[31] C.S. Landerman, Micro-channel flow boiling mechanisms leading to Burnout. Journal of Heat Transfer Electronic System ASME HTD-292 (1994) 124-136

[32] R.H. Nilson, S.W. Tchikanda, S.K. Griffinths, M.J. Martinez, Axially tapered micro-channels of high aspect ratio for evaporative cooling devices. International Journal of Heat and Mass Tranfer 126 (2004) 453-462

[33] Gad-el-Hak, M., The fluid mechanics of micro-devices. Journal of Fluid Engineering, 121 (1999) 7-33

[34] P. Gao, S. Le Person, M. Favre-Marinet, Scale effects on hydrodynamics and heat transfer in two-dimensional mini and micro-channels. International Journal of Thermal Sciences, 41 (2002) 1018-1027

[35] L.T. Hwang, I. Turlik, A. Reisman, A thermal module design for advancing packaging. J. Electron. Mater. 16 (5) (1987) 347-355

[36] J.H. Ryu, D.H. Choi, S.J.Kim, Numerical optimization of the thermal performance of a micro-channel heat sink. International Journal of heat and Mass Transfer 45 (2002) 2823-2827

[37] R.W. Knight, J.S. Goodling, D.J. Hall, Optimal thermal design of forced convection heat sinks-Analytical. ASME Journal of Electronic Packaging. 113 (1991) 313-321

[38] J.C.Y. Koh, R. Colony, Heat transfer of micro-structure for integrated circuits, Int. Commun. Heat Mass Transfer 13 (1986) 89-98

[39] H.S. Kou, J.J. Lee, C.W. Chen, Optimum thermal performance of micro-channel heat sink by adjusting channel width and height. International Communications in Heat and Mass Transfer 35 (2008) 577-582

[40] Z. Wen, K.F. Choo, The optimum thermal design of micro-channel heat sinks. IEEE/CPMT Electronic Packaging Technology Conference (1997) 123-129

[41] J. Li, G.P. Peterson, Three-dimensional numerical optimization of silicon-based high performance parallel micro-channel heat sink with liquid flow. International Journal of Heat and Mass Transfer 50 (2007) 2895-2904

[42] R.H. Nilson, S.W. Tchikanda, S.K. Griffinths, M.J. Martinez, Steady evaporation flow in rectangular micro-channels. International Journal of heat and Mass Transfer 49 (2006) 1603-1618

[43] S.W. Tchikanda, R.H. Nilson, S.K. Griffinths, Modeling of pressure and shear-driven flows in open rectangular micro-channels. International Journal of Heat and Mass Transfer 47 (2004) 527-538

[44] C.Suman, S.K. Som, Heat Transfer in an evaporating thin liquid film moving slowly along the walls of an inclined micro-channel. International Journal of Heat and Mass Transfer 48 (2005) 2801-2805

[45] H. Wang, S. V. Garimella, J.Y. Murthy, Characteristic of an evaporation thin film in a micro-channel. International Journal of Heat and Mass Transfer 50 (2007) 3933-3942

[46] S.J. Kim, J.K. Seo, K.H. Do, Analytical and experimental investigation on the operational characteristics and the thermal optimization of a miniature heat pipe with a grooved wick structure. International Journal of Heat and Mass Transfer 46 (2003) 2051-2063

[47] K.H. Do, S.J. Kim, S.V. Garimella, A mathematical model for analyzing the thermal characteristics of a flat micro heat pipe with a grooved wick, International Journal of Heat and Mass Transfer (2008) Article in press

[48] R. Hopkins, A. Faghri, D. Khrustalev, Flat miniature heat pipes with micro capillary grooves. ASME J. Heat Transfer 121 (1999) 102-109

[49] K. Park, K. Lee, Flow and heat transfer characteristics of the evaporating extended meniscus in a micro-capillary channel. International Journal of Heat and Mass Transfer 46 (2003) 4587-4594

[50] K. Park, K.J. Noh, K.S. Lee, Transport phenomena in the thin-film region of a micro-channel. International Journal of Heat and Mass Transfer 46 (2003) 2381-2388

[51] A.J. Jiao, H.B. Ma, J.K. Critser, Evaporation heat transfer characteristics of a grooved heat pipe with micro-trapezoidal grooves. International Journal of Heat and Mass Transfer 50 (2007) 2905-2911

[52] Y.P. Peles, L.P. Yarin, G. Hetsroni, Thermo-hydrodynamic characteristics of two-phase flow in a heated capillary. International Journal of Multiphase Flow 26 (2000) 1063-1093

[53] S.S. Shevade, M.M. Rahman, Heat transfer in rectangular micro-channels during volumetric heating of the substrate. International Communications in Heat and Mass Transfer 34 (2007) 661-672

[54] F. Incropera, D. Dewitt, Fundamentals of heat and mass transfer. John Wiley & Sons, Inc., 1990

[55] I. Catton, G.R. Stores, A Semi-analytical model to predict the capillary limit of heated inclined triangular capillary grooves. Journal of Heat Transfer ASME 124 (2002) 162-168

[56] T.S. Sheu, P.P. Ding, I.M. Lo, P.H. Chen, Effect of surface characteristics on capillary flow in triangular micro-grooves. Experimental Thermal and Fluid Science 22 (2000) 103-110

[57] G.P. Peterson, J.M. Ha, Capillary performance of evaporating flow in micro grooves: an approximate analytical approach and experimental investigation. ASME Journal of Heat Transfer 120 (1998) 743-751

[58]] W.R. Jong, T.H. Kuo, S.W. Ho, H.H. Chiu, S.H. Peng, Flows in rectangular micro-channels driven by capillary force and gravity. International Communications in Heat and Mass Transfer 34 (2007) 186-196

[59] I. Naoki, H. Kazuo, M. Ryutaro, Interface motion of capillary-driven flow in rectangular Micro-channel. Journal of Colloid and Interface Science 280 (2004) 155-164

[60] D. Yan, C. Yang, N.T. Nguyen, X. Huang, Electro-kinetic flow in micro-channels with finite reservoir size effects. Journal of Physics 34 (2006) 385-392

[61] K.H. Chang, D. Pan, Two-phase flow instability for boiling in a micro-channel heat sink. International Journal of Heat and Mass Transfer 50 (2007) 2078-2088

[62] R.H. Liu, M.A. Stremler, K.V. Sharp, M.G. Olsen, J.G. Santiago, R.J. Adrian, H. Aref, D.J. Beebe. Journal of Microelectromechanicl Systems 9 (2000) no. 2 pp. 190-197

[63] S.S. Hsieh, C.Y. Lin, C.F. Huang, H.H. Tsai, Liquid flow in a micro-channel. Journal of Micro-mechanics and micro-engineering, 14 (2004) 436-445

[64] S. Hardt, B. Schilder, D. Tiemann, G. Kolb, V. Hessel, P. Stephan, Analysis of flow patterns emerging during evaporation in parallel micro-channels. International Journal of Heat and Mass Transfer 50 (2007) 226-239

[65] T.A. Quy, D. Carpenter, C.D. Richards, D.F. Bahr, R.F. Richards, Evaporative heat transfer from ten-micron micro-channels. ASME IMECE (2005)

[66] H.K. Lee, T.A. Quy, C.D. Richards, D.F. Bahr, R.F. Richards, Experimental and numerical study of evaporative heat transfer from ten-micro-micro-channels. ASME IMECE (2006)

[67] H.K. Lee, C.D. Richards, R.F. Richards, Experimental and numerical study of evaporating flow in micro-channels. ASME IMECE (2007)

[68] Tai-Ran Hsu, MEMS & Micro-systems design and manufacture, Tata McGraq-Hill Edition 2002, second reprint 2003

[69] Kaveh Azar, Thermal Measurements in Electronics Cooling, CRC Press, pp62-63, 1997

[70] A. Wheeler, A. Ganji, Introduction to engineering experimentation, Prentice Hall, Inc., NJ (1996) pp.160-161

[71] R.W. Schrage, A theoretical study of inter-phase mass transfer, Columbia University Press 1953

[72] H. Ma, G. Peterson, The minimum meniscus radius and capillary heat transport limit in micro heat pipes, Journal of Heat Transfer, vol. 120 (1998) pp.227-232

[73] R.T. Rao, R. J. Eugene, K.G. Alan, Microelectronic packaging handbook, 1997

[74] T.A. Quy (2006), Characterization of micro-capillary wicking evaporators. Master Dissertation. Washington State University. Pullman, WA.

APPENDIX A

1. Cylindrical conduction equation

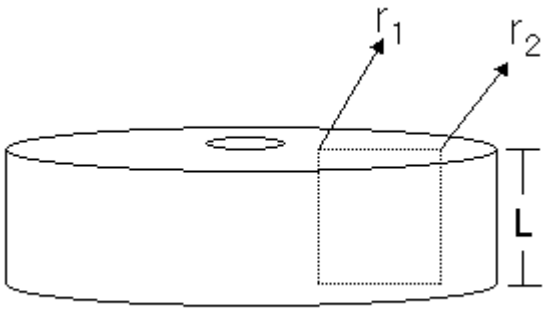


Figure A1 Cylindrical coordinates

The heat transfer, q, by Fourier's law is set at equation A1-1.

$$q = -kA\frac{dT}{dr} \tag{A1-1}$$

where

$$A = 2\pi rL \tag{A1-2}$$

so

$$q = -k(2\pi r)\frac{dT}{dr} \tag{A1-3}$$

Using the energy balance equation and Taylor expansion, the equation is rewritten as equation A1-4.

$$q_r - q_{r+dr} = -k(2\pi rL)\frac{dT}{dr} + k(2\pi rL)\frac{dT}{dr} - d\left(k(2\pi rL)\frac{dT}{dr}\right) = 0 \tag{A1-4}$$

By using equation A1-5 from equation A1-4, equation A1-6 is set up.

$$r\frac{dT}{dr} = cons\tan t \rightarrow \frac{dT}{dr} = \frac{constatnt}{r} = \frac{C}{r} \tag{A5}$$

$$T(r) = C_1 \ln r + C_2 \tag{A6}$$

By applying the boundary conditions, $T=T_1$ where $r=r_1$ and $T=T_2$ where $r=r_2$, to equation A1-2, constants C_1 and C_2 is determined as shown equation A1-7 and equation A1-8.

$$C_1 = \frac{T_1 - T_2}{\ln\left(\frac{r_1}{r_2}\right)} \tag{A1-7}$$

$$C_1 = T_1 - \frac{T_1 - T_2}{\ln\left(\frac{r_1}{r_2}\right)} \ln r_1 \qquad \text{(A1-8)}$$

The final cylindrical conduction equation is formed as equation A1-9 by substituting equation A1-6 to equation A1-3.

$$q = 2\pi LK \frac{T_1 - T_2}{\ln\left(\frac{r_2}{r_1}\right)} \qquad \text{(A1-9)}$$

2. 2-D Flow Analysis in micro-channel

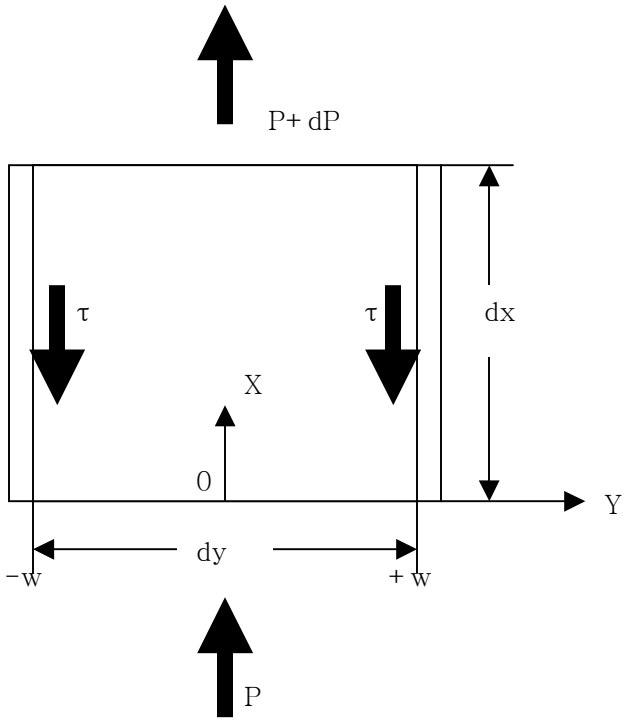


Figure A2 2-Dimensional Flow

From Figure A2, the x-direction of the force balance is set as equation A2-1.

$$\sum F_x = Pdy - Pdy - dPdy - 2\tau dx = 0 \qquad \text{(A2-1)}$$

wher p is pressure, x is distance along with the channel, y is the distance of width of channel.

The force balance equation of flow is derived as followed.

$$-dPdy - 2\tau dx = 0 \tag{A2-2}$$

$$-\,dPdy \;\; = \;\; 2\,\tau\,dx \tag{A2-3}$$

$$-\frac{dP}{dx} = \frac{2\tau}{dy} \tag{A2-4}$$

$$-\frac{dP}{dx} = 2\mu\frac{d^2u}{dy^2} \tag{A2-5}$$

From the boundary conditions, equation A2-5 can be solved as follows.

2-D B.C: (a) u=0, where y=±w

(b) du/dy=0, where y=0

From equation A2-5, du/dy and u is

$$\frac{du}{dy} = -\frac{1}{2\mu}\frac{dP}{dx}y + C_1 \tag{A2-6}$$

$$u = -\frac{1}{4\mu}\frac{dP}{dx}y^2 + C_1 y + C_2 \tag{A2-7}$$

From the boundary condition, C_1=0 and C_2=$\frac{1}{4\mu}\frac{dP}{dx}w^2$.

Finally, Flow velocity of the 2D flow can be written as follows.

$$u = -\frac{1}{4\mu}\frac{dP}{dx}(y^2 - w^2) \tag{A2-8}$$

3. Derivation of meniscus

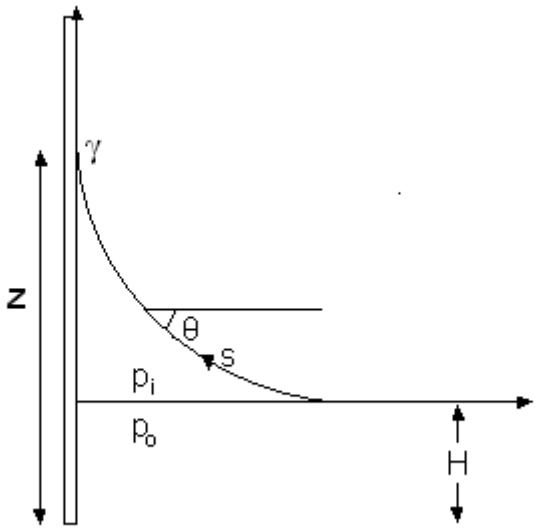


Figure A3 Meniscus surface

The meniscus is tending to be form a circular shape. The circular shape in meniscus can be described by using following mathematical calculations.

$$p_i - p_o = \gamma\frac{d\theta}{ds} \tag{A3-1}$$

$$p_i - p_o = \rho g(Z - H) \tag{A3-2}$$

$$\frac{dZ}{ds} = \sin\theta \tag{A3-3}$$

$$\therefore \gamma\frac{d\theta}{ds} = \rho g(Z - H) \tag{A3-4}$$

Multiply dZ/dS to both side of equation A3-4, and rewritten the equation A3-4.

$$\gamma \frac{d\theta}{ds}\frac{dZ}{dS} = \rho g(Z-H)\frac{dZ}{dS} \tag{A3-5}$$

The equation A3-5 becomes equation A3-6 with using equation A3-3.

$$\gamma \sin\theta \frac{d\theta}{dS} = \rho g(Z-H)\frac{dZ}{dS} \tag{A3-6}$$

$$\therefore \int \gamma \sin\theta d\theta = \int \rho g(Z-H)dZ \tag{A3-7}$$

Equation A3-7 is solved with boundary conditions of θ=0 at Z=H, then equation A3-7 is becomes equation A3-8.

$$\rho g\frac{Z^2}{2} - \rho gHZ + \gamma\cos\theta + \frac{\rho gH^2}{2} - \gamma = 0 \tag{A3-8}$$

Using solution of second polynomial equation solves the equation A3-8.

$$Z = H \pm \sqrt{\frac{2\gamma}{\rho g}(1-\cos\theta)} \tag{A3-9}$$

Since the solution Z is the function of cosine, the shape of meniscus should be forms a circular shape.

APPENDIX B

1. CALIBRATION TEST RESULTS

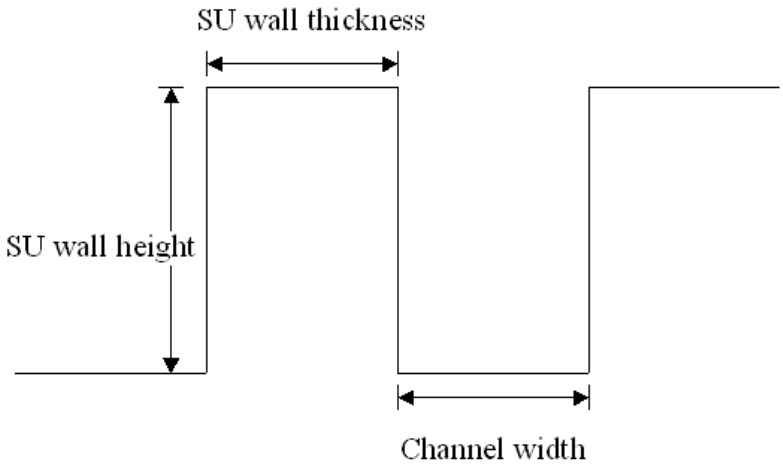


Figure B.1 Detailed dimensions of wick and channel structures

The experiment procedures of calibration tests are presented in section 3.3. The separate calibration tests are performed every time even though the testing membrane has the same wick structures, if the different testing membrane is used in the evaporation test. Calibration result for each wick dimension that used in this work is presented in previous section 5.3. First, a simple feature of 10 x 10 x 10 (10μm high SU8 wall, 10μm wide channel, and 10μm thick SU wall) wicking structure membrane with dual RTD is calibrated to validate the simple numerical calculation. The calibration results of 10 x 10 x 10 wicking structure membrane are shown in Figure B.2. The dimensions of 10 x 10 x 10 are SU8 wall height, channel width, and wall thickness in unit of microns (μm) as shown in Figure B.1. The silicon material membrane is used in this calibration.

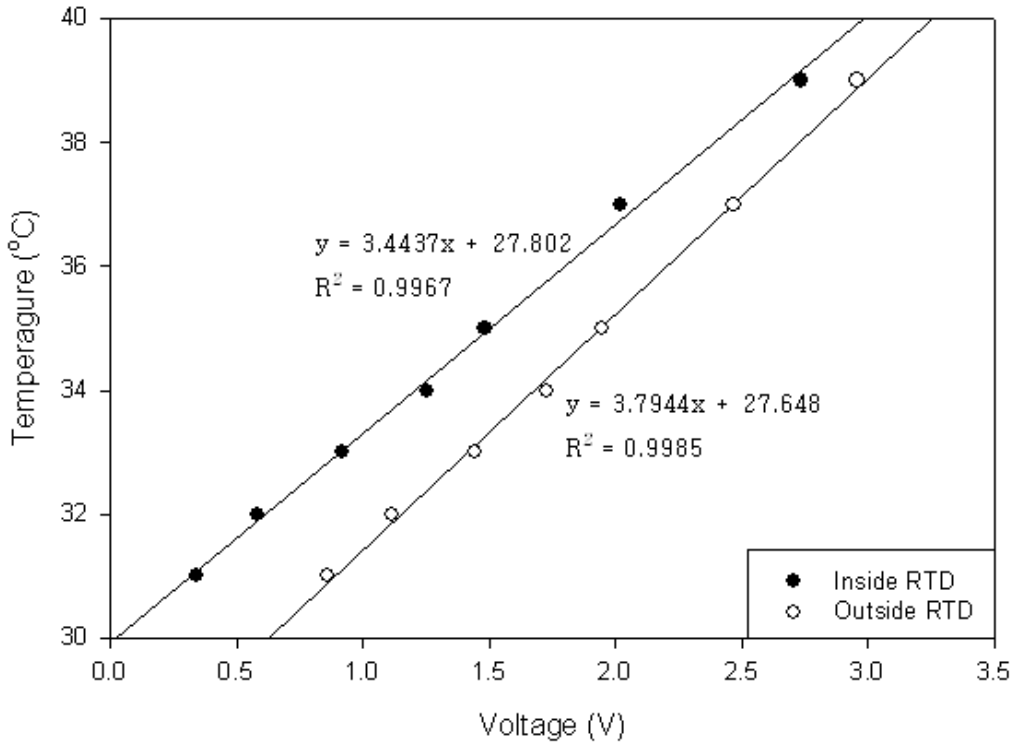


Figure B.2 Calibration test results of 10 x 10 x 10 wicking structure membrane

The voltage and temperature data are collected in a range of water bath temperatures from 30°C to 40°C, and the linear relationships of voltage and temperature are determined for the inside RTD and the outside RTD. The *y* is referred temperature and the *x* is referred voltage in the linear relationship equations. This calibration results has the R^2 value of 0.99, and this value is show that how well the actual experiment data is fitted in the linear curve. The R^2 value of 0.99 is possible and desired because the platinum has the linear relationship between the temperature value and the resistance value.

The dual RTD calibration test results of 40μm high SU8 wall and 5μm wide wicking wall dimensions with 35μm, 50μm, and 70μm wide micro-channels are shown in Figure B.3, Figure B.4, and Figure B.5 respectively. The silicon material membrane is used in these calibration tests.

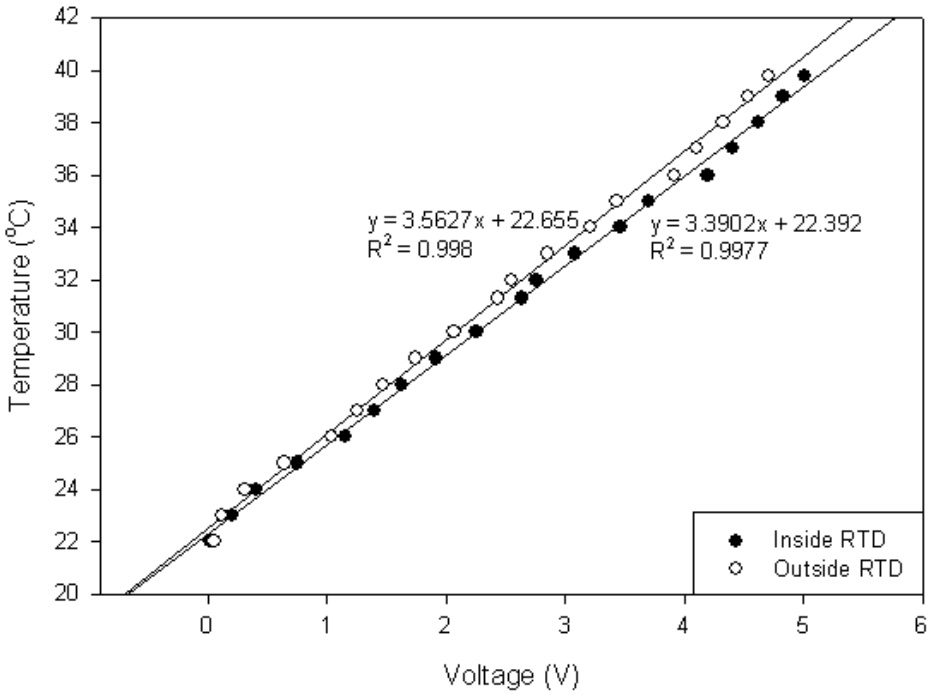


Figure B.3 Calibration test results of 40 x 35 x 5 wicking structure membrane

The voltage and temperature data are collected in a range of water bath temperatures from at 20°C to 42°C in these calibration tests. Again, the linear relationship is determined from these test data. The results also show the R^2 value of 0.99 for all test cases.

The Figure B.6 shows the triple RTD calibration test results of the rectangular micro-wick channel that has the dimensions of 40 x 70 x 5. The details of the triple RTD dimension are previously explained in section 3.2. The edge RTD shows in the Figure B.6 is the third RTD explained in section 3.2. The silicon material membrane is used. The calibration results are recorded between the water bath temperatures of 24°C and 42°C, and these results are used to establish the linear relationships between the temperature and voltage data. The R^2 value of 0.99 is obtained for all three RTDs.

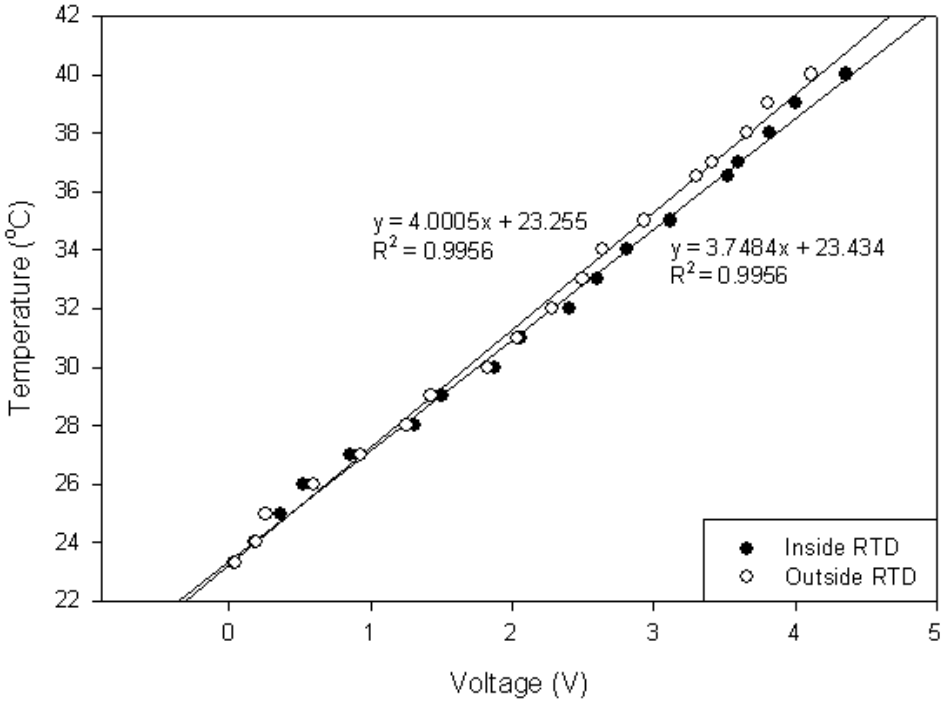


Figure B.4 Calibration test results of 40 x 50 x 5 wicking structure membrane

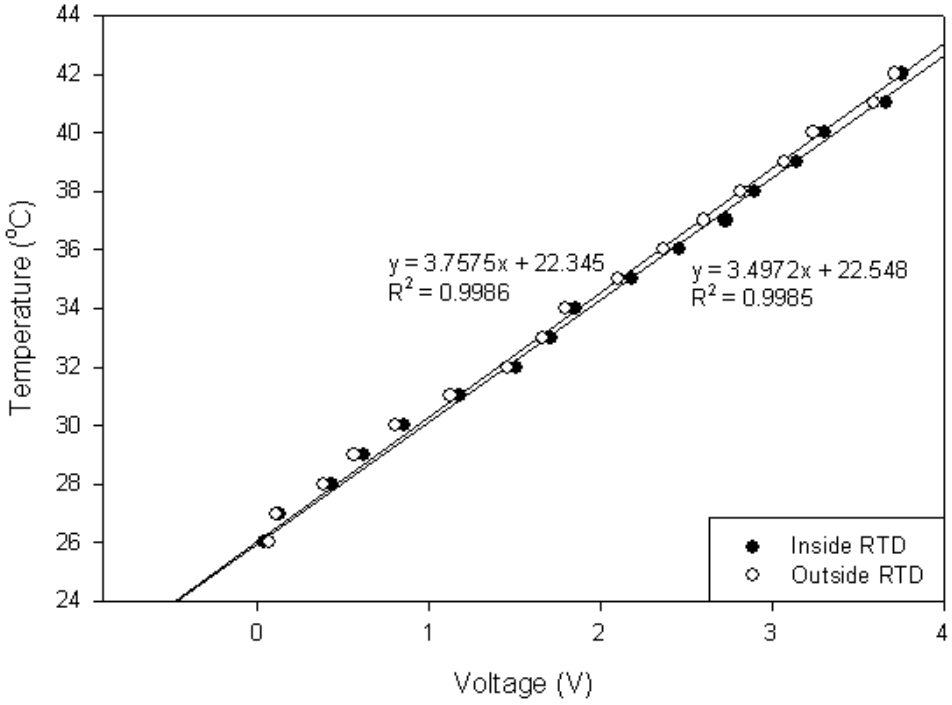


Figure B.5 Calibration test results of 40 x 70 x 5 wicking structure membrane

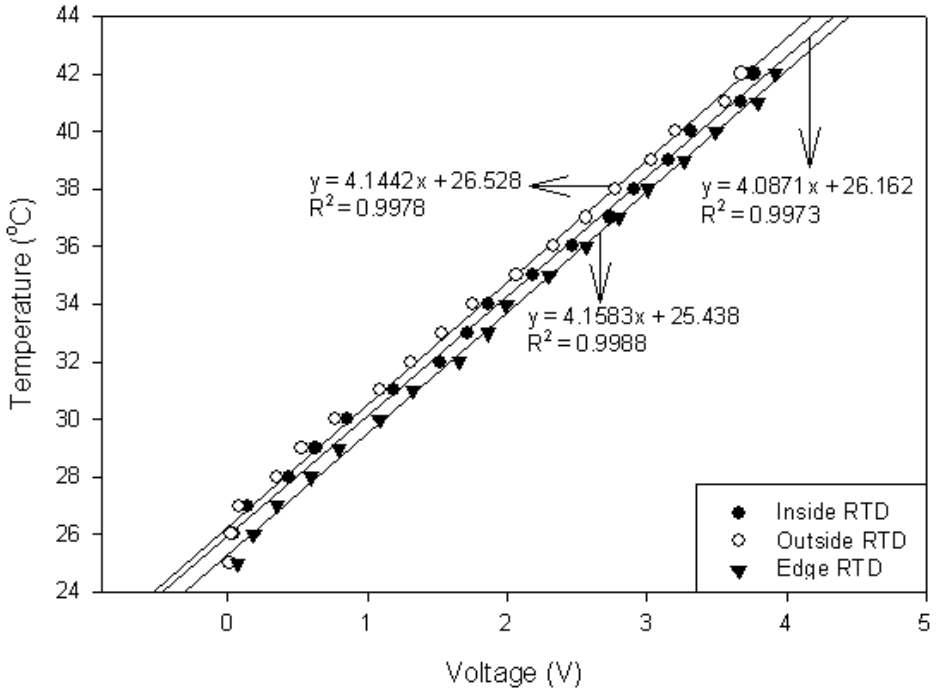


Figure B.6 Triple RTD calibration test results for 40 x 70 x 5 wicking structure membrane

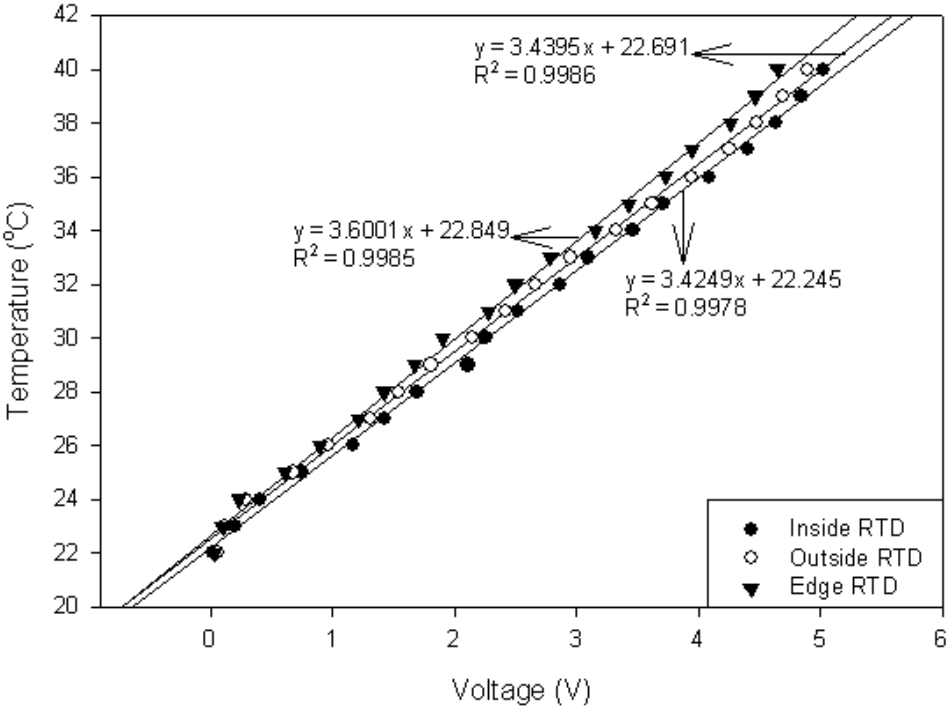


Figure B.7 Triple RTD calibration test results for 40 x 5~80 wicking structure (Si)

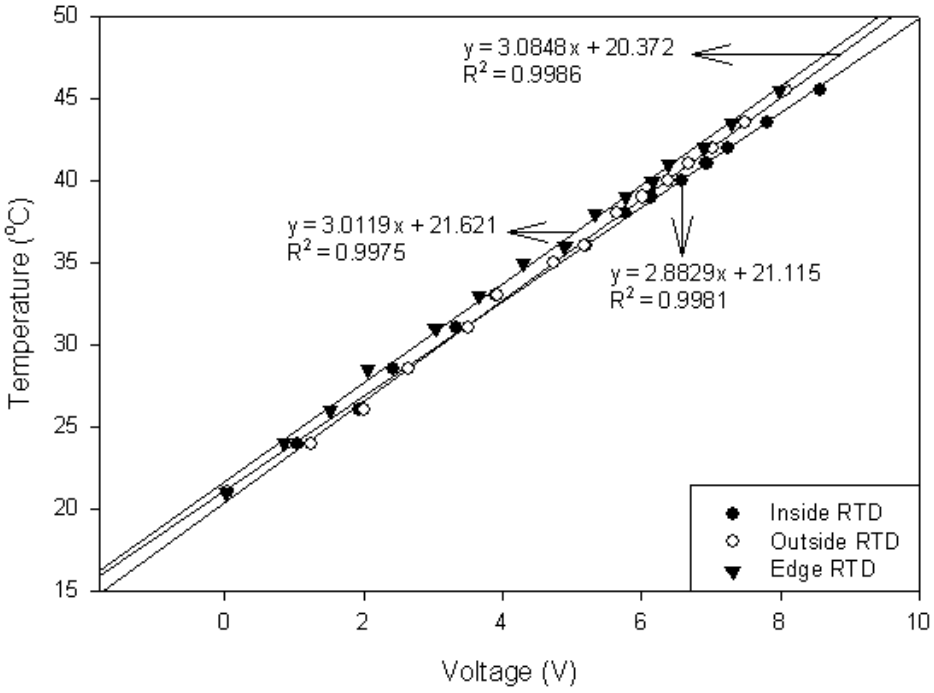


Figure B.8 Triple RTD calibration test results for 40 x 5~80 wicking structure (SiNx)

The triple RTD calibration test results of the 40μm high SU8 wall with the 80μm narrow to 5μm wide tapered channels and SU8 walls are presented in Figure B.7 and Figure B.8. The details of the triple RTD designs are previously explained previously in section 3.2, and the third RTD refers to the edge RTD in the chapter 5.

The silicon material membrane is used. The wicking structure membrane is calibrated with temperatures ranging from 22°C to 40°C. The equations in Figure B.7 shows the voltage to temperature relationship and the R^2 shows the accuracy fit between the linear curve and the data. The R^2 value of 0.99 is obtained.

The silicon nitride material membrane (SiNx) is used in the calibration test results shown in Figure B.8, and the dimensions of wicking structure and channels are the 40μm high SU8 wall with the 80μm narrow to 5μm wide tapered channels and SU8 walls. The

R^2 value of 0.99 is obtained between the experimental data and the calibrated linear curve. The linear relationships of the temperature and voltage are established and shown in equation forms as shown in Figure B.8. These calibration test results are used to convert voltage change to the temperature, thus a temperature profile across the membrane can be measured in the evaporation experiments.

APPENDIX C

1. STEADY STATE EXPERIMENTAL TEST RESULTS

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	35	33.6	33.1	28.6	27.0	7.1	85.0	21.0
			48.4	37.5	29.9	44.6	8.2	100.0	16.3
			57.5	39.0	31.2	46.3	9.1	108.3	15.7

Table C.1 40 x 35 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	35	33.6	33.2	28.4	27.0	6.3	82.0	18.7
			47.8	37.5	30.0	44.6	6.1	91.0	12.7
			57.5	39.0	30.2	54.0	6.4	104.3	11.1

Table C.2 40 x 35 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	50	32.2	31.2	28.7	14.9	2.0	63.3	9.3
			39.9	34.4	30.5	23.1	5.4	73.3	18.2
			49.0	39.0	32.2	40.6	6.1	76.8	12.5
			57.0	40.8	32.2	50.5	6.4	86.7	11.3

Table C.3 40 x 50 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	50	36.5	31.2	27.8	25.5	9.9	117.7	27.2
			50.1	36.2	29.7	38.4	10.2	120	20.3

Table C.4 40 x 50 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	50	35.5	31.5	27.4	24.6	7.9	95	22.4
			49.9	36.2	28.9	40.5	9.9	118.3	20.2
			70.3	41.4	30.9	60.1	10.1	133.3	15.8

Table C.5 40 x 50 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	50	34.0	32.2	28.2	24.6	7.9	93.3	23.2
			47.2	36.3	30.4	35.4	9.8	116.7	20.6
			62.4	41.0	32.2	51.2	11.0	131.7	17.6

Table C.6 40 x 50 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	70	32.0	32.1	28.2	23.4	9.1	108.3	28.4
			42.5	35.2	30.0	34.8	10.3	121.7	24.2
			53.9	40.4	33.2	44.3	11.4	136.7	21.2

Table C.7 40 x 70 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	70	34.1	32.3	28.4	23.4	9.6	108.8	28.1
			44.5	34.9	29.1	34.8	10.8	122.0	24.3
			55.8	37.8	30.8	46.3	11.4	136.7	20.5

Table C.8 40 x 70 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	70	40.0	35.2	30.0	30.6	10.4	125.0	25.9
			54.0	40.4	32.2	42.9	12.0	145.0	22.0
			65.1	42.2	33.2	53.3	12.7	151.2	19.5

Table C.9 40 x 70 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	70	41.0	35.7	30.4	30.6	10.2	122.7	25.2
			54.0	40.4	33.2	43.1	11.3	136.7	20.9
			64.9	42.8	33.4	53.5	12.3	146.7	18.8

Table C.10 40 x 70 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5	70	40.5	35.4	30.0	30.8	10.0	121.7	25.5
			54.0	40.6	33.4	42.9	11.0	131.7	20.4
			65.4	42.6	33.2	53.6	12.4	146.7	18.9

Table C.11 40 x 70 x 5 constant cross section rectangular micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	33.6	33.2	28.8	24.1	8.6	103.3	25.6
			43.7	36.9	31.0	32.7	9.3	111.7	21.3
			57.7	41.5	33.5	44.9	10.6	126.7	18.5

Table C.12 40 x 5~80 Tapered micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	34.4	32.8	28.6	23.5	9.3	113.3	27.0
			44.4	36.3	30.7	32.7	10.6	126.7	23.8
			57.6	41.0	33.3	44.0	11.6	138.5	20.4

Table C.13 40 x 5~80 Tapered micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	32.0	32.1	28.2	23.4	9.1	108.3	28.4
			42.5	35.2	30.0	34.8	10.3	121.2	24.2
			53.9	40.4	33.2	44.3	11.4	136.7	21.2

Table C.14 40 x 5~80 Tapered micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mW)	Inside RTD Temp (°C)	Outside RTD Temp (°C)	Edge RTD Temp (°C)	Power across Membrane (mW)	Power into Evap (mW)	Evaporation Rate (µg/sec)	Efficiency (%)
40	5~80	5~80	36.0	33.9	30.0	28.1	25.8	9.3	111.7	26.2
			42.4	36.9	31.8	30.3	30.0	10.5	125.0	24.7
			56.2	41.6	34.7	33.2	41.9	12.3	139.5	21.2

Table C.15 40 x 5~80 Tapered micro-channel evaporator: Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp1 (°C)	Edge RTD Temp2 (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	13.7	42.9	27.8	22.8	1.24	11.7	139.2	85.4

Table C.16 40 x 5~80 Tapered micro-channel evaporator: SiNx

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside RTD Temp (°C)	Outside RTD Temp1 (°C)	Edge RTD Temp2 (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	12.6	41.4	26.4	22.6	1.1	10.3	125.0	82

Table C.17 40 x 5~80 Tapered micro-channel evaporator: SiNx

2. TRANSIENT OPERATION EXPERIMENTAL TEST RESULTS

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	34.3	34.7	30.7	22.3	10.0	120.0	29.3
			44.0	40.0	34.5	30.9	11.4	136.7	25.9
			57.3	44.2	36.5	42.7	12.4	150.0	21.7

Table C.18 40 x 5~80 Tapered micro-channel evaporator: 10Hz 50% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	34.5	34.6	30.6	22.2	10.0	120.0	29.2
			43.3	40.1	34.5	30.9	11.4	136.7	26.0
			57.4	44.1	36.5	42.8	12.4	148.3	21.5

Table C.19 40 x 5~80 Tapered micro-channel evaporator: 10Hz 50% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	34.2	34.8	30.8	22.3	10.0	120.0	29.2
			44.0	38.5	32.9	30.8	11.2	133.3	24.8
			57.2	43.5	35.7	42.7	12.3	146.7	21.3

Table C.20 40 x 5~80 Tapered micro-channel evaporator: 10Hz 50% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	34.4	32.8	28.6	23.5	10.2	119.7	29.6
			44.4	36.3	30.7	31.7	12.3	133.3	27.7
			57.6	41.0	33.3	43.0	13.5	152.3	23.4

Table C.21 40 x 5~80 Tapered micro-channel evaporator: 10Hz 30% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	33.6	37.3	33.4	22.3	10.5	125.0	31.2
			43.8	42.1	36.7	30.9	12.0	143.3	27.3
			57.3	48.7	41.5	42.8	13.9	165.0	24.2

Table C.22 40 x 5~80 Tapered micro-channel evaporator: 10Hz 30% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	33.7	37.3	33.4	22.5	10.5	130.0	31.9
			43.8	42.1	36.7	31.0	12.0	148.3	28.2
			57.3	48.7	41.5	42.8	13.8	166.7	24.6

Table C.23 40 x 5~80 Tapered micro-channel evaporator: 10Hz 30% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	34.3	34.8	30.8	22.7	10.0	120.0	29.2
			44.1	38.5	32.9	30.9	11.0	131.7	24.8
			57.4	43.5	35.7	43.2	12.3	146.7	21.3

Table C.24 40 x 5~80 Tapered micro-channel evaporator: 20Hz 50% duty cycle, Si

SU8 Height (microns)	SU8 Wall Width (microns)	Channel Width (microns)	Power into Heater (mw)	Inside Avg RTD Temp (°C)	Outside Avg RTD Temp (°C)	Power across Membrane (mw)	Power into Evap (mw)	Evaporation Rate (µg/min)	Efficiency (%)
40	5~80	5~80	34.1	35.2	31.0	23.9	10.5	123.8	30.7
			44.5	40.6	34.5	34.2	11.8	143.3	26.6
			56.7	46.9	39.6	40.9	13.3	158.3	23.4

Table C.25 40 x 5~80 Tapered micro-channel evaporator: 20Hz 30% duty cycle, Si

APPENDIX D

```
Program ThreeD STEADY AND TRANSIENT

      Use MSIMSL
      Implicit Double Precision (a-h,o-z),Integer(i-n)
      Double Precision, Allocatable:: Coeff(:,:),Tnodal(:,:),Coe(:,:)
      Double Precision, Allocatable:: Tinp(:),Tout(:),Temp(:,:),T1(:)
      Double Precision, Allocatable:: conve(:,:),time(:),r(:),T2(:)
      Double Precision, Allocatable:: Einp(:),Est(:,:),Tavg(:,:)
      Double Precision, Allocatable:: pressg(:),volg(:),deflg(:)
      Double Precision, Allocatable:: Tgasav(:),delth(:),deltaP(:,:)
      Double Precision, Allocatable:: viniu(:),miniu(:),Tvg(:)
      Double Precision, Allocatable:: vinil(:),minil(:),massgain(:)
      Double Precision, Allocatable:: Qfix(:),sum5(:,:)
      Double Precision, Allocatable:: Pressavg(:),Tempavg(:)
      Double Precision, Allocatable:: Pcav(:,:),Tcav(:,:)
      Double Precision, Allocatable:: mgen(:,:),mtot(:,:),mnet(:)
      Double Precision, Allocatable::mfluxlow(:,:),mevaplow(:,:)
      Double Precision, Allocatable::mfluxupp(:,:),mcondupp(:,:)
      Double Precision, Allocatable::mevapsum(:),mevaptot(:,:)
      Double Precision, Allocatable::mcondsum(:),mcondtot(:,:)
      Double Precision, Allocatable:: qoutlow(:),qinupp(:)
      Double Precision, Allocatable:: Tlvlow(:,:),Tlvupp(:,:)
      Double Precision, Allocatable:: Liqlow(:),Liqupp(:)
      Double Precision, Allocatable:: mremlow(:),mremupp(:)
      Double Precision, Allocatable:: Ebulkstep(:,:),Ebulksum(:,:)
      Double Precision, Allocatable:: Elost(:),Workmemtot(:)
      Double Precision, Allocatable:: Pgage(:),Deflupp(:),Volupp(:)
      Double Precision, Allocatable:: evpenrg(:),conenrg(:)
      Double Precision, Allocatable:: Emats(:),Workmemu(:),Effupp(:)
      Double Precision, Allocatable:: Efflow(:),Efftot(:)
      Double Precision, Allocatable:: hcell(:,:),Workmeml(:)
      Double Precision, Allocatable:: hcellupp(:,:),hcelllow(:,:)
      Double Precision, Allocatable:: Defllow(:),Vollow(:),RTD(:)
      Double Precision thmat(15),deltay(15),Tfix(10),tthmat(15)
      Double Precision delr,distr,edgel,delt,pulse,deltayy(15)
      Double Precision hnwy,hnex,hney,hsex,hsey,hswy,LatE,TotM
      Double Precision Tnwy,Tnex,Tney,Tsex,Tsey,Tswy
      Double Precision Tnw,Tne,Tse,Tsw,Tn,Te,Ts,hn,hs,Ti
      Double Precision rmat1(6),rmat2(6),rmat3(6),rmat4(6),rmat5(6)
      Double Precision rmat6(6),rmat7(6),rmat8(6),rmat9(6),rmat10(6)
      Double Precision rmat11(6),rmat12(6),rmat13(6),rmat14(6)
      Double Precision rmat15(6),props(15,6)
      Double Precision rrmat1(6),rrmat2(6),rrmat3(6),rrmat4(6),rrmat5(6)
      Double Precision rrmat6(6),rrmat7(6),rrmat8(6),rrmat9(6)
      Double Precision rrmat10(6),rrmat11(6),rrmat12(6),rrmat13(6)
      Double Precision rrmat14(6),rrmat15(6),pprops(15,6)
      Double Precision heffe(15),contactangle
      Double Precision mairini,S1,SIO,FC,SU,SUR
      Double Precision mol,rhol,hfg
      Double Precision volcoeff,Vcoeff(9,2)
      Double Precision miniutot,miniltot
      Double Precision heat
      Integer, Allocatable:: nodmati(:,:),nodmat(:,:),intw(:),west(:)
      Integer, Allocatable:: inte(:),east(:),nodint(:,:),Nfix(:)
      Integer ndim1(6),ndim2(6),ndim3(6),ndim4(6),ndim5(6),ndim6(6)
      Integer ndim7(6),ndim8(6),ndim9(6),ndim10(6),ndim11(6),ndim12(6)
      Integer ndim13(6),ndim14(6),ndim15(6),pdim(15,6),ppdim(15,6)
      Integer nndim1(6),nndim2(6),nndim3(6),nndim4(6),nndim5(6)
      Integer nndim6(6),nndim7(6),nndim8(6),nndim9(6),nndim10(6)
      Integer nndim11(6),nndim12(6),nndim13(6),nndim14(6),nndim15(6)
   Integer nrow,ncol,neq,jj,nsteps,nflux,mats,skip,nangle
      Integer BCs,BCn,BCe,BCnw,BCne,BCsw,BCse
      Integer memtype,RTDnodes(2)
      Integer matll,ii1,ii2,ii7,ii9
      Integer adialow,adiabats,WF
```

```
      Parameter (Ipath=1)

C     Ouput Files
      Open(Unit=1012,File='tempsav1.txt')
      Open(Unit=1013,File='tempsav2.txt')
      Open(Unit=1014,File='tempsav3.txt')
      Open(Unit=1016,File='massgain.txt')

C     ------------------------------------------------------------------------------
C     GEOMETRY, MATERIALS, MESH, VARIABLES, AND TIME
C     ------------------------------------------------------------------------------
C     Membrane edge length [m]
      edgel=5.E-3
C     membrane type 1=Si 2=SiN
      memtype=1
C     Material thicknesses [m]
      S1=2.2E-6
      SIO=80.E-9
      FC=40.0E-6
      SU=40.0E-6
      thmat(1)=S1                     !Si or SiNx
      thmat(2)=SIO              !SiO
      thmat(3)=SU                     !SU8
      thmat(4)=FC                     !FC77 for near wall
      thmat(5)=FC                     !FC77 for near center_need to be calculated
      thmat(6)=SIO              !SiO
      thmat(7)=S1                     !Si
      thmat(8)=0
      thmat(9)=0
      thmat(10)=0
      thmat(11)=0
      thmat(12)=0
      thmat(13)=0
      thmat(14)=0
      thmat(15)=0

C     Remember to change write statement in output subroutine
      nrow=22                         !Number of nodes in y direction
      ncol=18                         !Number of nodes in x direction
      nangle=7                  !Number of nodes in theta direction
    neq=nrow*ncol*nangle              !Number of equations i.e. number of nodes

C     Number of materials in structure
      mats=7            !Total number of materials

C     Corner node locations of each material in mesh
      data ndim1 /1,1,5,18,1,3/
      data ndim2 /5,1,9,18,1,3/
      data ndim3 /9,1,22,18,1,3/
      data ndim4 /7,1,12,18,1,3/
      data ndim5 /12,1,18,18,1,3/
      data ndim6 /18,1,20,18,1,3/
      data ndim7 /20,1,22,18,1,3/
      data ndim8 /17,1,19,18,1,1/
      data ndim9 /1,1,3,18,1,3/
      data ndim10 /3,1,5,18,1,3/
      data ndim11 /5,1,7,18,1,3/
      data ndim12 /7,1,12,18,1,3/
      data ndim13 /12,1,18,18,1,3/
      data ndim14 /18,1,20,18,1,3/
      data ndim15 /20,1,22,18,1,3/

      data nndim1 /1,1,5,18,4,5/
      data nndim2 /5,1,9,18,4,5/
      data nndim3 /9,1,18,18,4,5/
      data nndim4 /18,1,22,18,4,5/
      data nndim5 /12,1,18,18,4,5/
      data nndim6 /18,1,20,18,4,5/
      data nndim7 /20,1,22,18,4,5/
      data nndim8 /17,1,19,18,1,1/
```

```
      data nndim9 /19,1,21,18,1,1/
      data nndim10 /15,1,17,18,1,1/
      data nndim11 /17,1,19,18,1,1/
      data nndim12 /19,1,21,18,1,1/
      data nndim13 /3,1,5,5,1,1/
      data nndim14 /3,1,5,5,1,1/
      data nndim15 /3,1,5,5,1,1/

C     Material Properties
C     /thermal conductivity [W/m*K],specific heat [J/kg*K],density[kg/m^3]/thermal diffusivity[m^2/s]/
      data rmat1 /148.,712.,2330.,8.9213E-5,0,0/
     data rmat2 /1.38,745.,2220.,8.3439E-7,0,0/
    data rmat3 /.057,1100.,1680.,3.0844E-8,0,0/
      data rmat4 /.0263,1007.,1.1614,2.2488E-5,0,0/
    data rmat5 /.057,1100.,1680.,3.0844E-8,0,0/
      data rmat6 /1.38,745.,2220.,8.3439E-7,0,0/
      data rmat7 /148.,712.,2330.,8.9213E-5,0,0/
      data rmat8 /.057,1100.,1680.,3.0844E-8,0,0/
      data rmat9 /.057,1100.,1680.,3.0844E-8,0,0/
      data rmat10 /.057,1100.,1680.,3.0844E-8,0,0/
      data rmat11 /1.38,745.,2220.,8.3439E-7,0,0/
      data rmat12 /148.,712.,2330.,8.9213E-5,0,0/
      data rmat13 /148.,712.,2330.,8.9213E-5,0,0/
      data rmat14 /148.,712.,2330.,8.9213E-5,0,0/
      data rmat15 /148.,712.,2330.,8.9213E-5,0,0/

      data rrmat1 /148.,712.,2330.,8.9213E-5,0,0/
    data rrmat2 /1.38,745.,2220.,8.3439E-7,0,0/
      data rrmat3 /.057,1100.,1680.,3.0844E-8,0,0/
      data rrmat4 /.0263,1007.,1.1614,2.2488E-5,0,0/
      data rrmat5 /.2,1190.,1280.,1.313E-7,0,0/
      data rrmat6 /1.38,745.,2220.,8.3439E-7,0,0/
      data rrmat7 /148.,712.,2330.,8.9213E-5,0,0/
      data rrmat8 /.057,1100.,1680.,3.0844E-8,0,0/
      data rrmat9 /.057,1100.,1680.,3.0844E-8,0,0/
      data rrmat10 /.057,1100.,1680.,3.0844E-8,0,0/
      data rrmat11 /1.38,745.,2220.,8.3439E-7,0,0/
      data rrmat12 /148.,712.,2330.,8.9213E-5,0,0/
      data rrmat13 /148.,712.,2330.,8.9213E-5,0,0/
      data rrmat14 /148.,712.,2330.,8.9213E-5,0,0/
      data rrmat15 /148.,712.,2330.,8.9213E-5,0,0/

C     Material Library
C    Si       /148.,712.,2330.,8.9213E-5/
C     SiO2     /1.38,745.,2220.,8.3439E-7/
C     Air            /.0263,1007.,1.1614,2.2488E-5/
C     FC77     /.057,1100.,1680.,3.0844E-8/
C     Brick    /1.,960.,2650.,3.9308E-7/
C     SiN            /15.,700.,2400,8.9286E-6/
C     Foam     /.026,1045.,70.,3.5543E-7/
C     Silver   /429.,235.,10500.,1.74E-4/
C     Epoxy    /.25,1890.,1280.,1.033E-7/
C     SU8            /.2,1190.,1280.,1.313E-7/
C     Copper  /401.,385.,8933.,1.166E-4/

C     Identify the working fluid as air or saturated liquid-vapor
      WF=1    !0=Air, 1=Saturated Liquid Vapor

C     Specifies liquid-vapor interface rows
      adialow=18                                    !Abiabatic suraface of membrane
     matll=5                                        !Material of membrane liquid layer

C     Time Conditions
      delt=.00001                   !Time step
      [s]
      pulse=1                 !Pulse width                                  [s]
      nsteps=10000            !Number of time steps
      skip=10                       !Number of time steps to skip when writing data

C     Initial material temperatures
```

```
      Ti=294.8                  !Initial nodal temperatures          [K]


C     Properties of working fluid
      rhol=1780.                      !Density of liquid FC-77       [kg/m^3]
      mol=.416                  !MW of liquid FC-77                  [kg/mol]
      hfg=84000.                      !Latent heat of FC-77                  [J/kg]
      contactangle=0.72     !contact angle [degree]
      SUR=8.0E-3        !surface tension of FC-77 [N/m]

C     ---------------------------------------------------------------------------
C     BOUNDARY CONDITIONS
C     ---------------------------------------------------------------------------
C     Flux or Temp at South Face: Flux=1 Temp=2
      nflux=1

      Allocate(Nfix(ncol*nangle),Qfix(ncol*nangle))
      num=380
c     i=1
c     Do n=1,(ncol-11)*nangle
c           If(n.EQ.(ncol-11)*i)Then
c           Nfix(n)=num
c           num=num+(nrow*ncol-(ncol-3))
c           i=i+1
c           Else
c           Nfix(n)=num
c           num=num+1
c           Endif
c     Enddo

c    num=380
      Do n=1,ncol-11
            Nfix(n)=num
            num=num+1
      Enddo
      num=776
      Do n=10,(ncol-11)*2
            Nfix(n)=num
            num=num+1
      Enddo
      num=1172
      Do n=19,(ncol-11)*3
            Nfix(n)=num
            num=num+1
      Enddo
      num=1568
      Do n=28,(ncol-11)*4
            Nfix(n)=num
            num=num+1
      Enddo
      num=1964
      Do n=37,(ncol-11)*5
            Nfix(n)=num
            num=num+1
      Enddo



      heat=.034000E6                                                  !heat input for unit area
(w/m^2)

      Do i=1,(ncol-11)*(nangle-2)
            Qfix(i)=heat
      Enddo

C     Specifies type of boundary condiditon at corners, east, and north faces
      BCn=1                           !1=Convection 2=Constant Temperature
      BCs=1                           !1=Convection 2=Constant Temperature
      BCe=1                           !1=Convection 2=Constant Temperature
      BCnw=1                          !1=Convection 2=Constant Temperature
```

```
      BCne=1                          !1=Convection 2=Constant Temperature
      BCsw=1                          !1=Convection 2=Constant Temperature
      BCse=1                          !1=Convection 2=Constant Temperature

C     Boundary Conditions at Corner Nodes
C     Northwest Corner
      Tnw=294.8                       !Temperature for fixed temp [K]
      Tnwy=294.8                      !Temperature for convection [K]
      hnwy=0.
C     Northeast Corner
      Tne=294.8                       !Temperature for fixed temp [K]
      Tnex=294.8                      !Temperature for convection [K]
      Tney=294.8                      !Temperature for convection [K]
      hnex=135000.          !Convective coeff                 [W/m^2*K]
      hney=0.                         !Convective coeff               [W/m^2*K]
C     Southeast Corner
      Tse=294.8                       !Temperature for fixed temp [K]
      Tsex=294.8                      !Temperature for convection [K]
      Tsey=294.8                      !Temperature for convection [K]
      hsex=135000.          !Convective coeff                 [W/m^2*K]
      hsey=0.                         !Convective coeff               [W/m^2*K]
C     Southwest Corner
      Tsw=294.8                       !Temperature for fixed temp [K]
      Tswy=294.8                      !Temperature for convection [K]
      hswy=0.                         !Convective coeff               [W/m^2*K]

C     Boundary Conditions at Surface Nodes
C     North Face
      Tn=294.8              !Temperature for fixed or convective temp [K]
      hn=250.                         !Convective coeff               [W/m^2*K]
C     South Face
      Ts=294.8              !Temperature for fixed or convective temp [K]
      hs=0.                           !Convective coeff               [W/m^2*K]
C     East Face
      Te=294.8              !Temperature for fixed or convective temp [K]

C     -------------------------------------------------------------------------
C     ALLOCATION OF VARIABLES
C     -------------------------------------------------------------------------
C     Allocates dimensions to matrices and vectors
c     neq=2772
      Allocate (Coeff(neq,neq),Tinp(neq),Tout(neq),Tnodal(neq,neq))
      Allocate (Temp(neq,nsteps+1),Tavg(nsteps+1,mats),Coe(396,396))
      Allocate (nodmati(neq,2),nodmat(neq,2),nodint(neq,2),T1(396))
      Allocate (east(mats),west(mats),inte(neq),intw(neq),T2(396))
      Allocate (conve(neq,2),time(nsteps+1),delth(neq),Tvg(nsteps+1))
      Allocate (r(ncol),Einp(nsteps+1),Est(nsteps+1,mats+3))
      Allocate (viniu(neq),miniu(neq))
      Allocate (vinil(neq),minil(neq),massgain(nsteps+1))
      Allocate (pressg(nsteps+1),volg(nsteps+1))
      Allocate (deflg(nsteps+1),Tgasav(nsteps+1))
      Allocate (Pressavg(nsteps+1),Tempavg(nsteps+1))
      Allocate (Pcav(ncol,nsteps+1),Tcav(ncol,nsteps+1))
      Allocate (mgen(ncol,nsteps+1),mtot(ncol,nsteps+1),mnet(nsteps+1))
      Allocate (mfluxlow(ncol,nsteps+1),mevaplow(ncol,nsteps+1))
      Allocate (mfluxupp(ncol,nsteps+1),mcondupp(ncol,nsteps+1))
      Allocate (mevapsum(nsteps+1),mevaptot(ncol,nsteps+1))
      Allocate (mcondsum(nsteps+1),mcondtot(ncol,nsteps+1))
      Allocate (qoutlow(neq),qinupp(neq))
      Allocate (Tlvlow(ncol,nsteps+1),Tlvupp(ncol,nsteps+1))
      Allocate (Liqlow(nsteps+1),Liqupp(nsteps+1))
      Allocate (mremlow(nsteps+1),mremupp(nsteps+1))
      Allocate (Elost(nsteps+1),sum5(ncol,nsteps+1))
      Allocate (Ebulkstep(mats,nsteps+1),Ebulksum(mats,2))
      Allocate (Pgage(nsteps+1),Deflupp(nsteps+1))
      Allocate (evpenrg(nsteps+1),conenrg(nsteps+1))
      Allocate (Emats(nsteps+1),Volupp(nsteps+1),Workmemu(nsteps+1))
      Allocate (Effupp(nsteps+1),hcell(ncol,nsteps+1))
      Allocate (hcellupp(ncol,nsteps+1),hcelllow(ncol,nsteps+1))
      Allocate (Defllow(nsteps+1),Vollow(nsteps+1))
```

```
      Allocate (Workmeml(nsteps+1),Workmemtot(nsteps+1))
      Allocate (Efflow(nsteps+1),Efftot(nsteps+1))
      Allocate (RTD(nsteps+1),deltaP(ncol,nsteps+1))

C     ------------------------------------------------------------------------
C     BEGINNING OF PROGRAM
C     ------------------------------------------------------------------------
C     Zeros matrices and vectors before computation
      Call Zero(Tinp,Tout,Coeff,Tnodal,Temp,time,
     +Est,Einp,r,conve,props,Emats,Effupp,Efflow,Efftot,
     +qoutlow,qinupp,Liqlow,Liqupp,mremlow,mremupp,
     +vinil,minil,pdim,nodmat,nodmati,east,west,thmat,
     +inte,intw,neq,ncol,mats,nsteps,ppdim,pprops,delth)

C     Stores material properties and nodal coordinates in a matrix
      Call Matprops(rmat1,rmat2,rmat3,rmat4,rmat5,
     +rmat6,rmat7,rmat8,rmat9,rmat10,rmat11,rmat12,rmat13,rmat14,rmat15,
     +props,thmat,tthmat,
     +ndim1,ndim2,ndim3,ndim4,ndim5,ndim6,ndim7,ndim8,ndim9,ndim10,
     +ndim11,ndim12,ndim13,ndim14,ndim15,pdim)

      Call MMatprops(rrmat1,rrmat2,rrmat3,rrmat4,rrmat5,
     +rrmat6,rrmat7,rrmat8,rrmat9,rrmat10,rrmat11,rrmat12,rrmat13,
     +rrmat14,rrmat15,pprops,tthmat,
     +nndim1,nndim2,nndim3,nndim4,nndim5,nndim6,nndim7,
     +nndim8,nndim9,nndim10,nndim11,nndim12,nndim13,nndim14,nndim15,
     +ppdim)

C     Calculates spatial steps and distances
      Call Distances(deltay,thmat,r,delr,distr,edgel,
     +pdim,mats,ncol,nangle,tthmat,deltayy,ppdim,neq,delth)

C     Assigns material number to the node numbers
      Call Assign(pdim,nodmat,nodmati,nodint,mats,neq,nrow,ncol,ppdim)

C     Finds interface nodes at east and west face
      Call Faceint(pdim,east,west,inte,intw,mats,neq,ncol,nrow,
      +nangle,ppdim)


C     Assigns effective conductive BC at east face
      Call EffeastBC(heffe,conve,pdim,neq,ncol,nrow,nangle,mats,ppdim)

      tstep=0.
      time(1)=0.
      Do 1310 kkk=1,1
C     Sets initial time and temperature of nodes and gas parameters
      Call Initial(Temp,Tavg,Tout,Ti,pressg,Tgasav,volg,
     +deflg,thmat,time,tstep,distr,mairini,
      +Pressavg,Tempavg,Tcav,Pcav,mgen,r,Tlvupp,Tlvlow,vcavi,Pvapi,
      +Pgage,Deflupp,Volupp,volcoeff,Vcoeff,edgel,hcell,
      +Vollow,Defllow,hcellupp,hcelllow,RTD,
     +delr,mol,ncol,neq,nsteps,mats)


C     ------------------------------------------------------------------------
C     BEGINNING OF TIME DEPENDANT LOOPING PORTION OF PROGRAM
C     ------------------------------------------------------------------------
c     if program is steady state condition (use this do loop)
      Do 10 jj=1,nsteps                    !Time loop
            tstep=tstep+delt               !Counts of in time
            time(jj+1)=tstep               !Stores time increment in vector

            If(tstep.GT.pulse)Then
                  Do i=1,ncol*(nangle-2)
                        Qfix(i)=0.         !Heat flux off when pulse time exceeded
                  Enddo
            Endif

c     if program is transient condition
```

```
C       use this do loop and change frequency and duty cycle
c       Also goto Call Energy, and Subroutine Energy: Change heating conditions
c       Do 10 jj=1,nsteps                       !Time loop
C               Print*,'Step', jj
c               tstep=tstep+delt                !Counts of in time
c               time(jj+1)=tstep                !Stores time increment in vector

c               If(tstep.GT.0.05)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0.              !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.1)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.15)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.2)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.25)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.3)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.35)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.4)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.45)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.5)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.55)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.6)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.65)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
```

```
c                       Enddo
c               Endif
c               If(tstep.GT.0.7)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.75)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.8)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.85)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.9)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.0.95)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.0)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif

c               If(tstep.GT.1.05)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0.              !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.1)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.15)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.2)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.25)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.3)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.1.35)Then
c                       Do i=1,ncol*(nangle-2)
```

```
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.4)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.45)Then
c               Do i=1,ncol*(nangle-2)
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.5)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.55)Then
c               Do i=1,ncol*(nangle-2)
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.6)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.65)Then
c               Do i=1,ncol*(nangle-2)
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.7)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.75)Then
c               Do i=1,ncol*(nangle-2)
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.8)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.85)Then
c               Do i=1,ncol*(nangle-2)
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.9)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.1.95)Then
c               Do i=1,ncol*(nangle-2)
c                       Qfix(i)=0       !Heat flux off when pulse time exceeded
c               Enddo
c       Endif
c       If(tstep.GT.2.0)Then
c               Do i=1,8
c                       Qfix(i)=heat            !Heat flux off when pulse time exceeded
c               Enddo
c       Endif

c       If(tstep.GT.2.05)Then
```

```
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0.          !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.1)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.15)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0           !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.2)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.25)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0           !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.3)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.35)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0           !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.4)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.45)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0           !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.5)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.55)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0           !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.6)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.65)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0           !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.7)Then
c                       Do i=1,8
c                               Qfix(i)=heat                !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.75)Then
```

```
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.8)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.85)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.9)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.2.95)Then
c                       Do i=1,ncol*(nangle-2)
c                               Qfix(i)=0               !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif
c               If(tstep.GT.3.0)Then
c                       Do i=1,8
c                               Qfix(i)=heat                    !Heat flux off when pulse time exceeded
c                       Enddo
c               Endif

C       Calculates deltay spacing every time step due to evaporation of liquid
        Call Distances(deltay,thmat,r,delr,distr,edgel,
     +pdim,mats,ncol,nangle,tthmat,deltayy,ppdim,neq,delth)

        If(WF.EQ.1)Then
C       Turns abiabats on at liquid-solid interfaces
        adiabats=1              !1=Adiabats are 'on', 0=Adiabats are 'off'
        Endif

        Call TempBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
        +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
     +pdim,nodmati,nodint,mats,Coe,T1,T2)

        Call TempBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
        +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
     +pdim,nodmati,nodint,mats,Coe,T1,T2,delth)

        Call TempBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
        +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
     +pdim,nodmati,nodint,mats,Coe,T1,T2,delth)

        Call TempBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
        +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adiabats,nangle,
     +ppdim,nodmati,nodint,mats,Coe,T1,T2,delth)

        Call TempBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
```

```
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
    +nflux,BCn,BCs,BCe,adiabats,nangle,
    +ppdim,nodmati,nodint,mats,Coe,T1,T2,delth)

     If(WF.EQ.1)Then
C    ==================Begin Mass Transfer=======================
C    Sets initlal liquid layer volumes for evaporation
     Call Liqvols(vinil,minil,viniu,miniu,miniltot,miniutot,
     +mremlow,mremupp,Liqlow,Liqupp,thmat,deltayy,
     +r,delr,rhol,deltay,pdim,ii7,ii9,tthmat,
    +mats,matll,nsteps,neq,ncol,ppdim)

C    Solves for mass transfer and saturated cavity conditions
     Call Evaporation(Pressavg,Tempavg,Pcav,Tcav,mgen,mtot,
    +mnet,mfluxlow,mevaplow,mfluxupp,mcondupp,mevapsum,mevaptot,
    +mcondsum,mcondtot,Tlvlow,Tlvupp,qoutlow,qinupp,
    +props,r,Tout,thmat,mol,hfg,delr,delt,delth,
     +minil,rhol,Liqlow,Liqupp,mremlow,mremupp,edgel,
     +Pgage,Deflupp,Volupp,volcoeff,evpenrg,conenrg,
    +matll,adialow,mats,distr,hcell,
     +Defllow,Vollow,hcellupp,hcelllow,sum5,deltaP,
    +ii1,ii2,ii3,ncol,nrow,neq,nsteps,jj,memtype)

C     Turns abiabats off at liquid-solid interfaces
      adiabats=0                    !1=Adiabats are 'on', 0=Adiabats are 'off'

     Do i=1,neq
          Tout(i)=Temp(i,jj)
     Enddo

     If (jj.EQ.nsteps)Then
     LatE=(((Temp(716,nsteps)+Temp(717,nsteps))/2-
     +Temp(718,nsteps))*6.28*thmat(1)*props(1,1))/0.163
     TotM=(heat/1E6-LatE)/(((nsteps**2)*delt)*hfg)
     Do i=1,nsteps
     massgain(i)=TotM*i
     Enddo
     Endif
     Call TTempBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
    +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
    +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
    +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
    +pdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,qoutlow)

     Call TTempBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
    +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
    +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
    +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
    +pdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,qoutlow,delth)

     Call TTempBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
    +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
    +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
    +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
    +pdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,qoutlow,delth)

     Call TTempBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
    +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
    +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
    +nflux,BCn,BCs,BCe,adiabats,nangle,
    +ppdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,delth)

     Call TTempBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
    +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
    +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
```

```
     +nflux,BCn,BCs,BCe,adiabats,nangle,
     +ppdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,delth)

            Do i=pdim(4,1)+1,pdim(4,3)-1
                  Do j=pdim(4,2)+1,pdim(4,4)-1
                        Do k=1,3
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        Tout(m)=Tcav(j-1,jj+1)
                     Enddo
                  Enddo
            Enddo
c           ================End Mass Transfer=========================
            Endif

            Call Energy(Est,Einp,props,Qfix,Tout,Tavg,Ti,mnet,time,hfg,
     +tstep,pulse,r,delr,delt,Nfix,pdim,thmat,heffe,Te,Ebulkstep,Emats,
     +Ebulksum,Elost,nrow,ncol,neq,nsteps,mats,k,jj)

                  Do i=1,neq
                        Temp(i,jj+1)=Tout(i)
                  Enddo

10          Continue


      Call Output1(Coeff,Tnodal,Tout,Temp,Est,time,deltay,r,
     +pressg,deflg,volg,Tgasav,nrow,ncol,neq,nsteps,mats,skip,massgain)

1310        Continue
            End

C           ***************************************************************
C           ********************** SUBROUTINES ****************************
C           ***************************************************************
C           ---------------------------------------------------------
C           ZEROS MATRICES AND VECTORS BEFORE COMPUTATION
            Subroutine Zero(Tinp,Tout,Coeff,Tnodal,Temp,time,
     +Est,Einp,r,conve,props,Emats,Effupp,Efflow,Efftot,
     +qoutlow,qinupp,Liqlow,Liqupp,mremlow,mremupp,
     +vinil,minil,pdim,nodmat,nodmati,east,west,thmat,
     +inte,intw,neq,ncol,mats,nsteps,ppdim,pprops,delth)

            Double Precision Tinp(neq),Tout(neq),Coeff(neq,neq)
            Double Precision Tnodal(neq,neq),Temp(neq,nsteps+1),r(ncol)
            Double Precision Est(nsteps+1,mats+3),Einp(nsteps+1),conve(neq,2)
            Double Precision time(nsteps+1),props(15,6),pprops(15,6)
            Double Precision qoutlow(neq),qinupp(neq),delth(neq)
            Double Precision vinil(neq),minil(neq),thmat(15)
            Double Precision Liqlow(nsteps+1),Liqupp(nsteps+1)
            Double Precision mremlow(nsteps+1),mremupp(nsteps+1)
            Double Precision Emats(nsteps+1),Effupp(nsteps+1)
            Double Precision Efflow(nsteps+1),Efftot(nsteps+1)
      Integer pdim(15,6),nodmat(neq,2),nodmati(neq,2),ppdim(15,6)
            Integer east(mats),west(mats),inte(neq),intw(neq),neq,ncol
            Integer mats,nsteps

            thmat(3)=1.0E-6
            thmat(4)=750.E-6

            Do i=1,neq
                  Do j=1,neq
                        Coeff(i,j)=0.
                        Tnodal(i,j)=0.
                  Enddo
            Enddo

            Do i=1,neq
                  Tinp(i)=0.
                  Tout(i)=0.
                  intw(i)=0.
```

```
                inte(i)=0.
                qoutlow(i)=0.
                qinupp(i)=0
                vinil(i)=0.
                minil(i)=0.
                Do j=1,nsteps+1
                        Temp(i,j)=0.
                        time(j)=0.
                        Einp(j)=0.
                        Liqlow(j)=0.
                        Liqupp(j)=0.
                        mremlow(j)=0.
                        mremupp(j)=0.
                        Emats(j)=0.
                        Effupp(j)=0.
                        Efflow(j)=0.
                        Efftot(j)=0.
                Enddo
        Enddo

        Do i=1,15
                Do j=1,6
                        pdim(i,j)=0.
                        ppdim(i,j)=0.
                        props(i,j)=0.
                        pprops(i,j)=0.
                Enddo
        Enddo

        Do i=1,neq
                Do j=1,2
                        nodmat(i,j)=0.
                        nodmati(i,j)=0.
                        conve(i,j)=0.
                Enddo
        Enddo

        Do i=1,nsteps+1
                Do j=1,mats+3
                        Est(i,j)=0.
                Enddo
        Enddo

        Do i=1,mats
                west(i)=0.
                east(i)=0.
        Enddo

        Do i=1,ncol
                r(i)=0.
        Enddo

        Do i=1,neq
                delth(i)=0
        Enddo
c       stop
        End


C       ----------------------------------------------------------------
C       STORES MATERIAL PROPERTIES AND NODAL DIMENSIONS IN A MATRIX
        Subroutine Matprops(rmat1,rmat2,rmat3,rmat4,rmat5,
     +rmat6,rmat7,rmat8,rmat9,rmat10,rmat11,rmat12,rmat13,rmat14,rmat15,
     +props,thmat,tthmat,
     +ndim1,ndim2,ndim3,ndim4,ndim5,ndim6,ndim7,ndim8,ndim9,ndim10,
     +ndim11,ndim12,ndim13,ndim14,ndim15,pdim)
        Double Precision rmat1(6),rmat2(6),rmat3(6),rmat4(6)
        Double Precision rmat5(6),rmat6(6),rmat7(6),rmat8(6),tthmat(15)
        Double Precision rmat9(6),rmat10(6),rmat11(6),rmat12(6),rmat13(6)
        Double Precision rmat14(6),rmat15(6),props(15,6),thmat(15)
```

```
      Integer ndim1(6),ndim2(6),ndim3(6),ndim4(6)
      Integer ndim5(6),ndim6(6),ndim7(6),ndim8(6)
      Integer ndim9(6),ndim10(6),ndim11(6),ndim12(6),ndim13(6)
      Integer ndim14(6),ndim15(6),pdim(15,6)

C     Stores material properties
      Do j=1,6
            props(1,j)=rmat1(j)
            pdim(1,j)=ndim1(j)
      Enddo
      Do j=1,6
            props(2,j)=rmat2(j)
            pdim(2,j)=ndim2(j)
      Enddo
      Do j=1,6
            props(3,j)=rmat3(j)
            pdim(3,j)=ndim3(j)
      Enddo
      Do j=1,6
            props(4,j)=rmat4(j)
            pdim(4,j)=ndim4(j)
      Enddo
      Do j=1,6
            props(5,j)=rmat5(j)
            pdim(5,j)=ndim5(j)
      Enddo
      Do j=1,6
            props(6,j)=rmat6(j)
            pdim(6,j)=ndim6(j)
      Enddo
      Do j=1,6
            props(7,j)=rmat7(j)
            pdim(7,j)=ndim7(j)
      Enddo
      Do j=1,6
            props(8,j)=rmat8(j)
            pdim(8,j)=ndim8(j)
      Enddo
      Do j=1,6
            props(9,j)=rmat9(j)
            pdim(9,j)=ndim9(j)
      Enddo
      Do j=1,6
            props(10,j)=rmat10(j)
            pdim(10,j)=ndim10(j)
      Enddo
      Do j=1,6
            props(11,j)=rmat11(j)
            pdim(11,j)=ndim11(j)
      Enddo
      Do j=1,6
            props(12,j)=rmat12(j)
            pdim(12,j)=ndim12(j)
      Enddo
      Do j=1,6
            props(13,j)=rmat13(j)
            pdim(13,j)=ndim13(j)
      Enddo
      Do j=1,6
            props(14,j)=rmat14(j)
            pdim(14,j)=ndim14(j)
      Enddo
      Do j=1,6
            props(15,j)=rmat15(j)
            pdim(15,j)=ndim15(j)
      Enddo
      Do j=1,15
            tthmat(j)=thmat(j)
      Enddo
      End
```

```
C       ---------------------------------------------------------------
C       STORES MATERIAL PROPERTIES AND NODAL DIMENSIONS IN A MATRIX
        Subroutine MMatprops(rrmat1,rrmat2,rrmat3,rrmat4,rrmat5,
     +rrmat6,rrmat7,rrmat8,rrmat9,rrmat10,rrmat11,rrmat12,rrmat13,
     +rrmat14,rrmat15,pprops,tthmat,
     +nndim1,nndim2,nndim3,nndim4,nndim5,nndim6,nndim7,
     +nndim8,nndim9,nndim10,nndim11,nndim12,nndim13,nndim14,nndim15,
     +ppdim)
        Double Precision rrmat1(6),rrmat2(6),rrmat3(6),rrmat4(6)
        Double Precision rrmat5(6),rrmat6(6),rrmat7(6),rrmat8(6)
        Double Precision rrmat9(6),rrmat10(6),rrmat11(6),rrmat12(6)
        Double Precision rrmat13(6),rrmat14(6),rrmat15(6),pprops(15,6)
        Double Precision tthmat(15)
        Integer nndim1(6),nndim2(6),nndim3(6),nndim4(6)
        Integer nndim5(6),nndim6(6),nndim7(6),nndim8(6)
        Integer nndim9(6),nndim10(6),nndim11(6),nndim12(6),nndim13(6)
        Integer nndim14(6),nndim15(6),ppdim(15,6)


        Do j=1,6
                pprops(1,j)=rrmat1(j)
                ppdim(1,j)=nndim1(j)
        Enddo
        Do j=1,6
                pprops(2,j)=rrmat2(j)
                ppdim(2,j)=nndim2(j)
        Enddo
        Do j=1,6
                pprops(3,j)=rrmat3(j)
                ppdim(3,j)=nndim3(j)
        Enddo
        Do j=1,6
                pprops(4,j)=rrmat4(j)
                ppdim(4,j)=nndim4(j)
        Enddo
        Do j=1,6
                pprops(5,j)=rrmat5(j)
                ppdim(5,j)=nndim5(j)
        Enddo
        Do j=1,6
                pprops(6,j)=rrmat6(j)
                ppdim(6,j)=nndim6(j)
        Enddo
        Do j=1,6
                pprops(7,j)=rrmat7(j)
                ppdim(7,j)=nndim7(j)
        Enddo
        Do j=1,6
                pprops(8,j)=rrmat8(j)
                ppdim(8,j)=nndim8(j)
        Enddo
        Do j=1,6
                pprops(9,j)=rrmat9(j)
                ppdim(9,j)=nndim9(j)
        Enddo
        Do j=1,6
                pprops(10,j)=rrmat10(j)
                ppdim(10,j)=nndim10(j)
        Enddo
        Do j=1,6
                pprops(11,j)=rrmat11(j)
                ppdim(11,j)=nndim11(j)
        Enddo
        Do j=1,6
                pprops(12,j)=rrmat12(j)
                ppdim(12,j)=nndim12(j)
        Enddo
        Do j=1,6
                pprops(13,j)=rrmat13(j)
```

```
                ppdim(13,j)=nndim13(j)
        Enddo
        Do j=1,6
                pprops(14,j)=rrmat14(j)
                ppdim(14,j)=nndim14(j)
        Enddo
        Do j=1,6
                pprops(15,j)=rrmat15(j)
                ppdim(15,j)=nndim15(j)
        Enddo


        End

C       ---------------------------------------------------------------
C       CALCULATES SPATIAL STEP SIZE AND DISTANCES
        Subroutine Distances(deltay,thmat,r,delr,distr,edgel,
   +pdim,mats,ncol,nangle,tthmat,deltayy,ppdim,neq,delth)
        Double Precision deltay(15),deltayy(15),thmat(15),tthmat(15)
        Double Precision delr,distr,edgel,radius,pi,r(ncol),delth(neq)
        Integer pdim(15,6),ppdim(15,6),mats,ncol,neq

C       Calculates the effective circular area of a square membrane
        pi=3.14159265
        distr=sqrt((edgel**2)/pi)                                              ![m^3]
        theta=2
        pii=pi/180

        delr=distr/(ncol-1)                                                        ![m]

        Do i=1,mats
                deltay(i)=thmat(i)/(pdim(i,3)-pdim(i,1))     ![m]
                deltayy(i)=tthmat(i)/(ppdim(i,3)-ppdim(i,1))
        Enddo

        radius=0.
        Do i=2,ncol
                radius=radius+delr
                r(i)=radius
        ![m]
        Enddo

        Do k=2,neq
        delth(k)=theta*pii/((nangle-2)-1)
        Enddo
        End


C       ---------------------------------------------------------------
C       ASSIGNS MATERIAL NUMBER TO THE NODE NUMBERS
        Subroutine Assign(pdim,nodmat,nodmati,nodint,mats,neq,nrow,ncol,
        +ppdim)
        Integer pdim(15,6),nodmat(neq,2),nodmati(neq,2),nodint(neq,2)
        Integer ppdim(15,6),mats,neq,ncol,nrow

        Do ii=1,mats
                Do i=pdim(ii,1)+1,pdim(ii,3)-1
                        Do j=pdim(ii,2)+1,pdim(ii,4)-1
                                Do k=pdim(ii,5),pdim(ii,6)
                                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                                nodmati(m,1)=m
                                nodmati(m,2)=ii
                                Enddo
                        Enddo
                Enddo
        Enddo

        Do ii=1,mats
                Do i=ppdim(ii,1)+1,ppdim(ii,3)-1
                        Do j=ppdim(ii,2)+1,ppdim(ii,4)-1
```

```
                        Do k=ppdim(ii,5),ppdim(ii,6)
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        nodmati(m,1)=m
                        nodmati(m,2)=ii
                        Enddo
                Enddo
        Enddo
Enddo

Do 450 ii=1,mats
        i=pdim(ii,1)
        Do j=pdim(ii,2),pdim(ii,4)
                Do k=pdim(ii,5),pdim(ii,6)
                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                nodmat(m,1)=m
                nodmat(m,2)=ii
                nodint(m,1)=m
                nodint(m,2)=1
                Enddo
        Enddo

        i=ppdim(ii,1)
        Do j=ppdim(ii,2),ppdim(ii,4)
                Do k=ppdim(ii,5),ppdim(ii,6)
                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                nodmat(m,1)=m
                nodmat(m,2)=ii
                nodint(m,1)=m
                nodint(m,2)=1
                Enddo
        Enddo


        i=pdim(ii,3)
        Do j=pdim(ii,2),pdim(ii,4)
                Do k=pdim(ii,5),pdim(ii,6)
                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                nodmat(m,1)=m
                nodmat(m,2)=ii
                nodint(m,1)=m
                nodint(m,2)=1
                Enddo
        Enddo

        i=ppdim(ii,3)
        Do j=ppdim(ii,2),ppdim(ii,4)
                Do k=ppdim(ii,5),ppdim(ii,6)
                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                nodmat(m,1)=m
                nodmat(m,2)=ii
                nodint(m,1)=m
                nodint(m,2)=1
                Enddo
        Enddo

        j=pdim(ii,2)
        Do i=pdim(ii,1),pdim(ii,3)
                Do k=pdim(ii,5),pdim(ii,6)
                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                nodmat(m,1)=m
                nodmat(m,2)=ii
                nodint(m,1)=m
                nodint(m,2)=2
                Enddo
        Enddo

        j=ppdim(ii,2)
        Do i=ppdim(ii,1),ppdim(ii,3)
                Do k=ppdim(ii,5),ppdim(ii,6)
                m=j+(i-1)*ncol+(k-1)*nrow*ncol
```

```
                        nodmat(m,1)=m
                        nodmat(m,2)=ii
                        nodint(m,1)=m
                        nodint(m,2)=2
                        Enddo
                Enddo

                j=pdim(ii,4)
                Do i=pdim(ii,1),pdim(ii,3)
                        Do k=pdim(ii,5),pdim(ii,6)
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        nodmat(m,1)=m
                        nodmat(m,2)=ii
                        nodint(m,1)=m
                        nodint(m,2)=2
                        Enddo
                Enddo

                j=ppdim(ii,4)
                Do i=ppdim(ii,1),ppdim(ii,3)
                        Do k=ppdim(ii,5),ppdim(ii,6)
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        nodmat(m,1)=m
                        nodmat(m,2)=ii
                        nodint(m,1)=m
                        nodint(m,2)=2
                        Enddo
                Enddo

450     Continue

        Do i=1,neq
                nodmat(i,1)=nodmat(i,1)+nodmati(i,1)
                nodmat(i,2)=nodmat(i,2)+nodmati(i,2)
        Enddo
        End


C       ------------------------------------------------------------------
C       FINDS INTERFACE NODES AT EAST AND WEST FACE
        Subroutine Faceint(pdim,east,west,inte,intw,mats,neq,ncol,nrow,
        +nangle,ppdim)
        Integer pdim(15,6),east(mats),west(mats),inte(neq),intw(neq)
        Integer mats,neq,ncol,nangle,nrow,ppdim(15,6)

        j=2             !West face of each material in pdim
        k=0             !Zeros to ensure proper beginning for counter
        Do i=1,mats
                If(pdim(i,j).EQ.1)Then
                        k=k+1
                        west(k)=i
                Endif
        Enddo
        j=1
        Do p=1,k-1
                Do kk=1,3
                i=pdim(west(p),3)
                m=j+(i-1)*ncol+(kk-1)*nrow*ncol
                intw(m)=m       !West interface nodes
                Enddo
        Enddo

        j=2             !West face of each material in pdim
        k=0             !Zeros to ensure proper beginning for counter
        Do i=1,mats
                If(ppdim(i,j).EQ.1)Then
                        k=k+1
                        west(k)=i
                Endif
        Enddo
```

```
      j=1
      Do p=1,k-1
            Do kk=4,(nangle-2)
            i=ppdim(west(p),3)
            m=j+(i-1)*ncol+(kk-1)*nrow*ncol
            intw(m)=m          !West interface nodes
            Enddo
      Enddo

      j=4                  !East face of each material in pdim
      k=0                  !Zeros to ensure proper beginning for counter
      Do i=1,mats
            If(pdim(i,j).EQ.ncol)Then
                  k=k+1
                  east(k)=i
            Endif
      Enddo

      j=ncol
      Do p=1,k-1
            Do kk=1,3
            i=pdim(east(p),3)
            m=j+(i-1)*ncol+(kk-1)*nrow*ncol
            inte(m)=m          !East interface nodes
            Enddo
      Enddo

      j=4                  !East face of each material in pdim
      k=0                  !Zeros to ensure proper beginning for counter
      Do i=1,mats
            If(ppdim(i,j).EQ.ncol)Then
                  k=k+1
                  east(k)=i
            Endif
      Enddo

      j=ncol
      Do p=1,k-1
            Do kk=4,(nangle-2)
            i=ppdim(east(p),3)
            m=j+(i-1)*ncol+(kk-1)*nrow*ncol
            inte(m)=m          !East interface nodes
            Enddo
      Enddo

      End


C     ------------------------------------------------------------------
C     ASSIGNS EFFECTIVE BC AT EAST FACE
      Subroutine EffeastBC(heffe,conve,pdim,neq,ncol,nrow,nangle,mats,
     +ppdim)
      Double Precision heffe(15),conve(neq,2)
      Integer pdim(15,6),mate(mats),neq,ncol,nrow,nangle,mats
      Integer ppdim(15,6)
      heffe(1)=135000
      heffe(2)=135000
      heffe(3)=0
      heffe(4)=0
      heffe(5)=0
      heffe(6)=135000
      heffe(7)=135000
      heffe(8)=0
      heffe(9)=0
      heffe(10)=0
      heffe(11)=0
      heffe(12)=0
      heffe(13)=0
      heffe(14)=0
```

```
      heffe(15)=0
      k=0                !Zeros to ensure proper beginning for counter
      j=4                !East face of each material in pdim
      Do i=1,mats
            If(pdim(i,j).EQ.ncol)Then
                  k=k+1
                  mate(k)=i
            Endif
      Enddo

      Do p=1,k
            ii=mate(p)
            j=ncol
            Do i=pdim(ii,1),pdim(ii,3)
                  Do kk=pdim(ii,5),pdim(ii,6)
                  m=j+(i-1)*ncol+(kk-1)*nrow*(nangle-2)
                  conve(m,1)=m
                  conve(m,2)=heffe(ii)
                  Enddo
            Enddo
      Enddo

      k=0                !Zeros to ensure proper beginning for counter
      j=4                !East face of each material in pdim
      Do i=1,mats
            If(ppdim(i,j).EQ.ncol)Then
                  k=k+1
                  mate(k)=i
            Endif
      Enddo

      Do p=1,k
            ii=mate(p)
            j=ncol
            Do i=ppdim(ii,1),ppdim(ii,3)
                  Do kk=ppdim(ii,5),ppdim(ii,6)
                  m=j+(i-1)*ncol+(kk-1)*nrow*(nangle-2)
                  conve(m,1)=m
                  conve(m,2)=heffe(ii)
                  Enddo
            Enddo
      Enddo
      End


C     ------------------------------------------------------------------
C     SETS INITIAL TIME AND TEMPERATURE OF NODES AND GAS PARAMETERS
      Subroutine Initial(Temp,Tavg,Tout,Ti,pressg,Tgasav,volg,
     +deflg,thmat,time,tstep,distr,mairini,
     +Pressavg,Tempavg,Tcav,Pcav,mgen,r,Tlvupp,Tlvlow,vcavi,Pvapi,
     +Pgage,Deflupp,Volupp,volcoeff,Vcoeff,edgel,hcell,
     +Vollow,Defllow,hcellupp,hcelllow,RTD,
     +delr,mol,ncol,neq,nsteps,mats)
      Double Precision Temp(neq,nsteps),Tavg(nsteps+1,mats),Tout(neq),Ti
      Double Precision pressg(nsteps+1),Tgasav(nsteps+1),volg(nsteps+1)
      Double Precision deflg(nsteps+1),time(nsteps+1),tstep,thmat(15)
      Double Precision volini,pi,distr,mairini
      Double Precision Pressavg(nsteps+1),Tempavg(nsteps+1)
      Double Precision Tcav(ncol,nsteps+1),Pcav(ncol,nsteps+1)
      Double Precision mgen(ncol,nsteps+1),r(ncol)
      Double Precision Tlvupp(ncol,nsteps+1),Tlvlow(ncol,nsteps+1)
      Double Precision vcavi,Pvapi,delr,mol
      Double Precision Pgage(nsteps+1),Deflupp(nsteps+1)
      Double Precision Volupp(nsteps+1),volcoeff,Vcoeff(9,2),edgel
      Double Precision hcell(ncol,nsteps+1),RTD(nsteps+1)
      Double Precision Vollow(nsteps+1),Defllow(nsteps+1)
      Double Precision hcellupp(ncol,nsteps+1),hcelllow(ncol,nsteps+1)
      Integer ncol
      Integer neq,nsteps
```

```
pi=3.14159265
Do i=1,neq
        Tout(i)=Ti
        Temp(i,1)=Ti
Enddo

Do i=1,16
        Tlvupp(i,1)=Ti
Enddo

Do i=1,16
        Tlvlow(i,1)=Ti
Enddo

Do i=1,nsteps+1
        Do j=1,mats
                Tavg(i,j)=Ti
        Enddo
Enddo

RTD(1)=Ti
Vcoeff(1,1)=2.E-3           !Membrane edge length                   [m]
Vcoeff(1,2)=2.007E-12       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(2,1)=3.E-3           !Membrane edge length                   [m]
Vcoeff(2,2)=4.528E-12       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(3,1)=4.E-3           !Membrane edge length                   [m]
Vcoeff(3,2)=8.173E-12       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(4,1)=5.E-3           !Membrane edge length                   [m]
Vcoeff(4,2)=1.258E-11       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(5,1)=6.E-3           !Membrane edge length                   [m]
Vcoeff(5,2)=1.811E-11       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(6,1)=7.E-3           !Membrane edge length                   [m]
Vcoeff(6,2)=2.471E-11       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(7,1)=8.E-3           !Membrane edge length                   [m]
Vcoeff(7,2)=3.220E-11       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(8,1)=9.E-3           !Membrane edge length                   [m]
Vcoeff(8,2)=4.075E-11       !Slope of Vol vs. Defl                  [m^3/um]
Vcoeff(9,1)=10.E-3          !Membrane edge length                   [m]
Vcoeff(9,2)=5.031E-11       !Slope of Vol vs. Defl                  [m^3/um]

Do i=1,9
        If(Vcoeff(i,1).EQ.edgel)Then
                volcoeff=Vcoeff(i,2)
        Endif
Enddo

pi=3.14159
volini=pi*thmat(4)*distr**2     !Initial volume of gas in cavity [m^3]
pressg(1)=1.46+101000.          !Initial pressure of gas in cavity          [abs Pa]
Tgasav(1)=Ti                    !Initial temperature of gas in cavity       [K]
volg(1)=volini                  !Initial volume of gas in cavity            [m^3]
deflg(1)=14.63                  !Initial deflection of membranes
[m]
mairini=(pressg(1)*volini*.02897)/(8.314*Ti)    ![kg]
vcavi=thmat(4)*pi*distr**2      ![m^3]
Pvapi=(1.101E-6*(Ti-273)**4+2.942E-5*(Ti-273)**3+
+.01648*(Ti-273)**2+.3891*(Ti-273)+8.966)*1000.
Pressavg(1)=Pvapi               !Initial pressure of the vapor      [abs Pa]
Pgage(1)=1.46                   !Initial gage pressure of the gas   [gage Pa]
Deflupp(1)=14.63                !Initial deflection of upper membrane   [um]
Defllow(1)=5.7                  !Initial deflection of lower membrane   [um]
Volupp(1)=volcoeff*14.63        !Initial volume of upper membrane   [m^3]
Vollow(1)=volcoeff*5.7          !Initial volume of upper membrane   [m^3]
Tempavg(1)=Ti                   ![K]

ii=0
Do i=2,17
        ii=ii+1
        Tcav(ii,1)=Ti           ![K]
        Pcav(ii,1)=Pvapi        ![abs Pa]
```

```
      Enddo

      ii=0
      Do j=2,17
            ii=ii+1
            mgen(ii,1)=(Pvapi*2.*pi*r(j)*delr*thmat(4)*mol)/
     +      (8.314*Ti)
      Enddo

      ii=0
      hcellupp(ii,1)=0.
      hcelllow(ii,1)=0.
      Do j=2,17
            ii=ii+1
            hcell(ii,1)=thmat(4)
      Enddo
      End


C     ---------------------------------------------------------------------
C     SETS INITIAL LIQUID LAYER VOLUMES FOR EVAPORATION
      Subroutine Liqvols(vinil,minil,viniu,miniu,miniltot,miniutot,
      +mremlow,mremupp,Liqlow,Liqupp,thmat,deltayy,
      +r,delr,rhol,deltay,pdim,ii7,ii9,tthmat,
     +mats,matll,nsteps,neq,ncol,ppdim)
      Double Precision vinil(neq),minil(neq),viniu(neq),miniu(neq)
      Double Precision miniltot,miniutot,r(ncol),delr,rhol,deltay(mats)
      Double Precision mremlow(nsteps+1),mremupp(nsteps+1),deltayy(mats)
      Double Precision Liqlow(nsteps+1),Liqupp(nsteps+1),thmat(15)
      Double Precision tthmat(15)
      Integer pdim(15,6),ii7,ii9,mats,matll,ppdim(15,6)
      Integer nsteps,neq,ncol

      pi=3.14159265
      sum=0.
     ii7=0
      Do j=2,17
            ii7=ii7+1
            Do i=1,mats
                  If(pdim(i,1).EQ.12)Then
                  kk=i
                  Am=pi*((r(j)+delr/2)**2-(r(j)-delr/2)**2)
                  vinil(ii7)=Am*(pdim(matll,3)-pdim(matll,1))*deltay(kk)
                  minil(ii7)=rhol*vinil(ii7)
                  Endif
            Enddo
            sum=sum+minil(ii7)
      Enddo
      miniltot=sum
      mremlow(1)=miniltot
      Liqlow(1)=thmat(matll)
      sum=0.
     ii9=0
      Do j=2,17
            ii9=ii9+1
            Do i=1,mats
                  If(pdim(i,3).EQ.7)Then
                  kk=i
                  Am=pi*((r(j)+delr/2)**2-(r(j)-delr/2)**2)
                  viniu(ii9)=Am*(pdim(3,3)-pdim(3,1))*deltay(kk)
                  miniu(ii9)=rhol*viniu(ii9)
                  Endif
            Enddo
            sum=sum+miniu(ii9)
      Enddo
      miniutot=sum
      mremupp(1)=miniutot
      Liqupp(1)=thmat(3)

      sum=0.
```

```
      ii9=0
            Do j=2,17
                  ii9=ii9+1
                  Do i=1,mats
                        If(ppdim(i,3).EQ.7)Then
                        kk=i
                        Am=pi*((r(j)+delr/2)**2-(r(j)-delr/2)**2)
                        viniu(ii9)=Am*(ppdim(3,3)-ppdim(3,1))*deltayy(kk)
                        miniu(ii9)=rhol*viniu(ii9)
                        Endif
                  Enddo
                  sum=sum+miniu(ii9)
            Enddo
            miniutot=sum
            mremupp(1)=miniutot
            Liqupp(1)=tthmat(3)

            End


C           ------------------------------------------------------------------------
            Subroutine TempBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
            +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
     +pdim,nodmati,nodint,mats,Coe,T1,T2)
            Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
            Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
            Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396)
            Double Precision hnwy,hnex,hney,hswy,hsex,hsey
            Double Precision delr,deltay(15),delt,r(ncol),props(15,6)
            Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
            Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
            Integer Nfix(48),inte(neq),intw(neq),nflux
            Integer BCn,BCs,BCe
            Integer adialow,adiabats
            Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
            Integer pdim(15,6),nodmati(neq,2),nangle
            Integer mats,nodint(neq,2)
            Parameter (Ipath=1)

            Call CornerBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe)

            Call FaceBC(Coeff,Tinp,Tout,conve,Qfix,Tfix,
            +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
            +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe)

            Call Interior(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe)

            Do i=1,396
            T1(i)=Tinp(i)
            Enddo
            Call DLSARG(396,Coe,396,T1,Ipath,T2)
            Do i=1,396
            m=i
            Tout(m)=T2(i)
            Enddo
            END

            Subroutine TTempBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
            +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
```

```
     +pdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,qoutlow)
           Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
           Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
           Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396)
           Double Precision hnwy,hnex,hney,hswy,hsex,hsey
           Double Precision delr,deltay(15),delt,r(ncol),props(15,6)
           Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
           Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
           Double Precision qinupp(neq),qoutlow(neq)
           Integer Nfix(48),inte(neq),intw(neq),nflux
           Integer BCn,BCs,BCe
           Integer adialow,adiabats
           Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
           Integer pdim(15,6),nodmati(neq,2),nangle
           Integer mats,nodint(neq,2)
           Parameter (Ipath=1)

           Call CornerBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe)

           Call FaceBC(Coeff,Tinp,Tout,conve,Qfix,Tfix,
           +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
           +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe)

           Call Interior(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe)

      Call Intinterf(Tinp,Tout,props,deltay,delt,
     +nodmat,qinupp,qoutlow,ncol,nrow,neq)

           Do i=1,396
           T1(i)=Tinp(i)
           Enddo
           Call DLSARG(396,Coe,396,T1,Ipath,T2)
           Do i=1,396
           m=i
           Tout(m)=T2(i)
           Enddo
           END

C          ------------------------------------------------------------------
           Subroutine TempBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
           +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
     +pdim,nodmati,nodint,mats,Coe,T1,T2,delth)
           Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
           Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
           Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
           Double Precision hnwy,hnex,hney,hswy,hsex,hsey
           Double Precision delr,deltay(15),delt,r(ncol),props(15,6)
           Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
           Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
           Integer Nfix(48),inte(neq),intw(neq),nflux
           Integer BCn,BCs,BCe
           Integer adialow,adiabats
           Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
           Integer pdim(15,6),nodmati(neq,2),nangle
           Integer mats,nodint(neq,2)
           Parameter (Ipath=1)

           Call CornerBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)
```

```
      Call FaceBC2(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe,
     +delth)

      Call Interior2(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)

      Do i=1,396
      m=i+396
      T1(i)=Tinp(m)
      Enddo

      Call DLSARG(396,Coe,396,T1,Ipath,T2)
      Do i=1,396
      m=i+396
      Tout(m)=T2(i)
      Enddo
      END

      Subroutine TTempBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
     +pdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,qoutlow,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltay(15),delt,r(ncol),props(15,6)
      Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
      Double Precision qinupp(neq),qoutlow(neq)
      Integer Nfix(48),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe
      Integer adialow,adiabats
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
      Integer pdim(15,6),nodmati(neq,2),nangle
      Integer mats,nodint(neq,2)
      Parameter (Ipath=1)

      Call CornerBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)

      Call FaceBC2(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe,
     +delth)

      Call Interior2(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)

      Do i=1,396
      m=i+396
      T1(i)=Tinp(m)
      Enddo
      Call DLSARG(396,Coe,396,T1,Ipath,T2)
      Do i=1,396
      m=i+396
      Tout(m)=T2(i)
      Enddo
      END

C     ------------------------------------------------------------------------
      Subroutine TempBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
```

```
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
      +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
+nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
+pdim,nodmati,nodint,mats,Coe,T1,T2,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltay(15),delt,r(ncol),props(15,6)
      Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
      Integer Nfix(48),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe
      Integer adialow,adiabats
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
      Integer pdim(15,6),nodmati(neq,2),nangle
      Integer mats,nodint(neq,2)
      Parameter (Ipath=1)

      Call CornerBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
+nrow,ncol,neq,Coe,delth)

      Call FaceBC3(Coeff,Tinp,Tout,conve,Qfix,Tfix,
      +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
      +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe,
+delth)

      Call Interior3(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
+pdim,nodmati,nodmat,nodint,adialow,adiabats,
+nrow,ncol,neq,mats,Coe,delth)

      Do i=1,396
      m=i+792
      T1(i)=Tinp(m)
      Enddo
      Call DLSARG(396,Coe,396,T1,Ipath,T2)
      Do i=1,396
      m=i+792
      Tout(m)=T2(i)
      Enddo
      END

      Subroutine TTempBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
      +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
+nflux,BCn,BCs,BCe,adialow,adiabats,nangle,
+pdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,qoutlow,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltay(15),delt,r(ncol),props(15,6)
      Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
      Double Precision qinupp(neq),qoutlow(neq)
      Integer Nfix(48),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe
      Integer adialow,adiabats
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
      Integer pdim(15,6),nodmati(neq,2),nangle
      Integer mats,nodint(neq,2)
      Parameter (Ipath=1)

      Call CornerBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
```

```
     +nrow,ncol,neq,Coe,delth)

           Call FaceBC3(Coeff,Tinp,Tout,conve,Qfix,Tfix,
           +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
           +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe,
     +delth)

           Call Interior3(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)

           Do i=1,396
           m=i+792
           T1(i)=Tinp(m)
           Enddo
           Call DLSARG(396,Coe,396,T1,Ipath,T2)
           Do i=1,396
           m=i+792
           Tout(m)=T2(i)
           Enddo
           END

c          4th layer
C          ------------------------------------------------------------------------
           Subroutine TempBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
           +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adiabats,nangle,
     +ppdim,nodmati,nodint,mats,Coe,T1,T2,delth)
           Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
           Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
           Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
           Double Precision hnwy,hnex,hney,hswy,hsex,hsey
           Double Precision delr,deltayy(15),delt,r(ncol),pprops(15,6)
           Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
           Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
           Integer Nfix(48),inte(neq),intw(neq),nflux
           Integer BCn,BCs,BCe
           Integer adiabats
           Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
           Integer ppdim(15,6),nodmati(neq,2),nangle
           Integer mats,nodint(neq,2)
           Parameter (Ipath=1)

           Call CornerBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)

           Call FaceBC4(Coeff,Tinp,Tout,conve,Qfix,Tfix,
           +r,pprops,Tn,Ts,Te,hn,hs,delr,deltayy,delt,Nfix,nodmat,inte,intw,
           +nflux,BCn,BCs,BCe,adiabats,ncol,nrow,neq,Coe,delth)

           Call Interior4(Coeff,Tinp,Tout,pprops,r,delr,deltayy,delt,
     +ppdim,nodmati,nodmat,nodint,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)

           Do i=1,396
           m=i+1188
           T1(i)=Tinp(m)
           Enddo
           Call DLSARG(396,Coe,396,T1,Ipath,T2)
           Do i=1,396
           m=i+1188
           Tout(m)=T2(i)
           Enddo
           END

           Subroutine TTempBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
```

```
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adiabats,nangle,
     +ppdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,delth)
           Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
           Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
           Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
           Double Precision hnwy,hnex,hney,hswy,hsex,hsey
           Double Precision delr,deltayy(15),delt,r(ncol),pprops(15,6)
           Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
           Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
           Double Precision qinupp(neq)
           Integer Nfix(48),inte(neq),intw(neq),nflux
           Integer BCn,BCs,BCe
           Integer adiabats
           Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
           Integer ppdim(15,6),nodmati(neq,2),nangle
           Integer mats,nodint(neq,2)
           Parameter (Ipath=1)

           Call CornerBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)

           Call FaceBC4(Coeff,Tinp,Tout,conve,Qfix,Tfix,
           +r,pprops,Tn,Ts,Te,hn,hs,delr,deltayy,delt,Nfix,nodmat,inte,intw,
           +nflux,BCn,BCs,BCe,adiabats,ncol,nrow,neq,Coe,delth)

           Call Interior4(Coeff,Tinp,Tout,pprops,r,delr,deltayy,delt,
     +ppdim,nodmati,nodmat,nodint,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)
           Do i=1,396
           m=i+1188
           T1(i)=Tinp(m)
           Enddo
           Call DLSARG(396,Coe,396,T1,Ipath,T2)
           Do i=1,396
           m=i+1188
           Tout(m)=T2(i)
           Enddo
           END

C          ------------------------------------------------------------------------
           Subroutine TempBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
           +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
     +nflux,BCn,BCs,BCe,adiabats,nangle,
     +ppdim,nodmati,nodint,mats,Coe,T1,T2,delth)
           Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
           Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
           Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
           Double Precision hnwy,hnex,hney,hswy,hsex,hsey
           Double Precision delr,deltayy(15),delt,r(ncol),pprops(15,6)
           Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
           Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
           Integer Nfix(48),inte(neq),intw(neq),nflux
           Integer BCn,BCs,BCe
           Integer adiabats
           Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
           Integer ppdim(15,6),nodmati(neq,2),nangle
           Integer mats,nodint(neq,2)
           Parameter (Ipath=1)

           Call CornerBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)
```

```
      Call FaceBC5(Coeff,Tinp,Tout,conve,Qfix,Tfix,
      +r,pprops,Tn,Ts,Te,hn,hs,delr,deltayy,delt,Nfix,nodmat,inte,intw,
      +nflux,BCn,BCs,BCe,adiabats,ncol,nrow,neq,Coe,delth)

      Call Interior5(Coeff,Tinp,Tout,pprops,r,delr,deltayy,delt,
+ppdim,nodmati,nodmat,nodint,adiabats,
+nrow,ncol,neq,mats,Coe,delth)

      Do i=1,396
      m=i+1584
      T1(i)=Tinp(m)
      Enddo
      Call DLSARG(396,Coe,396,T1,Ipath,T2)
      Do i=1,396
      m=i+1584
      Tout(m)=T2(i)
      Enddo
      END

      Subroutine TTempBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
      +nrow,ncol,neq,conve,Qfix,Tfix,Tn,Ts,Te,hn,hs,Nfix,inte,intw,
+nflux,BCn,BCs,BCe,adiabats,nangle,
+ppdim,nodmati,nodint,mats,Coe,T1,T2,qinupp,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltayy(15),delt,r(ncol),pprops(15,6)
      Double Precision Qfix(48),Tfix(10),T1(396),T2(396)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2)
      Double Precision qinupp(neq)
      Integer Nfix(48),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe
      Integer adiabats
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
      Integer ppdim(15,6),nodmati(neq,2),nangle
      Integer mats,nodint(neq,2)
      Parameter (Ipath=1)

      Call CornerBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
+nrow,ncol,neq,Coe,delth)

      Call FaceBC5(Coeff,Tinp,Tout,conve,Qfix,Tfix,
      +r,pprops,Tn,Ts,Te,hn,hs,delr,deltayy,delt,Nfix,nodmat,inte,intw,
      +nflux,BCn,BCs,BCe,adiabats,ncol,nrow,neq,Coe,delth)

      Call Interior5(Coeff,Tinp,Tout,pprops,r,delr,deltayy,delt,
+ppdim,nodmati,nodmat,nodint,adiabats,
+nrow,ncol,neq,mats,Coe,delth)

      Do i=1,396
      m=i+1584
      T1(i)=Tinp(m)
      Enddo
      Call DLSARG(396,Coe,396,T1,Ipath,T2)
      Do i=1,396
      m=i+1584
      Tout(m)=T2(i)
      Enddo
      END

      Subroutine CornerBC(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
      +nrow,ncol,neq,Coe)
```

```
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltay(15),delt,r(ncol),props(15,4)
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq

      If (BCnw.eq.1) Then
            Call BCnwC(Coeff,Tinp,Tout,props,Tnwy,hnwy,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe)

      Else
      Endif

      If (BCnw.eq.2) Then
            Call BCnwT(Coeff,Tinp,Tnw,ncol,neq)
      Else
      Endif

      If (BCne.eq.1) Then
            Call BCneC(Coeff,Tinp,Tout,r,props,Tnex,Tney,hnex,hney,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe)
      Else
      Endif

      If (BCne.eq.2) Then
            Call BCneT(Coeff,Tinp,Tne,ncol,neq)
      Else
      Endif

      If (BCsw.eq.1) Then
            Call BCswC(Coeff,Tinp,Tout,props,Tswy,hswy,
     +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe)
      Else
      Endif

      If (BCsw.eq.2) Then
            Call BCswT(Coeff,Tinp,Tsw,nrow,ncol,neq)
      Else
      Endif
      If (BCse.eq.1) Then
            Call BCseC(Coeff,Tinp,Tout,r,props,Tsex,Tsey,hsex,hsey,
     +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe)

      Else
      Endif

      If (BCse.eq.2) Then
            Call BCseT(Coeff,Tinp,Tse,nrow,ncol,neq)
      Else
      Endif
      End


C     ------------------------------------------------------------------------
C     SETS TYPE OF BOUNDARY CONDIITONS AT FACES
      Subroutine FaceBC(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Qfix(10),Tfix(10),props(15,4),r(ncol)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2),Coe(396,396)
      Double Precision delr,deltay(15),delt
      Integer Nfix(10),nodmat(neq,2),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe,ncol,nrow,neq
      Integer adialow,adiabats
      If (BCn.eq.1) Then
            Call BCnorthC(Coeff,Tinp,Tout,r,props,Tn,hn,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe)
      Else
```

```
      Endif

      If (BCn.eq.2) Then
            Call BCnorthT(Coeff,Tinp,Tn,ncol,neq)
      Else
      Endif

      If (BCs.eq.1) Then
            Call BCsouthC(Coeff,Tinp,Tout,r,props,Qfix,Tfix,Ts,hs,
     +delr,deltay,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe)

      Else
      Endif

      If (BCs.eq.2) Then
            Call BCsouthT(Coeff,Tinp,Ts,nrow,ncol,neq)
      Else
      Endif

      If (BCe.eq.1) Then
            Call BCeastC(Coeff,Tinp,Tout,r,props,conve,Te,
     +delr,deltay,delt,nodmat,inte,adialow,adiabats,
+nrow,ncol,neq,Coe)
      Else
      Endif

      If (BCe.eq.2) Then
            Call BCeastT(Coeff,Tinp,Te,nrow,ncol,neq)
      Else
      Endif

            Call BCwestC(Coeff,Tinp,Tout,props,
     +delr,deltay,delt,nodmat,intw,adialow,adiabats,
+nrow,ncol,neq,Coe)
      End


      Subroutine CornerBC2(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
+Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
+delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltay(15),delt,r(ncol),props(15,4)
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq

      If (BCnw.eq.1) Then
            Call BCnwC2(Coeff,Tinp,Tout,props,Tnwy,hnwy,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCne.eq.1) Then
            Call BCneC2(Coeff,Tinp,Tout,r,props,Tnex,Tney,hnex,hney,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCsw.eq.1) Then
            Call BCswC2(Coeff,Tinp,Tout,props,Tswy,hswy,
     +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Else
      Endif

      If (BCse.eq.1) Then
            Call BCseC2(Coeff,Tinp,Tout,r,props,Tsex,Tsey,hsex,hsey,
```

```
     +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)

      Else
      Endif
      End

C     ------------------------------------------------------------------------
C     SETS TYPE OF BOUNDARY CONDIITONS AT FACES
      Subroutine FaceBC2(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe,
     +delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Qfix(10),Tfix(10),props(15,4),r(ncol)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2),Coe(396,396)
      Double Precision delr,deltay(15),delt,delth(neq)
      Integer Nfix(10),nodmat(neq,2),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe,ncol,nrow,neq
      Integer adialow,adiabats

      If (BCn.eq.1) Then
            Call BCnorthC2(Coeff,Tinp,Tout,r,props,Tn,hn,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Else
      Endif

      If (BCs.eq.1) Then
            Call BCsouthC2(Coeff,Tinp,Tout,r,props,Qfix,Tfix,Ts,hs,
     +delr,deltay,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)

      Else
      Endif

      If (BCe.eq.1) Then
            Call BCeastC2(Coeff,Tinp,Tout,r,props,conve,Te,
     +delr,deltay,delt,nodmat,inte,adialow,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Else
      Endif
            Call BCwestC2(Coeff,Tinp,Tout,props,
     +delr,deltay,delt,nodmat,intw,adialow,adiabats,
     +nrow,ncol,neq,Coe,delth)
      End

      Subroutine CornerBC3(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltay,delt,r,props,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396),delth(neq)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey
      Double Precision delr,deltay(15),delt,r(ncol),props(15,4)
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq

      If (BCnw.eq.1) Then
            Call BCnwC3(Coeff,Tinp,Tout,props,Tnwy,hnwy,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCne.eq.1) Then
            Call BCneC3(Coeff,Tinp,Tout,r,props,Tnex,Tney,hnex,hney,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCsw.eq.1) Then
```

```
                Call BCswC3(Coeff,Tinp,Tout,props,Tswy,hswy,
     +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)
     Else
     Endif

     If (BCse.eq.1) Then
                Call BCseC3(Coeff,Tinp,Tout,r,props,Tsex,Tsey,hsex,hsey,
     +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)

     Else
     Endif
     End

C    ---------------------------------------------------------------------
C    SETS TYPE OF BOUNDARY CONDIITONS AT FACES
     Subroutine FaceBC3(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,props,Tn,Ts,Te,hn,hs,delr,deltay,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adialow,adiabats,ncol,nrow,neq,Coe,
    +delth)
     Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision Qfix(10),Tfix(10),props(15,4),r(ncol)
     Double Precision Tn,Ts,Te,hn,hs,conve(neq,2),Coe(396,396)
     Double Precision delr,deltay(15),delt,delth(neq)
     Integer Nfix(10),nodmat(neq,2),inte(neq),intw(neq),nflux
     Integer BCn,BCs,BCe,ncol,nrow,neq
     Integer adialow,adiabats

     If (BCn.eq.1) Then
                Call BCnorthC3(Coeff,Tinp,Tout,r,props,Tn,hn,
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
     Else
     Endif

     If (BCs.eq.1) Then
                Call BCsouthC3(Coeff,Tinp,Tout,r,props,Qfix,Tfix,Ts,hs,
     +delr,deltay,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)

     Else
     Endif

     If (BCe.eq.1) Then
                Call BCeastC3(Coeff,Tinp,Tout,r,props,conve,Te,
     +delr,deltay,delt,nodmat,inte,adialow,adiabats,
    +nrow,ncol,neq,Coe,delth)
     Else
     Endif

                Call BCwestC3(Coeff,Tinp,Tout,props,
     +delr,deltay,delt,nodmat,intw,adialow,adiabats,
    +nrow,ncol,neq,Coe,delth)
     End

     Subroutine CornerBC4(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
    +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
    +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)
     Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey,delth(neq)
     Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396)
     Double Precision hnwy,hnex,hney,hswy,hsex,hsey
     Double Precision delr,deltayy(15),delt,r(ncol),pprops(15,4)
     Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq

     If (BCnw.eq.1) Then
                Call BCnwC4(Coeff,Tinp,Tout,pprops,Tnwy,hnwy,
     +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)

     Else
     Endif
```

```
      If (BCne.eq.1) Then
            Call BCneC4(Coeff,Tinp,Tout,r,pprops,Tnex,Tney,hnex,hney,
     +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCsw.eq.1) Then
            Call BCswC4(Coeff,Tinp,Tout,pprops,Tswy,hswy,
     +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Else
      Endif

      If (BCse.eq.1) Then
            Call BCseC4(Coeff,Tinp,Tout,r,pprops,Tsex,Tsey,hsex,hsey,
     +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)

      Else
      Endif
      End

C     ------------------------------------------------------------------------
C     SETS TYPE OF BOUNDARY CONDIITONS AT FACES
      Subroutine FaceBC4(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,pprops,Tn,Ts,Te,hn,hs,delr,deltayy,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adiabats,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Qfix(10),Tfix(10),pprops(15,4),r(ncol)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2),Coe(396,396)
      Double Precision delr,deltayy(15),delt,delth(neq)
      Integer Nfix(10),nodmat(neq,2),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe,ncol,nrow,neq
      Integer adiabats

      If (BCn.eq.1) Then
            Call BCnorthC4(Coeff,Tinp,Tout,r,pprops,Tn,hn,
     +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Else
      Endif

      If (BCs.eq.1) Then
            Call BCsouthC4(Coeff,Tinp,Tout,r,pprops,Qfix,Tfix,Ts,hs,
     +delr,deltayy,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)

      Else
      Endif

      If (BCe.eq.1) Then
            Call BCeastC4(Coeff,Tinp,Tout,r,pprops,conve,Te,
     +delr,deltayy,delt,nodmat,inte,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Else
      Endif

            Call BCwestC4(Coeff,Tinp,Tout,pprops,
     +delr,deltayy,delt,nodmat,intw,adiabats,
     +nrow,ncol,neq,Coe,delth)
      End

C     SETS TYPE OF BOUNDARY CONDITION AT CORNERS
      Subroutine CornerBC5(Coeff,Tinp,Tout,Tnwy,Tnex,Tney,Tswy,
     +Tsex,Tsey,Tnw,Tne,Tsw,Tse,hnwy,hnex,hney,hswy,hsex,hsey,
     +delr,deltayy,delt,r,pprops,nodmat,BCnw,BCne,BCsw,BCse,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Tnwy,Tnex,Tney,Tswy,Tsex,Tsey
      Double Precision Tnw,Tne,Tsw,Tse,Coe(396,396)
      Double Precision hnwy,hnex,hney,hswy,hsex,hsey,delth(neq)
      Double Precision delr,deltayy(15),delt,r(ncol),pprops(15,4)
      Integer nodmat(neq,2),BCnw,BCne,BCsw,BCse,nrow,ncol,neq
```

```
      If (BCnw.eq.1) Then
            Call BCnwC5(Coeff,Tinp,Tout,pprops,Tnwy,hnwy,
     +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCne.eq.1) Then
            Call BCneC5(Coeff,Tinp,Tout,r,pprops,Tnex,Tney,hnex,hney,
     +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)

      Else
      Endif

      If (BCsw.eq.1) Then
            Call BCswC5(Coeff,Tinp,Tout,pprops,Tswy,hswy,
     +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Else
      Endif

      If (BCse.eq.1) Then
            Call BCseC5(Coeff,Tinp,Tout,r,pprops,Tsex,Tsey,hsex,hsey,
     +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)

      Else
      Endif
      End

C     ------------------------------------------------------------------------
C     SETS TYPE OF BOUNDARY CONDIITONS AT FACES
      Subroutine FaceBC5(Coeff,Tinp,Tout,conve,Qfix,Tfix,
     +r,pprops,Tn,Ts,Te,hn,hs,delr,deltayy,delt,Nfix,nodmat,inte,intw,
     +nflux,BCn,BCs,BCe,adiabats,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision Qfix(10),Tfix(10),pprops(15,4),r(ncol)
      Double Precision Tn,Ts,Te,hn,hs,conve(neq,2),Coe(396,396)
      Double Precision delr,deltayy(15),delt,delth(neq)
      Integer Nfix(10),nodmat(neq,2),inte(neq),intw(neq),nflux
      Integer BCn,BCs,BCe,ncol,nrow,neq
      Integer adiabats

      If (BCn.eq.1) Then
            Call BCnorthC5(Coeff,Tinp,Tout,r,pprops,Tn,hn,
     +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Else
      Endif

      If (BCs.eq.1) Then
            Call BCsouthC5(Coeff,Tinp,Tout,r,pprops,Qfix,Tfix,Ts,hs,
     +delr,deltayy,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)

      Else
      Endif

      If (BCe.eq.1) Then
            Call BCeastC5(Coeff,Tinp,Tout,r,pprops,conve,Te,
     +delr,deltayy,delt,nodmat,inte,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Else
      Endif

            Call BCwestC5(Coeff,Tinp,Tout,pprops,
     +delr,deltayy,delt,nodmat,intw,adiabats,
     +nrow,ncol,neq,Coe,delth)
      End

C     ------------------------------------------------------------------------
C     SETS ACTUAL BOUNDARY CONDITIONS AT CORNER NODES
      Subroutine BCnwC(Coeff,Tinp,Tout,props,Tnwy,hnwy,
```

```
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision props(15,4),Tnwy,hnwy,Coe(396,396)
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D
      Integer nodmat(neq,2),ncol,neq,nrow

      i=1
      j=1
      k=1
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number

      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
      B=(4*props(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
      D=(2*hnwy*delt)/(deltay(nodmat(m,2)))

     Tinp(m)=Tout(m)+(D/A)*Tnwy
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A

      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End


C     Northwest Corner Temperature BC
      Subroutine BCnwT(Coeff,Tinp,Tnw,ncol,neq)
      Double Precision Coeff(neq,neq),Tinp(neq),Tnw
      Integer ncol,neq

      i=1
      j=1
      m=j+(i-1)*ncol !This gives the equation number
      Tinp(m)=Tnw
      Coeff(m,j+(i-1)*ncol)=1
      End


C     Northeast Corner Convection BC
      Subroutine BCneC(Coeff,Tinp,Tout,r,props,Tnex,Tney,hnex,hney,
      +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision r(ncol),props(15,4),Tnex,Tney,hnex,hney
      Double Precision delr,deltay(15),delt,Coe(396,396)
      Double Precision A,B,C,D,E
      Integer nodmat(neq,2),ncol,neq,nrow

      i=1
      j=ncol
      k=1
      m=j+(i-1)*ncol+(k-1)*nrow*ncol

C     Coefficients from finite difference equation
      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
    +  (deltay(nodmat(m,2))**2)
      D=(2*hney*delt*(r(j)-delr/4))/(deltay(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)

      Tinp(m)=Tout(m)+(D/A)*Tney+(E/A)*Tnex
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
```

```
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A

      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCneT(Coeff,Tinp,Tne,ncol,neq)
      Double Precision Coeff(neq,neq),Tinp(neq),Tne
      Integer ncol,neq

      i=1
      j=ncol
      m=j+(i-1)*ncol !This gives the equation number
      Tinp(m)=Tne
      Coeff(m,j+(i-1)*ncol)=1
      End

      Subroutine BCswC(Coeff,Tinp,Tout,props,Tswy,hswy,
      +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision props(15,4),Tswy,hswy
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq

i=nrow
      j=1
      k=1
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number

      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
      B=(4*props(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
      D=(2*hswy*delt)/(deltay(nodmat(m,2)))

      Tinp(m)=Tout(m)+(D/A)*Tswy
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A

      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A


      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCswT(Coeff,Tinp,Tsw,nrow,ncol,neq)
      Double Precision Coeff(neq,neq),Tinp(neq),Tsw
      Integer nrow,ncol,neq

      i=nrow
      j=1
      m=j+(i-1)*ncol !This gives the equation number
      Tinp(m)=Tsw
      Coeff(m,j+(i-1)*ncol)=1
      End
```

```
      Subroutine BCseC(Coeff,Tinp,Tout,r,props,Tsex,Tsey,hsex,hsey,
      +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
 Double Precision r(ncol),props(15,4),Tsex,Tsey,hsex,hsey
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq

      i=nrow
      j=ncol
      k=1
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number

      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltay(nodmat(m,2))**2)
      D=(2*hsey*delt*(r(j)-delr/4))/(deltay(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)

      Tinp(m)=Tout(m)+(D/A)*Tsey+(E/A)*Tsex
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A

      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
 End
      Subroutine BCseT(Coeff,Tinp,Tse,nrow,ncol,neq)
      Double Precision Coeff(neq,neq),Tinp(neq),Tse
      Integer nrow,ncol,neq

      i=nrow
      j=ncol
      m=j+(i-1)*ncol !This gives the equation number
      Tinp(m)=Tse
      Coeff(m,j+(i-1)*ncol)=1
      End

C     ------------------------------------------------------------------------
C     SETS ACTUAL BOUNDARY CONDITIONS AT FACE NODES

      Subroutine BCnorthC(Coeff,Tinp,Tout,r,props,Tn,hn,
      +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),Tn,hn,delr,deltay(15),delt
      Double Precision A,B,C,D,E,Coe(396,396)
      Integer nodmat(neq,2),ncol,nrow,neq
      hn=0
      Do 1100 j=2,ncol-1
            i=1
            k=1
            m=j+(i-1)*ncol+(k-1)*nrow*ncol        !This gives the equation number

            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
            B=(props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +        (r(j)*delr**2)
            C=(props(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +        (r(j)*delr**2)
            D=(2*props(nodmat(m,2),1)*delt)/
     +        (deltay(nodmat(m,2))**2)
            E=(2*hn*delt)/(deltay(nodmat(m,2)))

            Tinp(m)=Tout(m)+(E/A)*Tn
```

```
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A

              Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

              Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

              Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)

              Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
1100          Continue
              End

              Subroutine BCnorthT(Coeff,Tinp,Tn,ncol,neq)
              Double Precision Coeff(neq,neq),Tinp(neq),Tn
              Integer ncol,neq

              Do j=2,ncol-1
                i=1
                m=j+(i-1)*ncol          !This gives the equation number
                Tinp(m)=Tn
                Coeff(m,j+(i-1)*ncol)=1
              Enddo
              End

              Subroutine BCsouthC(Coeff,Tinp,Tout,r,props,Qfix,Tfix,Ts,hs,
              +delr,deltay,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe)
              Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
              Double Precision r(ncol),props(15,4),Qfix(10),Tfix(10),Ts,hs
              Double Precision delr,deltay(15),delt
              Double Precision A,B,C,D,E,Coe(396,396)
              Integer nodmat(neq,2),Nfix(10),nflux,nrow,ncol,neq

              n=1
              Do 1300 j=2,ncol-1
                i=nrow
                k=1
                m=j+(i-1)*ncol+(k-1)*nrow*ncol              !This gives the equation number

                A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
                B=(props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
              +   (r(j)*delr**2)
                C=(props(nodmat(m,2),1)*delt*(r(j)+delr/2))/
              +   (r(j)*delr**2)
                D=(2*props(nodmat(m,2),1)*delt)/
              +   (deltay(nodmat(m,2))**2)
                E=(2*hs*delt)/(deltay(nodmat(m,2)))

                Tinp(m)=Tout(m)+(E/A)*Ts
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A

                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A

              Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

              Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

              Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
              +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
```

```
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)

      If(nflux.EQ.1)Then
            If(m.EQ.Nfix(n))Then
                  Tinp(m)=Tout(m)+(E/A)*Ts+
     +            ((2*Qfix(n)*delt)/(deltay(nodmat(m,2))))/A
            n=n+1
            Endif
      Endif

      If(nflux.EQ.2)Then
            If(m.EQ.Nfix(n))Then
                  Tinp(m)=Tfix(n)
                  Coeff(m,m)=1
                  Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.

                  Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.
                  Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.


      Coe(m,m)=Coeff(m,m)

      Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                  n=n+1
            Endif
      Endif
1300  Continue
      End

      Subroutine BCsouthT(Coeff,Tinp,Ts,nrow,ncol,neq)
      Double Precision Coeff(neq,neq),Tinp(neq),Ts
      Integer nrow,ncol,neq

      Do j=2,ncol-1
            i=nrow
            m=j+(i-1)*ncol        !This gives the equation number
            Tinp(m)=Ts
            Coeff(m,j+(i-1)*ncol)=1
      Enddo
      End

      Subroutine BCeastC(Coeff,Tinp,Tout,r,props,conve,Te,
      +delr,deltay,delt,nodmat,inte,adialow,adiabats,
     +nrow,ncol,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),conve(neq,2),Te
      Double Precision delr,deltay(15),delt,Coe(396,396)
      Double Precision A,B,C,D,E,F,Ara,Arb,pi
      Integer nodmat(neq,2),inte(neq),nrow,ncol,neq
      Integer adialow,adiabats

      pi=3.14159265
      Do 1500 i=2,nrow-1
            j=ncol
            k=1
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number

            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
            B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +    (delr**2)
            C=(props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +      (deltay(nodmat(m,2))**2)
```

```
                D=(props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +     (deltay(nodmat(m,2))**2)
                E=(2*conve(m,2)*r(j)*delt)/(delr)

                Tinp(m)=Tout(m)+(E/A)*Te
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
              Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A

            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

            Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

            Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)

            Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
C               ------------------------------------------------------------
              If(adiabats.EQ.1)Then
                        If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
            Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A
            Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        Else
                Endif
            If(m.EQ.inte(m))Then
            Ara=pi*(r(j)-delr/2)*deltay(nodmat(m-ncol,2)) !Area of east face          [m^2]
            Arb=pi*(r(j)-delr/2)*deltay(nodmat(m+ncol,2)) !Area of east face          [m^2]
            A=(props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)*
     +  (r(j)-delr/4)*deltay(nodmat(m-ncol,2)))+
     +  (props(nodmat(m+ncol,2),2)*props(nodmat(m+ncol,2),3)*
     +  (r(j)-delr/4)*deltay(nodmat(m+ncol,2)))
            B=((2*delt)/(pi*delr))*
     +  ((delr/(Ara*props(nodmat(m-ncol,2),1))+delr/
     +  (Arb*props(nodmat(m+ncol,2),1)))/
     +  ((delr/(Ara*props(nodmat(m-ncol,2),1)))*
     +  (delr/(Arb*props(nodmat(m+ncol,2),1)))))
            C=(2*delt*(r(j)-delr/4)*props(nodmat(m+ncol,2),1))/
     +  (deltay(nodmat(m+ncol,2)))
            D=(2*delt*(r(j)-delr/4)*props(nodmat(m-ncol,2),1))/
     +  (deltay(nodmat(m-ncol,2)))
            E=(2*conve(m-ncol,2)*r(j)*delt*deltay(nodmat(m-ncol,2)))/(delr)
            F=(2*conve(m+ncol,2)*r(j)*delt*deltay(nodmat(m+ncol,2)))/(delr)
            Tinp(m)=Tout(m)+(E/A)*Te+(F/A)*Te
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A                    !West Node
            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A                    !North Node
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A                    !South Node
            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
```

```
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif
            If(adiabats.EQ.1)Then
                  If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.          !South Node
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A    !Center Node
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
                  If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.          !North Node
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A    !Center Node
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif
1500  Continue
      End

      Subroutine BCeastT(Coeff,Tinp,Te,nrow,ncol,neq)
      Double Precision Coeff(neq,neq),Tinp(neq),Te
      Integer nrow,ncol,neq

      Do i=2,nrow-1
            j=ncol
            m=j+(i-1)*ncol        !This gives the equation number
            Tinp(m)=Te
            Coeff(m,j+(i-1)*ncol)=1
      Enddo
      End

      Subroutine BCwestC(Coeff,Tinp,Tout,props,
      +delr,deltay,delt,nodmat,intw,adialow,adiabats,
     +nrow,ncol,neq,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision props(15,4),Coe(396,396)
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,Ara,Arb,pi
      Integer nodmat(neq,2),intw(neq),nrow,ncol,neq
      Integer adialow,adiabats

      pi=3.14159265
      Do 1700 i=2,nrow-1

            j=1
            k=1
            m=j+(i-1)*ncol+(k-1)*nrow*ncol        !This gives the equation number
            A=props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)
            B=(4*props(nodmat(m,2),1)*delt)/(delr)
            C=(props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
            D=(props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
            Tinp(m)=Tout(m)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D)/A                    !Center Node
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A                       !East Node
        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A                    !North Node
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A                       !South Node
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
```

```
            If(adiabats.EQ.1)Then
                        If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                    Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.                !North Node
                                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C)/A              !Center Node
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                                    Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.                !South Node
                                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D)/A              !Center Node
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
            Else
            Endif

      If(m.EQ.intw(m))Then
      Ara=(pi*delr*deltay(nodmat(m-ncol,2)))/2
      Arb=(pi*delr*deltay(nodmat(m+ncol,2)))/2
      A=(props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)*
     + deltay(nodmat(m-ncol,2)))+
     + (props(nodmat(m+ncol,2),2)*props(nodmat(m+ncol,2),3)*
     + deltay(nodmat(m+ncol,2)))
      B=(8*delt/(pi*delr**2))*
      +  ((delr/(Ara*props(nodmat(m-ncol,2),1)))+
     + (delr/(Arb*props(nodmat(m+ncol,2),1))))/
     + ((delr/(Ara*props(nodmat(m-ncol,2),1)))*
     + (delr/(Arb*props(nodmat(m+ncol,2),1))))
      C=(2*props(nodmat(m+ncol,2),1)*delt)/(deltay(nodmat(m+ncol,2)))
      D=(2*props(nodmat(m-ncol,2),1)*delt)/(deltay(nodmat(m-ncol,2)))
      Tinp(m)=Tout(m)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A                                            !East Node
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A                                            !North Node
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A                                            !South Node
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

            If(adiabats.EQ.1)Then
                        If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                    Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.                !South Node
                                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D)/A              !Center Node
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                                    Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.                !North Node
                                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C)/A              !Center Node
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
            Else
            Endif
1700  Continue
```

```
            End

C           SETS ACTUAL BOUNDARY CONDITIONS AT CORNER NODES
            Subroutine BCnwC2(Coeff,Tinp,Tout,props,Tnwy,hnwy,
            +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
            Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision props(15,4),Tnwy,hnwy,Coe(396,396)
            Double Precision delr,deltay(15),delt,delth(neq)
            Double Precision A,B,C,D
            Integer nodmat(neq,2),ncol,neq,nrow

            i=1
            j=1
            k=2
            m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number

            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
            B=(4*props(nodmat(m,2),1)*delt)/(delr**2)
            C=(2*props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
            D=(2*hnwy*delt)/(deltay(nodmat(m,2)))
            E=(16*props(nodmat(m,2),1)*delt)/((delr**2)*(delth(m)**2))
            F=(16*props(nodmat(m,2),1)*delt)/((delr**2)*(delth(m)**2))

      Tinp(m)=Tout(m)+(D/A)*Tnwy+
            +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
            +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            End

            Subroutine BCneC2(Coeff,Tinp,Tout,r,props,Tnex,Tney,hnex,hney,
            +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
            Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),Tnex,Tney,hnex,hney
            Double Precision delr,deltay(15),delt,Coe(396,396)
            Double Precision A,B,C,D,E,F,G,delth(neq)
            Integer nodmat(neq,2),ncol,neq,nrow
            i=1
            j=ncol
            k=2
            m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
            B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
            C=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +  (deltay(nodmat(m,2))**2)
            D=(2*hney*delt*(r(j)-delr/4))/(deltay(nodmat(m,2)))
            E=(2*135000*r(j)*delt)/(delr)
            F=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            G=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            Tinp(m)=Tout(m)+(D/A)*Tney+(E/A)*Tnex+
            +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
            +(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            End
```

```
      Subroutine BCswC2(Coeff,Tinp,Tout,props,Tswy,hswy,
      +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
 Double Precision props(15,4),Tswy,hswy
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq
 i=nrow
      j=1
      k=2
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
      B=(4*props(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
      D=(2*hswy*delt)/(deltay(nodmat(m,2)))
      E=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      F=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tswy+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCseC2(Coeff,Tinp,Tout,r,props,Tsex,Tsey,hsex,hsey,
      +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
 Double Precision r(ncol),props(15,4),Tsex,Tsey,hsex,hsey
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,G,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq

      i=nrow
      j=ncol
      k=2
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltay(nodmat(m,2))**2)
      D=(2*hsey*delt*(r(j)-delr/4))/(deltay(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)
      F=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      G=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tsey+(E/A)*Tsex+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
 End

C     -------------------------------------------------------------------------
C     SETS ACTUAL BOUNDARY CONDITIONS AT FACE NODES
      Subroutine BCnorthC2(Coeff,Tinp,Tout,r,props,Tn,hn,
      +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
```

```
      Double Precision r(ncol),props(15,4),Tn,hn,delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,G,Coe(396,396),delth(neq)
      Integer nodmat(neq,2),ncol,nrow,neq
      hn=0
      Do 1101 j=2,ncol-1
            i=1
            k=2
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
            B=(props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +        (r(j)*delr**2)
            C=(props(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +        (r(j)*delr**2)
            D=(2*props(nodmat(m,2),1)*delt)/
     +        (deltay(nodmat(m,2))**2)
            E=(2*hn*delt)/(deltay(nodmat(m,2)))
            F=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
            G=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Tn+
     +      (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +      (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
1101  Continue
      End

      Subroutine BCsouthC2(Coeff,Tinp,Tout,r,props,Qfix,Tfix,Ts,hs,
     +delr,deltay,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),Qfix(10),Tfix(10),Ts,hs
      Double Precision delr,deltay(15),delt,delth(neq)
      Double Precision A,B,C,D,E,F,G,Coe(396,396)
      Integer nodmat(neq,2),Nfix(10),nflux,nrow,ncol,neq

      n=17
      Do 1301 j=2,ncol-1
            i=nrow
            k=2
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
            B=(props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +        (r(j)*delr**2)
            C=(props(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +        (r(j)*delr**2)
            D=(2*props(nodmat(m,2),1)*delt)/
     +        (deltay(nodmat(m,2))**2)
            E=(2*hs*delt)/(deltay(nodmat(m,2)))
            F=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
            G=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Ts+
     +      (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +      (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
```

```
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      If(nflux.EQ.1)Then
            If(m.EQ.Nfix(n))Then
                  Tinp(m)=Tout(m)+(E/A)*Ts+
      +           (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +           (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)+
     +            ((2*Qfix(n)*delt)/(deltay(nodmat(m,2))))/A
            n=n+1
            Endif
      Endif

      If(nflux.EQ.2)Then
            If(m.EQ.Nfix(n))Then
                  Tinp(m)=Tfix(n)+
      +           (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +           (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                  Coeff(m,m)=1
                  Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.

                  Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.

                  Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.

      Coe(m-396,m-396)=Coeff(m,m)
      Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                  n=n+1
            Endif
      Endif
1301  Continue
      End

      Subroutine BCeastC2(Coeff,Tinp,Tout,r,props,conve,Te,
      +delr,deltay,delt,nodmat,inte,adialow,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),conve(neq,2),Te
      Double Precision delr,deltay(15),delt,Coe(396,396)
      Double Precision A,B,C,D,E,F,G,H,Ara,Arb,pi,delth(neq)
      Integer nodmat(neq,2),inte(neq),nrow,ncol,neq
      Integer adialow,adiabats
      pi=3.14159265
      Do 1501 i=2,nrow-1
            j=ncol
            k=2
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
            B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +    (delr**2)
            C=(props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +      (deltay(nodmat(m,2))**2)
            D=(props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +      (deltay(nodmat(m,2))**2)
            E=(2*conve(m,2)*r(j)*delt)/(delr)
            G=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            H=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Te+
      +     (G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +     (H/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+G+H)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
```

```
            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
C                   ------------------------------------------------------------
              If(adiabats.EQ.1)Then
                              If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                              Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G+H)/A

            Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)

            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                              Endif
                              If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                                      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                              Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F+G+H)/A

            Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)

            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                              Endif
                              Else
                      Endif

            If(m.EQ.inte(m))Then

            Ara=pi*(r(j)-delr/2)*deltay(nodmat(m-ncol,2))
            Arb=pi*(r(j)-delr/2)*deltay(nodmat(m+ncol,2))

            A=(props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)*
        +  (r(j)-delr/4)*deltay(nodmat(m-ncol,2)))+
        +  (props(nodmat(m+ncol,2),2)*props(nodmat(m+ncol,2),3)*
        +  (r(j)-delr/4)*deltay(nodmat(m+ncol,2)))
            B=((2*delt)/(pi*delr))*
        +  ((delr/(Ara*props(nodmat(m-ncol,2),1))+delr/
        +  (Arb*props(nodmat(m+ncol,2),1)))/
        +  ((delr/(Ara*props(nodmat(m-ncol,2),1)))*
        +  (delr/(Arb*props(nodmat(m+ncol,2),1)))))
            C=(2*delt*(r(j)-delr/4)*props(nodmat(m+ncol,2),1))/
        +  (deltay(nodmat(m+ncol,2)))
            D=(2*delt*(r(j)-delr/4)*props(nodmat(m-ncol,2),1))/
        +  (deltay(nodmat(m-ncol,2)))
            E=(2*conve(m-ncol,2)*r(j)*delt*deltay(nodmat(m-ncol,2)))/(delr)
            F=(2*conve(m+ncol,2)*r(j)*delt*deltay(nodmat(m+ncol,2)))/(delr)
            G=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
            +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*delt/
            +((r(j)-delr/4)*delth(m))**2)
            H=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
            +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*delt/
            +((r(j)-delr/4)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Te+(F/A)*Te+
            +(G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
            +(H/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G+H)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)

            Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
```

```
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)

      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)

      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

c               ------------------------------------------------------------
                If(adiabats.EQ.1)Then
                        If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F+G+H)/A
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G+H)/A
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                Else
                Endif
1501    Continue
      End

      Subroutine BCwestC2(Coeff,Tinp,Tout,props,
      +delr,deltay,delt,nodmat,intw,adialow,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision props(15,4),Coe(396,396)
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,Ara,Arb,pi,delth(neq)
      Integer nodmat(neq,2),intw(neq),nrow,ncol,neq
      Integer adialow,adiabats

      pi=3.14159265
      Do 1701 i=2,nrow-1
                j=1
                k=2
                m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
                A=props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)
                B=(4*props(nodmat(m,2),1)*delt)/(delr)
                C=(props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
                D=(props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
                E=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
                F=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
                Tinp(m)=Tout(m)+
      +         (E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +         (F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
                Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
C               ------------------------------------------------------------
        If(adiabats.EQ.1)Then
```

```
                        If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
              Else
              Endif

      If(m.EQ.intw(m))Then
      Ara=(pi*delr*deltay(nodmat(m-ncol,2)))/2
      Arb=(pi*delr*deltay(nodmat(m+ncol,2)))/2
      A=(props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)*
+ deltay(nodmat(m-ncol,2)))+
+ (props(nodmat(m+ncol,2),2)*props(nodmat(m+ncol,2),3)*
+ deltay(nodmat(m+ncol,2)))
      B=(8*delt/(pi*delr**2))*
      +  ((delr/(Ara*props(nodmat(m-ncol,2),1)))+
+ (delr/(Arb*props(nodmat(m+ncol,2),1))))/
+ ((delr/(Ara*props(nodmat(m-ncol,2),1)))*
+ (delr/(Arb*props(nodmat(m+ncol,2),1))))
      C=(2*props(nodmat(m+ncol,2),1)*delt)/(deltay(nodmat(m+ncol,2)))
      D=(2*props(nodmat(m-ncol,2),1)*delt)/(deltay(nodmat(m-ncol,2)))
      E=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
      +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*4*delt/
      +(delr*delth(m))**2)
      F=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
      +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*4*delt/
      +(delr*delth(m))**2)
      Tinp(m)=Tout(m)+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

              If(adiabats.EQ.1)Then
                        If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
```

```
            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                                    Endif
                        Else
                        Endif
1701        Continue
            End

            Subroutine BCnwC3(Coeff,Tinp,Tout,props,Tnwy,hnwy,
            +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
            Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision props(15,4),Tnwy,hnwy,Coe(396,396)
            Double Precision delr,deltay(15),delt
            Double Precision A,B,C,D,E,F,delth(neq)
            Integer nodmat(neq,2),ncol,neq,nrow
            i=1
            j=1
            k=3
            m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
            B=(4*props(nodmat(m,2),1)*delt)/(delr**2)
            C=(2*props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
            D=(2*hnwy*delt)/(deltay(nodmat(m,2)))
            E=(16*props(nodmat(m,2),1)*delt)/((delr**2)*(delth(m)**2))
            F=(16*props(nodmat(m,2),1)*delt)/((delr**2)*(delth(m)**2))
      Tinp(m)=Tout(m)+(D/A)*Tnwy+
            +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
            +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
            +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            End

            Subroutine BCneC3(Coeff,Tinp,Tout,r,props,Tnex,Tney,hnex,hney,
            +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
            Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),Tnex,Tney,hnex,hney
            Double Precision delr,deltay(15),delt,Coe(396,396)
            Double Precision A,B,C,D,E,F,G,delth(neq)
            Integer nodmat(neq,2),ncol,neq,nrow

            i=1
            j=ncol
            k=3
            m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
            A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
            B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
            C=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +  (deltay(nodmat(m,2))**2)
            D=(2*hney*delt*(r(j)-delr/4))/(deltay(nodmat(m,2)))
            E=(2*135000*r(j)*delt)/(delr)
            F=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            G=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            Tinp(m)=Tout(m)+(D/A)*Tney+(E/A)*Tnex+
            +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
            +(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
```

```
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCswC3(Coeff,Tinp,Tout,props,Tswy,hswy,
      +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision props(15,4),Tswy,hswy
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq
i=nrow
      j=1
      k=3
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
      B=(4*props(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
      D=(2*hswy*delt)/(deltay(nodmat(m,2)))
      E=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      F=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tswy+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A

      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCseC3(Coeff,Tinp,Tout,r,props,Tsex,Tsey,hsex,hsey,
      +delr,deltay,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision r(ncol),props(15,4),Tsex,Tsey,hsex,hsey
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,G,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq

      i=nrow
      j=ncol
      k=3
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltay(nodmat(m,2))**2)
      D=(2*hsey*delt*(r(j)-delr/4))/(deltay(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)
      F=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      G=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tsey+(E/A)*Tsex+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
End

      Subroutine BCnorthC3(Coeff,Tinp,Tout,r,props,Tn,hn,
```

```
     +delr,deltay,delt,nodmat,ncol,nrow,neq,Coe,delth)
     Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision r(ncol),props(15,4),Tn,hn,delr,deltay(15),delt
     Double Precision A,B,C,D,E,F,G,Coe(396,396),delth(neq)
     Integer nodmat(neq,2),ncol,nrow,neq
     hn=0
     Do 1102 j=2,ncol-1
               i=1
               k=3
               m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
               A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
               B=(props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +           (r(j)*delr**2)
               C=(props(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +           (r(j)*delr**2)
               D=(2*props(nodmat(m,2),1)*delt)/
     +           (deltay(nodmat(m,2))**2)
               E=(2*hn*delt)/(deltay(nodmat(m,2)))
               F=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
               G=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
               Tinp(m)=Tout(m)+(E/A)*Tn+
     +         (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +         (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
               Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
               Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
               Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
     Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
     +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
     +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
1102 Continue
     End

     Subroutine BCsouthC3(Coeff,Tinp,Tout,r,props,Qfix,Tfix,Ts,hs,
     +delr,deltay,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)
     Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision r(ncol),props(15,4),Qfix(10),Tfix(10),Ts,hs
     Double Precision delr,deltay(15),delt,delth(neq)
     Double Precision A,B,C,D,E,F,G,Coe(396,396)
     Integer nodmat(neq,2),Nfix(10),nflux,nrow,ncol,neq
     n=33
     Do 1302 j=2,ncol-1
               i=nrow
               k=3
               m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
               A=props(nodmat(m,2),2)*props(nodmat(m,2),3)
               B=(props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +           (r(j)*delr**2)
               C=(props(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +           (r(j)*delr**2)
               D=(2*props(nodmat(m,2),1)*delt)/
     +           (deltay(nodmat(m,2))**2)
               E=(2*hs*delt)/(deltay(nodmat(m,2)))
               F=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
               G=(props(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
               Tinp(m)=Tout(m)+(E/A)*Ts+
     +         (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +         (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
               Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
               Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
               Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
     Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
```

```
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)

      If(nflux.EQ.1)Then
            If(m.EQ.Nfix(n))Then
                  Tinp(m)=Tout(m)+(E/A)*Ts+
     +            (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +            (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)+
     +            ((2*Qfix(n)*delt)/(deltay(nodmat(m,2))))/A
            n=n+1
            Endif
      Endif

      If(nflux.EQ.2)Then
            If(m.EQ.Nfix(n))Then
                  Tinp(m)=Tfix(n)+
     +            (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +            (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                  Coeff(m,m)=1
                  Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.

                  Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.

                  Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.

      Coe(m-792,m-792)=Coeff(m,m)
      Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                  n=n+1
            Endif
      Endif
1302  Continue
      End

      Subroutine BCeastC3(Coeff,Tinp,Tout,r,props,conve,Te,
      +delr,deltay,delt,nodmat,inte,adialow,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,4),conve(neq,2),Te
      Double Precision delr,deltay(15),delt,Coe(396,396)
      Double Precision A,B,C,D,E,F,G,H,Ara,Arb,pi,delth(neq)
      Integer nodmat(neq,2),inte(neq),nrow,ncol,neq
      Integer adialow,adiabats

      pi=3.14159265
      Do 1502 i=2,nrow-1
            j=ncol
            k=3
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
                  A=props(nodmat(m,2),2)*props(nodmat(m,2),3)*(r(j)-delr/4)
            B=(2*props(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +    (delr**2)
            C=(props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +      (deltay(nodmat(m,2))**2)
            D=(props(nodmat(m,2),1)*delt*(r(j)-delr/4))/
     +      (deltay(nodmat(m,2))**2)
            E=(2*conve(m,2)*r(j)*delt)/(delr)
            G=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            H=(props(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Te+
      +     (G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +     (h/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+G+H)/A
```

```
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
           Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
C                 ------------------------------------------------------------
           If(adiabats.EQ.1)Then
                      If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                           Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G+H)/A
          Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                      Endif
                      If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                           Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F+G+H)/A
          Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                      Endif
                      Else
                Endif

          If(m.EQ.inte(m))Then

          Ara=pi*(r(j)-delr/2)*deltay(nodmat(m-ncol,2))
          Arb=pi*(r(j)-delr/2)*deltay(nodmat(m+ncol,2))
          A=(props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)*
     +  (r(j)-delr/4)*deltay(nodmat(m-ncol,2)))+
     +  (props(nodmat(m+ncol,2),2)*props(nodmat(m+ncol,2),3)*
     +  (r(j)-delr/4)*deltay(nodmat(m+ncol,2)))
          B=((2*delt)/(pi*delr))*
     +  ((delr/(Ara*props(nodmat(m-ncol,2),1))+delr/
     +  (Arb*props(nodmat(m+ncol,2),1)))/
     +  ((delr/(Ara*props(nodmat(m-ncol,2),1)))*
     +  (delr/(Arb*props(nodmat(m+ncol,2),1)))))
          C=(2*delt*(r(j)-delr/4)*props(nodmat(m+ncol,2),1))/
     +  (deltay(nodmat(m+ncol,2)))
          D=(2*delt*(r(j)-delr/4)*props(nodmat(m-ncol,2),1))/
     +  (deltay(nodmat(m-ncol,2)))
          E=(2*conve(m-ncol,2)*r(j)*delt*deltay(nodmat(m-ncol,2)))/(delr)
          F=(2*conve(m+ncol,2)*r(j)*delt*deltay(nodmat(m+ncol,2)))/(delr)
          G=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
          +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*delt/
          +((r(j)-delr/4)*delth(m))**2)
          H=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
          +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*delt/
          +((r(j)-delr/4)*delth(m))**2)
          Tinp(m)=Tout(m)+(E/A)*Te+(F/A)*Te+
          +(G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
          +(H/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
          Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G+H)/A
          Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
          Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
          Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
```

```
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

            If(adiabats.EQ.1)Then
                  If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                  Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                  Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F+G+H)/A
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
                  If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                  Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G+H)/A
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif
1502  Continue
      End

      Subroutine BCwestC3(Coeff,Tinp,Tout,props,
      +delr,deltay,delt,nodmat,intw,adialow,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision props(15,4),Coe(396,396)
      Double Precision delr,deltay(15),delt
      Double Precision A,B,C,D,E,F,Ara,Arb,pi,delth(neq)
      Integer nodmat(neq,2),intw(neq),nrow,ncol,neq
      Integer adialow,adiabats
      pi=3.14159265
      Do 1702 i=2,nrow-1
            j=1
            k=3
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
            A=props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)
            B=(4*props(nodmat(m,2),1)*delt)/(delr)
            C=(props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
            D=(props(nodmat(m,2),1)*delt)/(deltay(nodmat(m,2))**2)
            E=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
            F=(16*props(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
            Tinp(m)=Tout(m)+
     +      (E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +      (F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
C           ------------------------------------------------------------
        If(adiabats.EQ.1)Then
                  If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
```

```
                        Endif
                        If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
              Else
              Endif

      If(m.EQ.intw(m))Then
      Ara=(pi*delr*deltay(nodmat(m-ncol,2)))/2
      Arb=(pi*delr*deltay(nodmat(m+ncol,2)))/2
      A=(props(nodmat(m-ncol,2),2)*props(nodmat(m-ncol,2),3)*
+ deltay(nodmat(m-ncol,2)))+
+ (props(nodmat(m+ncol,2),2)*props(nodmat(m+ncol,2),3)*
+ deltay(nodmat(m+ncol,2)))
      B=(8*delt/(pi*delr**2))*
      +  ((delr/(Ara*props(nodmat(m-ncol,2),1)))+
+ (delr/(Arb*props(nodmat(m+ncol,2),1))))/
+ ((delr/(Ara*props(nodmat(m-ncol,2),1)))*
+ (delr/(Arb*props(nodmat(m+ncol,2),1))))
      C=(2*props(nodmat(m+ncol,2),1)*delt)/(deltay(nodmat(m+ncol,2)))
      D=(2*props(nodmat(m-ncol,2),1)*delt)/(deltay(nodmat(m-ncol,2)))
      E=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
      +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*4*delt/
      +(delr*delth(m))**2)
      F=((props(nodmat(m-ncol,2),1)*deltay(nodmat(m-ncol,2))+
      +props(nodmat(m+ncol,2),1)*deltay(nodmat(m+ncol,2)))*4*delt/
      +(delr*delth(m))**2)
      Tinp(m)=Tout(m)+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

              If(adiabats.EQ.1)Then
                        If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
              Else
              Endif
1702  Continue
      End
```

```
      Subroutine BCnwC4(Coeff,Tinp,Tout,pprops,Tnwy,hnwy,
      +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision pprops(15,4),Tnwy,hnwy,Coe(396,396)
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,F,delth(neq)
      Integer nodmat(neq,2),ncol,neq,nrow
      i=1
      j=1
      k=4
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
      B=(4*pprops(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
      D=(2*hnwy*delt)/(deltayy(nodmat(m,2)))
      E=(16*pprops(nodmat(m,2),1)*delt)/((delr**2)*(delth(m)**2))
      F=(16*pprops(nodmat(m,2),1)*delt)/((delr**2)*(delth(m)**2))
Tinp(m)=Tout(m)+(D/A)*Tnwy+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCneC4(Coeff,Tinp,Tout,r,pprops,Tnex,Tney,hnex,hney,
      +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision r(ncol),pprops(15,4),Tnex,Tney,hnex,hney
      Double Precision delr,deltayy(15),delt,Coe(396,396)
      Double Precision A,B,C,D,E,F,G,delth(neq)
      Integer nodmat(neq,2),ncol,neq,nrow

      i=1
      j=ncol
      k=4
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltayy(nodmat(m,2))**2)
      D=(2*hney*delt*(r(j)-delr/4))/(deltayy(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)
      F=(pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      G=(pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tney+(E/A)*Tnex+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCswC4(Coeff,Tinp,Tout,pprops,Tswy,hswy,
      +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision pprops(15,4),Tswy,hswy
```

```
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,F,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq

i=nrow
      j=1
      k=4
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
      B=(4*pprops(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
      D=(2*hswy*delt)/(deltayy(nodmat(m,2)))
      E=(16*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      F=(16*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tswy+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A

      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A

      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCseC4(Coeff,Tinp,Tout,r,pprops,Tsex,Tsey,hsex,hsey,
      +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
Double Precision r(ncol),pprops(15,4),Tsex,Tsey,hsex,hsey
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,F,G,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq
      i=nrow
      j=ncol
      k=4
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltayy(nodmat(m,2))**2)
      D=(2*hsey*delt*(r(j)-delr/4))/(deltayy(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)
      F=(pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      G=(pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tsey+(E/A)*Tsex+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
      +(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
End

      Subroutine BCnorthC4(Coeff,Tinp,Tout,r,pprops,Tn,hn,
      +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),pprops(15,4),Tn,hn,delr,deltayy(15),delt
      Double Precision A,B,C,D,E,F,G,Coe(396,396),delth(neq)
      Integer nodmat(neq,2),ncol,nrow,neq
      hn=0
```

```
      Do 1103 j=2,ncol-1
                  i=1
                  k=4
                  m=j+(i-1)*ncol+(k-1)*nrow*ncol              !This gives the equation number
                  A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
                  B=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +              (r(j)*delr**2)
                  C=(pprops(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +              (r(j)*delr**2)
                  D=(2*pprops(nodmat(m,2),1)*delt)/
     +              (deltayy(nodmat(m,2))**2)
                  E=(2*hn*delt)/(deltayy(nodmat(m,2)))
                  F=(pprops(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
                  G=(pprops(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
                  Tinp(m)=Tout(m)+(E/A)*Tn+
     +            (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +            (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                  Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
                  Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
         Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                  Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
1103  Continue
      End

      Subroutine BCsouthC4(Coeff,Tinp,Tout,r,pprops,Qfix,Tfix,Ts,hs,
      +delr,deltayy,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),pprops(15,4),Qfix(10),Tfix(10),Ts,hs
      Double Precision delr,deltayy(15),delt,delth(neq)
      Double Precision A,B,C,D,E,F,G,Coe(396,396)
      Integer nodmat(neq,2),Nfix(10),nflux,nrow,ncol,neq
      n=49
      Do 1303 j=2,ncol-1
                  i=nrow
                  k=4
                  m=j+(i-1)*ncol+(k-1)*nrow*ncol              !This gives the equation number
                  A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
                  B=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +              (r(j)*delr**2)
                  C=(pprops(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +              (r(j)*delr**2)
                  D=(2*pprops(nodmat(m,2),1)*delt)/
     +              (deltayy(nodmat(m,2))**2)
                  E=(2*hs*delt)/(deltayy(nodmat(m,2)))
                  F=(pprops(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
                  G=(pprops(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
                  Tinp(m)=Tout(m)+(E/A)*Ts+
     +            (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +            (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                  Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
                  Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
         Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                  Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      If(nflux.EQ.1)Then
```

```
                If(m.EQ.Nfix(n))Then
                        Tinp(m)=Tout(m)+(E/A)*Ts+
     +          (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +          (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)+
     +                  ((2*Qfix(n)*delt)/(deltayy(nodmat(m,2))))/A
                n=n+1
                Endif
        Endif

        If(nflux.EQ.2)Then
                If(m.EQ.Nfix(n))Then
                        Tinp(m)=Tfix(n)+
     +          (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +          (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                        Coeff(m,m)=1
                        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
        Coe(m-1188,m-1188)=Coeff(m,m)
        Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                        n=n+1
                Endif
        Endif
1303    Continue
        End

        Subroutine BCeastC4(Coeff,Tinp,Tout,r,pprops,conve,Te,
        +delr,deltayy,delt,nodmat,inte,adiabats,
   +nrow,ncol,neq,Coe,delth)
        Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
        Double Precision r(ncol),pprops(15,6),conve(neq,2),Te
        Double Precision delr,deltayy(15),delt,Coe(396,396)
        Double Precision A,B,C,D,E,F,G,H,Ara,Arb,pi,delth(neq)
        Integer nodmat(neq,2),inte(neq),nrow,ncol,neq
        Integer adiabats
        pi=3.14159265
        Do 1503 i=2,nrow-1
                j=ncol
                k=4
                m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
                A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)*(r(j)-delr/4)
                B=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/
   +    (delr**2)
                C=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
   +      (deltayy(nodmat(m,2))**2)
                D=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
   +      (deltayy(nodmat(m,2))**2)
                E=(2*conve(m,2)*r(j)*delt)/(delr)
                G=(pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
                H=(pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
                Tinp(m)=Tout(m)+(E/A)*Te+
        +       (G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
        +       (H/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+G+H)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
          Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
        Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
```

```
C                   ------------------------------------------------------------
              If(adiabats.EQ.1)Then
                              If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                              Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G+H)/A
            Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                              Endif
                              Else
                      Endif

            If(m.EQ.inte(m))Then
            Ara=pi*(r(j)-delr/2)*deltayy(nodmat(m-ncol,2))
            Arb=pi*(r(j)-delr/2)*deltayy(nodmat(m+ncol,2))
            A=(pprops(nodmat(m-ncol,2),2)*pprops(nodmat(m-ncol,2),3)*
     +  (r(j)-delr/4)*deltayy(nodmat(m-ncol,2)))+
     +  (pprops(nodmat(m+ncol,2),2)*pprops(nodmat(m+ncol,2),3)*
     +  (r(j)-delr/4)*deltayy(nodmat(m+ncol,2)))
            B=((2*delt)/(pi*delr))*
     +  ((delr/(Ara*pprops(nodmat(m-ncol,2),1))+delr/
     +  (Arb*pprops(nodmat(m+ncol,2),1)))/
     +  ((delr/(Ara*pprops(nodmat(m-ncol,2),1)))*
     +  (delr/(Arb*pprops(nodmat(m+ncol,2),1)))))
            C=(2*delt*(r(j)-delr/4)*pprops(nodmat(m+ncol,2),1))/
     +  (deltayy(nodmat(m+ncol,2)))
            D=(2*delt*(r(j)-delr/4)*pprops(nodmat(m-ncol,2),1))/
     +  (deltayy(nodmat(m-ncol,2)))
            E=(2*conve(m-ncol,2)*r(j)*delt*deltayy(nodmat(m-ncol,2)))/(delr)
            F=(2*conve(m+ncol,2)*r(j)*delt*deltayy(nodmat(m+ncol,2)))/(delr)
            G=((pprops(nodmat(m-ncol,2),1)*deltayy(nodmat(m-ncol,2))+
            +pprops(nodmat(m+ncol,2),1)*deltayy(nodmat(m+ncol,2)))*delt/
            +((r(j)-delr/4)*delth(m))**2)
            H=((pprops(nodmat(m-ncol,2),1)*deltayy(nodmat(m-ncol,2))+
            +pprops(nodmat(m+ncol,2),1)*deltayy(nodmat(m+ncol,2)))*delt/
            +((r(j)-delr/4)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Te+(F/A)*Te+
            +(G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
            +(H/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G+H)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            Endif

                      If(adiabats.EQ.1)Then
                              If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                              Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F+G+H)/A
            Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                              Endif
                      Else
                      Endif
1503        Continue
            End

            Subroutine BCwestC4(Coeff,Tinp,Tout,pprops,
            +delr,deltayy,delt,nodmat,intw,adiabats,
```

```
     +nrow,ncol,neq,Coe,delth)
           Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
           Double Precision pprops(15,6),Coe(396,396)
           Double Precision delr,deltayy(15),delt
           Double Precision A,B,C,D,E,F,Ara,Arb,pi,delth(neq)
           Integer nodmat(neq,2),intw(neq),nrow,ncol,neq
           Integer adiabats
           pi=3.14159265
           Do 1703 i=2,nrow-1
                 j=1
                 k=4
                 m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
                 A=pprops(nodmat(m-ncol,2),2)*pprops(nodmat(m-ncol,2),3)
                 B=(4*pprops(nodmat(m,2),1)*delt)/(delr)
                 C=(pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
                 D=(pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
                 E=(16*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
                 F=(16*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
                 Tinp(m)=Tout(m)+
           +     (E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
           +     (F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                 Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
                 Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
             Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                 Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
           Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
           +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
           Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
           +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
           Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
           +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
           Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
           +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
     C           ------------------------------------------------------------
             If(adiabats.EQ.1)Then
                       If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                             Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                             Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
           Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
           +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
           Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
           +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                       Endif
                 Else
                 Endif
           If(m.EQ.intw(m))Then
           Ara=(pi*delr*deltayy(nodmat(m-ncol,2)))/2
           Arb=(pi*delr*deltayy(nodmat(m+ncol,2)))/2
           A=(pprops(nodmat(m-ncol,2),2)*pprops(nodmat(m-ncol,2),3)*
     +  deltayy(nodmat(m-ncol,2)))+
     +  (pprops(nodmat(m+ncol,2),2)*pprops(nodmat(m+ncol,2),3)*
     +  deltayy(nodmat(m+ncol,2)))
           B=(8*delt/(pi*delr**2))*
           +  ((delr/(Ara*pprops(nodmat(m-ncol,2),1)))+
     +  (delr/(Arb*pprops(nodmat(m+ncol,2),1))))/
     +  ((delr/(Ara*pprops(nodmat(m-ncol,2),1)))*
     +  (delr/(Arb*pprops(nodmat(m+ncol,2),1))))
           C=(2*pprops(nodmat(m+ncol,2),1)*delt)/(deltayy(nodmat(m+ncol,2)))
           D=(2*pprops(nodmat(m-ncol,2),1)*delt)/(deltayy(nodmat(m-ncol,2)))
           E=((pprops(nodmat(m-ncol,2),1)*deltayy(nodmat(m-ncol,2))+
           +pprops(nodmat(m+ncol,2),1)*deltayy(nodmat(m+ncol,2)))*4*delt/
           +(delr*delth(m))**2)
           F=((pprops(nodmat(m-ncol,2),1)*deltayy(nodmat(m-ncol,2))+
           +pprops(nodmat(m+ncol,2),1)*deltayy(nodmat(m+ncol,2)))*4*delt/
           +(delr*delth(m))**2)
           Tinp(m)=Tout(m)+
           +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
           +(F/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
           Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
           Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
```

```
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

            If(adiabats.EQ.1)Then

                  If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F)/A
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif

1703  Continue
      End

      Subroutine BCnwC5(Coeff,Tinp,Tout,pprops,Tnwy,hnwy,
      +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
   Double Precision pprops(15,4),Tnwy,hnwy,Coe(396,396)
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,delth(neq)
      Integer nodmat(neq,2),ncol,neq,nrow

      i=1
      j=1
      k=5
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
      B=(4*pprops(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
      D=(2*hnwy*delt)/(deltayy(nodmat(m,2)))
      E=(64*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
   Tinp(m)=Tout(m)+(D/A)*Tnwy+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCneC5(Coeff,Tinp,Tout,r,pprops,Tnex,Tney,hnex,hney,
      +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
   Double Precision r(ncol),pprops(15,4),Tnex,Tney,hnex,hney
      Double Precision delr,deltayy(15),delt,Coe(396,396)
      Double Precision A,B,C,D,E,F,delth(neq)
      Integer nodmat(neq,2),ncol,neq,nrow
      i=1
      j=ncol
      k=5
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)*(r(j)-delr/4)
```

```
      B=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltayy(nodmat(m,2))**2)
      D=(2*hney*delt*(r(j)-delr/4))/(deltayy(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)
      F=(16*pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tney+(E/A)*Tnex+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCswC5(Coeff,Tinp,Tout,pprops,Tswy,hswy,
      +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
 Double Precision pprops(15,4),Tswy,hswy
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq

 i=nrow
      j=1
      k=5
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
      B=(4*pprops(nodmat(m,2),1)*delt)/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
      D=(2*hswy*delt)/(deltayy(nodmat(m,2)))
      E=(64*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tswy+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A

      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A

      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCseC5(Coeff,Tinp,Tout,r,pprops,Tsex,Tsey,hsex,hsey,
      +delr,deltayy,delt,nodmat,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
 Double Precision r(ncol),pprops(15,4),Tsex,Tsey,hsex,hsey
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,F,delth(neq),Coe(396,396)
      Integer nodmat(neq,2),nrow,ncol,neq
      i=nrow
      j=ncol
      k=5
      m=j+(i-1)*ncol+(k-1)*nrow*ncol !This gives the equation number
      A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)*(r(j)-delr/4)
      B=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/(delr**2)
      C=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
+  (deltayy(nodmat(m,2))**2)
      D=(2*hsey*delt*(r(j)-delr/4))/(deltayy(nodmat(m,2)))
      E=(2*135000*r(j)*delt)/(delr)
      F=(16*pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
      Tinp(m)=Tout(m)+(D/A)*Tsey+(E/A)*Tsex+
```

```
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      End

      Subroutine BCnorthC5(Coeff,Tinp,Tout,r,pprops,Tn,hn,
      +delr,deltayy,delt,nodmat,ncol,nrow,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),pprops(15,4),Tn,hn,delr,deltayy(15),delt
      Double Precision A,B,C,D,E,F,Coe(396,396),delth(neq)
      Integer nodmat(neq,2),ncol,nrow,neq
      hn=0
      Do 1104 j=2,ncol-1
            i=1
            k=5
            m=j+(i-1)*ncol+(k-1)*nrow*ncol        !This gives the equation number
            A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
            B=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +        (r(j)*delr**2)
            C=(pprops(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +        (r(j)*delr**2)
            D=(2*pprops(nodmat(m,2),1)*delt)/
     +        (deltayy(nodmat(m,2))**2)
            E=(2*hn*delt)/(deltayy(nodmat(m,2)))
            F=(4*pprops(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
            Tinp(m)=Tout(m)+(E/A)*Tn+
     +        (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
1104  Continue
      End

      Subroutine BCsouthC5(Coeff,Tinp,Tout,r,pprops,Qfix,Tfix,Ts,hs,
      +delr,deltayy,delt,nodmat,Nfix,nflux,nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),pprops(15,4),Qfix(10),Tfix(10),Ts,hs
      Double Precision delr,deltayy(15),delt,delth(neq)
      Double Precision A,B,C,D,E,F,Coe(396,396)
      Integer nodmat(neq,2),Nfix(10),nflux,nrow,ncol,neq
      n=65
      Do 1304 j=2,ncol-1
            i=nrow
            k=5
            m=j+(i-1)*ncol+(k-1)*nrow*ncol        !This gives the equation number
            A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)
            B=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/
     +        (r(j)*delr**2)
            C=(pprops(nodmat(m,2),1)*delt*(r(j)+delr/2))/
     +        (r(j)*delr**2)
            D=(2*pprops(nodmat(m,2),1)*delt)/
     +        (deltayy(nodmat(m,2))**2)
            E=(2*hs*delt)/(deltayy(nodmat(m,2)))
            F=(4*pprops(nodmat(m,2),1)*delt)/((r(j)*delth(m))**2)
```

```
                Tinp(m)=Tout(m)+(E/A)*Ts+
     +          (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A

       Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A

     Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)

     If(nflux.EQ.1)Then
                If(m.EQ.Nfix(n))Then
                        Tinp(m)=Tout(m)+(E/A)*Ts+
     +                  (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
  +                     ((2*Qfix(n)*delt)/(deltayy(nodmat(m,2))))/A
                n=n+1
                Endif
     Endif

     If(nflux.EQ.2)Then
                If(m.EQ.Nfix(n))Then
                        Tinp(m)=Tfix(n)+
     +          (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
                        Coeff(m,m)=1
                        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.

                        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
     Coe(m-1584,m-1584)=Coeff(m,m)
     Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                        n=n+1
                Endif
     Endif
1304 Continue
     End

     Subroutine BCeastC5(Coeff,Tinp,Tout,r,pprops,conve,Te,
     +delr,deltayy,delt,nodmat,inte,adiabats,
  +nrow,ncol,neq,Coe,delth)
     Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
     Double Precision r(ncol),pprops(15,6),conve(neq,2),Te
     Double Precision delr,deltayy(15),delt,Coe(396,396)
     Double Precision A,B,C,D,E,F,G,Ara,Arb,pi,delth(neq)
     Integer nodmat(neq,2),inte(neq),nrow,ncol,neq
     Integer adiabats
     pi=3.14159265
     Do 1504 i=2,nrow-1
                j=ncol
                k=5
                m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
                A=pprops(nodmat(m,2),2)*pprops(nodmat(m,2),3)*(r(j)-delr/4)
                B=(2*pprops(nodmat(m,2),1)*delt*(r(j)-delr/2))/
  +     (delr**2)
                C=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
  +       (deltayy(nodmat(m,2))**2)
                D=(pprops(nodmat(m,2),1)*delt*(r(j)-delr/4))/
  +       (deltayy(nodmat(m,2))**2)
                E=(2*conve(m,2)*r(j)*delt)/(delr)
```

```
                G=(4*pprops(nodmat(m,2),1)*delt)/(((r(j)-delr/4)*delth(m))**2)
                Tinp(m)=Tout(m)+(E/A)*Te+
     +          (G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+G)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
       Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
     Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
C               ------------------------------------------------------------
       If(adiabats.EQ.1)Then
                   If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G)/A
     Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                   Endif
                   Else
                Endif

     If(m.EQ.inte(m))Then
     Ara=pi*(r(j)-delr/2)*deltayy(nodmat(m-ncol,2))
     Arb=pi*(r(j)-delr/2)*deltayy(nodmat(m+ncol,2))
     A=(pprops(nodmat(m-ncol,2),2)*pprops(nodmat(m-ncol,2),3)*
    + (r(j)-delr/4)*deltayy(nodmat(m-ncol,2)))+
    + (pprops(nodmat(m+ncol,2),2)*pprops(nodmat(m+ncol,2),3)*
    + (r(j)-delr/4)*deltayy(nodmat(m+ncol,2)))
     B=((2*delt)/(pi*delr))*
    + ((delr/(Ara*pprops(nodmat(m-ncol,2),1))+delr/
    + (Arb*pprops(nodmat(m+ncol,2),1)))/
    + ((delr/(Ara*pprops(nodmat(m-ncol,2),1)))*
    + (delr/(Arb*pprops(nodmat(m+ncol,2),1)))))
     C=(2*delt*(r(j)-delr/4)*pprops(nodmat(m+ncol,2),1))/
    + (deltayy(nodmat(m+ncol,2)))
     D=(2*delt*(r(j)-delr/4)*pprops(nodmat(m-ncol,2),1))/
    + (deltayy(nodmat(m-ncol,2)))
     E=(2*conve(m-ncol,2)*r(j)*delt*deltayy(nodmat(m-ncol,2)))/(delr)
     F=(2*conve(m+ncol,2)*r(j)*delt*deltayy(nodmat(m+ncol,2)))/(delr)
     G=((pprops(nodmat(m-ncol,2),1)*deltayy(nodmat(m-ncol,2))+
     +pprops(nodmat(m+ncol,2),1)*deltayy(nodmat(m+ncol,2)))*delt/
     +((r(j)-delr/4)*delth(m)/2)**2)
     Tinp(m)=Tout(m)+(E/A)*Te+(F/A)*Te+
     +(G/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
     Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
     Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
     Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
     Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
     Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
     Endif

                If(adiabats.EQ.1)Then
                   If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E+F+G)/A
     Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
```

```
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif
1504  Continue
      End

      Subroutine BCwestC5(Coeff,Tinp,Tout,pprops,
      +delr,deltayy,delt,nodmat,intw,adiabats,
     +nrow,ncol,neq,Coe,delth)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision pprops(15,6),Coe(396,396)
      Double Precision delr,deltayy(15),delt
      Double Precision A,B,C,D,E,Ara,Arb,pi,delth(neq)
      Integer nodmat(neq,2),intw(neq),nrow,ncol,neq
      Integer adiabats
      pi=3.14159265
      Do 1704 i=2,nrow-1

            j=1
            k=5
            m=j+(i-1)*ncol+(k-1)*nrow*ncol          !This gives the equation number
            A=pprops(nodmat(m-ncol,2),2)*pprops(nodmat(m-ncol,2),3)
            B=(4*pprops(nodmat(m,2),1)*delt)/(delr)
            C=(pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
            D=(pprops(nodmat(m,2),1)*delt)/(deltayy(nodmat(m,2))**2)
            E=(64*pprops(nodmat(m,2),1)*delt)/((delr*delth(m))**2)
            Tinp(m)=Tout(m)+
      +     (E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)

        If(adiabats.EQ.1)Then
                  If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E)/A
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif

      If(m.EQ.intw(m))Then
      Ara=(pi*delr*deltayy(nodmat(m-ncol,2)))/2
      Arb=(pi*delr*deltayy(nodmat(m+ncol,2)))/2
      A=(pprops(nodmat(m-ncol,2),2)*pprops(nodmat(m-ncol,2),3)*
     + deltayy(nodmat(m-ncol,2)))+
     + (pprops(nodmat(m+ncol,2),2)*pprops(nodmat(m+ncol,2),3)*
     + deltayy(nodmat(m+ncol,2)))
      B=(8*delt/(pi*delr**2))*
      +  ((delr/(Ara*pprops(nodmat(m-ncol,2),1)))+
     + (delr/(Arb*pprops(nodmat(m+ncol,2),1))))/
     + ((delr/(Ara*pprops(nodmat(m-ncol,2),1)))*
     + (delr/(Arb*pprops(nodmat(m+ncol,2),1))))
      C=(2*pprops(nodmat(m+ncol,2),1)*delt)/(deltayy(nodmat(m+ncol,2)))
      D=(2*pprops(nodmat(m-ncol,2),1)*delt)/(deltayy(nodmat(m-ncol,2)))
```

```
      E=((pprops(nodmat(m-ncol,2),1)*deltayy(nodmat(m-ncol,2))+
      +pprops(nodmat(m+ncol,2),1)*deltayy(nodmat(m+ncol,2)))*4*delt/
      +(delr*delth(m)/2)**2)
      Tinp(m)=Tout(m)+
      +(E/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Endif

            If(adiabats.EQ.1)Then
                  If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+D+E)/A
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif
1704  Continue
      End

      Subroutine Interior(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe)
      Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
      Double Precision r(ncol),props(15,6),delr,deltay(15),delt
      Double Precision A,B,C,D,E,pi,Coe(396,396)
     Double Precision Aya,Ayb,Ayc,Ayd,Ara,Arb,Arc,Ard
      Double Precision knw,kne,ksw,kse,cpnw,cpne,cpsw,cpse
      Double Precision rhonw,rhone,rhosw,rhose
      Double Precision Arain,Araout,Arbin,Arbout
      Double Precision Ayain,Ayaout,Aybin,Aybout
      Double Precision kn,ks,cpn,cps,rhon,rhos
      Double Precision kw,ke,cpw,cpe,rhow,rhoe
      Integer mnw,mne,msw,mse
      Integer pdim(15,6),nodmati(neq,2),nodmat(neq,2)
      Integer nrow,ncol,neq,mats,nodint(neq,2)
      Integer adialow,adiabats
      pi=3.14159265
      Do 2000 ii=1,mats
            Do 2100 i=pdim(ii,1)+1,pdim(ii,3)-1
                  Do 2200 j=pdim(ii,2)+1,pdim(ii,4)-1
                  k=1
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        A=props(ii,2)*props(ii,3)
                        B=(props(ii,1)*delt*(r(j)-delr/2))/
     +                   (delr**2*r(j))
                        C=(props(ii,1)*delt*(r(j)+delr/2))/
     +                   (delr**2*r(j))
                        D=(props(ii,1)*delt)/
     +                     (deltay(nodmat(m,2))**2)
                        E=(props(ii,1)*delt)/
     +                     (deltay(nodmat(m,2))**2)
                        Tinp(m)=Tout(m)
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
```

```
                                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)

                        If(adiabats.EQ.1)Then
                                If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D)/A
            Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                                Endif
                                If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E)/A
            Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                                Endif
                        Else
                        Endif
c                       ------------------------------------
2200                            Continue
2100                    Continue
2000    Continue

            Do 2600 i=2,nrow-1
                        Do 2700 j=2,ncol-1
                        k=1
                                m=j+(i-1)*ncol+(k-1)*nrow*ncol

                                If(m.NE.nodmati(m,1))Then
      mnw=(j-1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
            knw=props(nodmat(mnw,2),1)
            cpnw=props(nodmat(mnw,2),2)
            rhonw=props(nodmat(mnw,2),3)
      mne=(j+1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
            kne=props(nodmat(mne,2),1)
            cpne=props(nodmat(mne,2),2)
            rhone=props(nodmat(mne,2),3)
      msw=(j-1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
            ksw=props(nodmat(msw,2),1)
            cpsw=props(nodmat(msw,2),2)
            rhosw=props(nodmat(msw,2),3)
            mse=(j+1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
            kse=props(nodmat(mse,2),1)
            cpse=props(nodmat(mse,2),2)
            rhose=props(nodmat(mse,2),3)
            Aya=pi*(r(j)**2-(r(j)-delr/2)**2)
            Ayc=Aya
            Ayb=pi*((r(j)+delr/2)**2-r(j)**2)
            Ayd=Ayb
            Ara=2*pi*(r(j)-delr/2)*deltay(nodmat(mnw,2))
            Arc=2*pi*(r(j)-delr/2)*deltay(nodmat(msw,2))
            Arb=2*pi*(r(j)+delr/2)*deltay(nodmat(mne,2))
            Ard=2*pi*(r(j)+delr/2)*deltay(nodmat(mse,2))
            A=(rhonw*cpnw*(r(j)-delr/4)*deltay(nodmat(mnw,2)))+
     +  (rhone*cpne*(r(j)+delr/4)*deltay(nodmat(mne,2)))+
     +  (rhosw*cpsw*(r(j)-delr/4)*deltay(nodmat(msw,2)))+
     +  (rhose*cpse*(r(j)+delr/4)*deltay(nodmat(mse,2)))
```

```
      B=((2*delt)/(pi*delr))*
+  ((delr/(Ara*knw)+delr/(Arc*ksw))/
+  ((delr/(Ara*knw))*(delr/(Arc*ksw))))
      C=((2*delt)/(pi*delr))*
+  ((delr/(Arb*kne)+delr/(Ard*kse))/
+  ((delr/(Arb*kne))*(delr/(Ard*kse))))
      D=((2*delt)/(pi*delr))*
+  (((deltay(nodmat(msw,2))/(ksw*Ayc))+
+  (deltay(nodmat(mse,2))/(kse*Ayd)))/
+  ((deltay(nodmat(msw,2))/(ksw*Ayc))*
+  (deltay(nodmat(mse,2))/(kse*Ayd))))
      E=((2*delt)/(pi*delr))*
+  (((deltay(nodmat(mnw,2))/(knw*Aya))+
+  (deltay(nodmat(mne,2))/(kne*Ayb)))/
+  ((deltay(nodmat(mnw,2))/(knw*Aya))*
+  (deltay(nodmat(mne,2))/(kne*Ayb))))
      Tinp(m)=Tout(m)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                  Endif
                  If(nodint(m,2).EQ.1)Then
            Arain=2*pi*(r(j)-(delr/2))*deltay(nodmat(m-ncol,2))
            Araout=2*pi*(r(j)+(delr/2))*deltay(nodmat(m-ncol,2))
            Arbin=2*pi*(r(j)-(delr/2))*deltay(nodmat(m+ncol,2))
            Arbout=2*pi*(r(j)+(delr/2))*deltay(nodmat(m+ncol,2))
            kn=props(nodmat(m-ncol,2),1)
            cpn=props(nodmat(m-ncol,2),2)
            rhon=props(nodmat(m-ncol,2),3)
            ks=props(nodmat(m+ncol,2),1)
            cps=props(nodmat(m+ncol,2),2)
            rhos=props(nodmat(m+ncol,2),3)
        A=(rhon*cpn*deltay(nodmat(m-ncol,2)))+
+       (rhos*cps*deltay(nodmat(m+ncol,2)))
            B=((delt)/(pi*r(j)*delr))*
+       ((delr/(Arain*kn)+delr/(Arbin*ks))/
+       ((delr/(Arain*kn))*(delr/(Arbin*ks))))
            C=((delt)/(pi*r(j)*delr))*
+       ((delr/(Araout*kn)+delr/(Arbout*ks))/
+       ((delr/(Araout*kn))*(delr/(Arbout*ks))))
            D=(2*ks*delt)/deltay(nodmat(m+ncol,2))
            E=(2*kn*delt)/deltay(nodmat(m-ncol,2))
            Tinp(m)=Tout(m)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
```

```
                If(adiabats.EQ.1)Then
                        If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E)/A
        Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
        Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D)/A
        Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
        Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                Else
                Endif
                        Endif
                        If(nodint(m,2).EQ.2)Then
                Ayain=pi*(r(j)**2-(r(j)-delr/2)**2)
                Ayaout=Ayain
                Aybin=pi*((r(j)+delr/2)**2-r(j)**2)
                Aybout=Aybin
                kw=props(nodmat(m-1,2),1)
                cpw=props(nodmat(m-1,2),2)
                rhow=props(nodmat(m-1,2),3)
                ke=props(nodmat(m+1,2),1)
                cpe=props(nodmat(m+1,2),2)
                rhoe=props(nodmat(m+1,2),3)
                A=(rhow*cpw*(r(j)-(delr/4))*deltay(nodmat(m-1,2)))+
     +          (rhoe*cpe*(r(j)+(delr/4))*deltay(nodmat(m+1,2)))
                B=(kw*delt*(r(j)-(delr/2))*deltay(nodmat(m-1,2)))/delr**2
                C=(ke*delt*(r(j)+(delr/2))*deltay(nodmat(m+1,2)))/delr**2
                D=((delt)/(2*pi*delr))*
     +          ((delr/(Ayain*kw)+delr/(Aybin*ke))/
     +          ((delr/(Ayain*kw))*(delr/(Aybin*ke))))
                E=((delt)/(2*pi*delr))*
     +          ((delr/(Ayaout*kw)+delr/(Aybout*ke))/
     +          ((delr/(Ayaout*kw))*(delr/(Aybout*ke))))
                Tinp(m)=Tout(m)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
        Coe(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
        Coe(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=
        +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                        Endif
2700            Continue
2600    Continue
        End

        Subroutine Interior2(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
    +pdim,nodmati,nodmat,nodint,adialow,adiabats,
    +nrow,ncol,neq,mats,Coe,delth)
        Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
        Double Precision r(ncol),props(15,6),delr,deltay(15),delt
        Double Precision A,B,C,D,E,F,G,pi,Coe(396,396),delth(neq)
     Double Precision Aya,Ayb,Ayc,Ayd,Ara,Arb,Arc,Ard
        Double Precision knw,kne,ksw,kse,cpnw,cpne,cpsw,cpse
```

```
      Double Precision rhonw,rhone,rhosw,rhose
      Double Precision Arain,Araout,Arbin,Arbout
      Double Precision Ayain,Ayaout,Aybin,Aybout
      Double Precision kn,ks,cpn,cps,rhon,rhos
      Double Precision kw,ke,cpw,cpe,rhow,rhoe
      Integer mnw,mne,msw,mse
      Integer pdim(15,6),nodmati(neq,2),nodmat(neq,2)
      Integer nrow,ncol,neq,mats,nodint(neq,2)
      Integer adialow,adiabats
      pi=3.14159265
      Do 2001 ii=1,mats
            Do 2101 i=pdim(ii,1)+1,pdim(ii,3)-1
                  Do 2201 j=pdim(ii,2)+1,pdim(ii,4)-1
                  k=2
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        A=props(ii,2)*props(ii,3)
                        B=(props(ii,1)*delt*(r(j)-delr/2))/
     +                   (delr**2*r(j))
                        C=(props(ii,1)*delt*(r(j)+delr/2))/
     +                   (delr**2*r(j))
                        D=(props(ii,1)*delt)/
     +                     (deltay(nodmat(m,2))**2)
                        E=(props(ii,1)*delt)/
     +                     (deltay(nodmat(m,2))**2)
                        F=(props(ii,1)*delt)/((r(j)*delth(m))**2)
                        G=(props(ii,1)*delt)/((r(j)*delth(m))**2)
                        Tinp(m)=Tout(m)+
     +                  (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +                  (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                  Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
                        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
c           ------------------------------------
            If(adiabats.EQ.1)Then
                  If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+F+G)/A
      Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
                  If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G)/A
      Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                  Endif
            Else
            Endif
c           ------------------------------------
2201                    Continue
2101              Continue
2001      Continue

      Do 2601 i=2,nrow-1
```

```
                Do 2701 j=2,ncol-1
                k=2
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol

                        If(m.NE.nodmati(m,1))Then
 mnw=(j-1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
        knw=props(nodmat(mnw,2),1)
        cpnw=props(nodmat(mnw,2),2)
        rhonw=props(nodmat(mnw,2),3)
 mne=(j+1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
        kne=props(nodmat(mne,2),1)
        cpne=props(nodmat(mne,2),2)
        rhone=props(nodmat(mne,2),3)
 msw=(j-1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
        ksw=props(nodmat(msw,2),1)
        cpsw=props(nodmat(msw,2),2)
        rhosw=props(nodmat(msw,2),3)
        mse=(j+1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
        kse=props(nodmat(mse,2),1)
        cpse=props(nodmat(mse,2),2)
        rhose=props(nodmat(mse,2),3)
        Aya=pi*(r(j)**2-(r(j)-delr/2)**2)
        Ayc=Aya
        Ayb=pi*((r(j)+delr/2)**2-r(j)**2)
        Ayd=Ayb
        Ara=2*pi*(r(j)-delr/2)*deltay(nodmat(mnw,2))
        Arc=2*pi*(r(j)-delr/2)*deltay(nodmat(msw,2))
        Arb=2*pi*(r(j)+delr/2)*deltay(nodmat(mne,2))
        Ard=2*pi*(r(j)+delr/2)*deltay(nodmat(mse,2))
        A=(rhonw*cpnw*(r(j)-delr/4)*deltay(nodmat(mnw,2)))+
+  (rhone*cpne*(r(j)+delr/4)*deltay(nodmat(mne,2)))+
+  (rhosw*cpsw*(r(j)-delr/4)*deltay(nodmat(msw,2)))+
+  (rhose*cpse*(r(j)+delr/4)*deltay(nodmat(mse,2)))
        B=((2*delt)/(pi*delr))*
+  ((delr/(Ara*knw)+delr/(Arc*ksw))/
+  ((delr/(Ara*knw))*(delr/(Arc*ksw))))
        C=((2*delt)/(pi*delr))*
+  ((delr/(Arb*kne)+delr/(Ard*kse))/
+  ((delr/(Arb*kne))*(delr/(Ard*kse))))
        D=((2*delt)/(pi*delr))*
+  (((deltay(nodmat(msw,2))/(ksw*Ayc))+
+  (deltay(nodmat(mse,2))/(kse*Ayd)))/
+  ((deltay(nodmat(msw,2))/(ksw*Ayc))*
+  (deltay(nodmat(mse,2))/(kse*Ayd))))
        E=((2*delt)/(pi*delr))*
+  (((deltay(nodmat(mnw,2))/(knw*Aya))+
+  (deltay(nodmat(mne,2))/(kne*Ayb)))/
+  ((deltay(nodmat(mnw,2))/(knw*Aya))*
+  (deltay(nodmat(mne,2))/(kne*Ayb))))
        F=((knw*deltay(nodmat(m+ncol,2))+
        +       ksw*deltay(nodmat(m-ncol,2))*delt)/
        +       (r(j)*delth(nodmat(m+ncol,2)))**2)
        G=((knw*deltay(nodmat(m+ncol,2))+
        +       ksw*deltay(nodmat(m-ncol,2))*delt)/
        +       (r(j)*delth(nodmat(m+ncol,2)))**2)
        Tinp(m)=Tout(m)+
        +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
+(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
        Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
        +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
        +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
```

```
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                    Endif
                    If(nodint(m,2).EQ.1)Then
               Arain=2*pi*(r(j)-(delr/2))*deltay(nodmat(m-ncol,2))
               Araout=2*pi*(r(j)+(delr/2))*deltay(nodmat(m-ncol,2))
               Arbin=2*pi*(r(j)-(delr/2))*deltay(nodmat(m+ncol,2))
               Arbout=2*pi*(r(j)+(delr/2))*deltay(nodmat(m+ncol,2))
               kn=props(nodmat(m-ncol,2),1)
               cpn=props(nodmat(m-ncol,2),2)
               rhon=props(nodmat(m-ncol,2),3)
               ks=props(nodmat(m+ncol,2),1)
               cps=props(nodmat(m+ncol,2),2)
               rhos=props(nodmat(m+ncol,2),3)
      A=(rhon*cpn*deltay(nodmat(m-ncol,2)))+
     +  (rhos*cps*deltay(nodmat(m+ncol,2)))
               B=((delt)/(pi*r(j)*delr))*
     +  ((delr/(Arain*kn)+delr/(Arbin*ks))/
     +  ((delr/(Arain*kn))*(delr/(Arbin*ks))))
               C=((delt)/(pi*r(j)*delr))*
     +  ((delr/(Araout*kn)+delr/(Arbout*ks))/
     +  ((delr/(Araout*kn))*(delr/(Arbout*ks))))
               D=(2*ks*delt)/deltay(nodmat(m+ncol,2))
               E=(2*kn*delt)/deltay(nodmat(m-ncol,2))
     F=((kn*deltay(nodmat(m+ncol,2))+
     +       ks*deltay(nodmat(m-ncol,2))*delt)/
     +       (r(j)*delth(nodmat(m+ncol,2)))**2)
     G=((kn*deltay(nodmat(m+ncol,2))+
     +       ks*deltay(nodmat(m-ncol,2))*delt)/
     +       (r(j)*delth(nodmat(m+ncol,2)))**2)
               Tinp(m)=Tout(m)+
     +       (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +  (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
               Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
               Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
               Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
               Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
               Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
     Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
c              ----------------------------------
               If(adiabats.EQ.1)Then
                    If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                         Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                         Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G)/A
     Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                    Endif
                    If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                         Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                         Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+F+G)/A
     Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
     +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                    Endif
               Else
               Endif
                    Endif
```

```
                        If(nodint(m,2).EQ.2)Then
                    Ayain=pi*(r(j)**2-(r(j)-delr/2)**2)
                    Ayaout=Ayain
                    Aybin=pi*((r(j)+delr/2)**2-r(j)**2)
                    Aybout=Aybin
                    kw=props(nodmat(m-1,2),1)
                    cpw=props(nodmat(m-1,2),2)
                    rhow=props(nodmat(m-1,2),3)
                    ke=props(nodmat(m+1,2),1)
                    cpe=props(nodmat(m+1,2),2)
                    rhoe=props(nodmat(m+1,2),3)
                    A=(rhow*cpw*(r(j)-(delr/4))*deltay(nodmat(m-1,2)))+
     +       (rhoe*cpe*(r(j)+(delr/4))*deltay(nodmat(m+1,2)))
                    B=(kw*delt*(r(j)-(delr/2))*deltay(nodmat(m-1,2)))/delr**2
            C=(ke*delt*(r(j)+(delr/2))*deltay(nodmat(m+1,2)))/delr**2
                    D=((delt)/(2*pi*delr))*
     +       ((delr/(Ayain*kw)+delr/(Aybin*ke))/
     +       ((delr/(Ayain*kw))*(delr/(Aybin*ke))))
                    E=((delt)/(2*pi*delr))*
     +       ((delr/(Ayaout*kw)+delr/(Aybout*ke))/
     +       ((delr/(Ayaout*kw))*(delr/(Aybout*ke))))
                    Tinp(m)=Tout(m)
                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                    Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                    Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                    Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                    Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A                                    !
            Coe(m-396,j+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-396,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-396)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                        Endif
2701                Continue
2601        Continue
            End

            Subroutine Interior3(Coeff,Tinp,Tout,props,r,delr,deltay,delt,
     +pdim,nodmati,nodmat,nodint,adialow,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)
            Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
            Double Precision r(ncol),props(15,6),delr,deltay(15),delt
            Double Precision A,B,C,D,E,F,G,pi,Coe(396,396)
     Double Precision Aya,Ayb,Ayc,Ayd,Ara,Arb,Arc,Ard
            Double Precision knw,kne,ksw,kse,cpnw,cpne,cpsw,cpse
            Double Precision rhonw,rhone,rhosw,rhose,delth(neq)
            Double Precision Arain,Araout,Arbin,Arbout
            Double Precision Ayain,Ayaout,Aybin,Aybout
            Double Precision kn,ks,cpn,cps,rhon,rhos
            Double Precision kw,ke,cpw,cpe,rhow,rhoe
            Integer mnw,mne,msw,mse
            Integer pdim(15,6),nodmati(neq,2),nodmat(neq,2)
            Integer nrow,ncol,neq,mats,nodint(neq,2)
            Integer adialow,adiabats
            pi=3.14159265
            Do 2002 ii=1,mats
                    Do 2102 i=pdim(ii,1)+1,pdim(ii,3)-1
                            Do 2202 j=pdim(ii,2)+1,pdim(ii,4)-1
                            k=3
                                    m=j+(i-1)*ncol+(k-1)*nrow*ncol
                                    A=props(ii,2)*props(ii,3)
                                    B=(props(ii,1)*delt*(r(j)-delr/2))/
     +                               (delr**2*r(j))
                                    C=(props(ii,1)*delt*(r(j)+delr/2))/
     +                               (delr**2*r(j))
```

```
                                    D=(props(ii,1)*delt)/
     +                                (deltay(nodmat(m,2))**2)
                                    E=(props(ii,1)*delt)/
     +                                (deltay(nodmat(m,2))**2)
                                    F=(props(ii,1)*delt)/((r(j)*delth(m))**2)
                                    G=(props(ii,1)*delt)/((r(j)*delth(m))**2)
                                    Tinp(m)=Tout(m)+
     +                              (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
     +                              (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                          Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
                                    Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                                    Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                                    Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
                                    Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
c                   ------------------------------------
                    If(adiabats.EQ.1)Then
                              If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                    Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+F+G)/A
          Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                              Endif
                              If(m.EQ.j+((adialow-1)-1)*ncol+(k-1)*nrow*ncol)Then
                                    Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                    Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G)/A
          Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
          Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
          +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                              Endif
                    Else
                    Endif
c                   ------------------------------------
2202                          Continue
2102                Continue
2002      Continue

          Do 2602 i=2,nrow-1
                    Do 2702 j=2,ncol-1
                    k=3
                              m=j+(i-1)*ncol+(k-1)*nrow*ncol
                              If(m.NE.nodmati(m,1))Then
     mnw=(j-1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
          knw=props(nodmat(mnw,2),1)
          cpnw=props(nodmat(mnw,2),2)
          rhonw=props(nodmat(mnw,2),3)
     mne=(j+1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
          kne=props(nodmat(mne,2),1)
          cpne=props(nodmat(mne,2),2)
          rhone=props(nodmat(mne,2),3)
     msw=(j-1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
          ksw=props(nodmat(msw,2),1)
          cpsw=props(nodmat(msw,2),2)
          rhosw=props(nodmat(msw,2),3)
          mse=(j+1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
          kse=props(nodmat(mse,2),1)
          cpse=props(nodmat(mse,2),2)
          rhose=props(nodmat(mse,2),3)
```

```
      Aya=pi*(r(j)**2-(r(j)-delr/2)**2)
      Ayc=Aya
      Ayb=pi*((r(j)+delr/2)**2-r(j)**2)
      Ayd=Ayb
      Ara=2*pi*(r(j)-delr/2)*deltay(nodmat(mnw,2))
      Arc=2*pi*(r(j)-delr/2)*deltay(nodmat(msw,2))
      Arb=2*pi*(r(j)+delr/2)*deltay(nodmat(mne,2))
      Ard=2*pi*(r(j)+delr/2)*deltay(nodmat(mse,2))
      A=(rhonw*cpnw*(r(j)-delr/4)*deltay(nodmat(mnw,2)))+
+  (rhone*cpne*(r(j)+delr/4)*deltay(nodmat(mne,2)))+
+  (rhosw*cpsw*(r(j)-delr/4)*deltay(nodmat(msw,2)))+
+  (rhose*cpse*(r(j)+delr/4)*deltay(nodmat(mse,2)))
      B=((2*delt)/(pi*delr))*
+  ((delr/(Ara*knw)+delr/(Arc*ksw))/
+  ((delr/(Ara*knw))*(delr/(Arc*ksw))))
      C=((2*delt)/(pi*delr))*
+  ((delr/(Arb*kne)+delr/(Ard*kse))/
+  ((delr/(Arb*kne))*(delr/(Ard*kse))))
      D=((2*delt)/(pi*delr))*
+  (((deltay(nodmat(msw,2))/(ksw*Ayc))+
+  (deltay(nodmat(mse,2))/(kse*Ayd)))/
+  ((deltay(nodmat(msw,2))/(ksw*Ayc))*
+  (deltay(nodmat(mse,2))/(kse*Ayd))))
      E=((2*delt)/(pi*delr))*
+  (((deltay(nodmat(mnw,2))/(knw*Aya))+
+  (deltay(nodmat(mne,2))/(kne*Ayb)))/
+  ((deltay(nodmat(mnw,2))/(knw*Aya))*
+  (deltay(nodmat(mne,2))/(kne*Ayb))))
      F=((knw*deltay(nodmat(m+ncol,2))+
      +          ksw*deltay(nodmat(m-ncol,2))*delt)/
      +          (r(j)*delth(nodmat(m+ncol,2)))**2)
      G=((knw*deltay(nodmat(m+ncol,2))+
      +          ksw*deltay(nodmat(m-ncol,2))*delt)/
      +          (r(j)*delth(nodmat(m+ncol,2)))**2)
      Tinp(m)=Tout(m)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
+(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                  Endif
                  If(nodint(m,2).EQ.1)Then
            Arain=2*pi*(r(j)-(delr/2))*deltay(nodmat(m-ncol,2))
            Araout=2*pi*(r(j)+(delr/2))*deltay(nodmat(m-ncol,2))
            Arbin=2*pi*(r(j)-(delr/2))*deltay(nodmat(m+ncol,2))
            Arbout=2*pi*(r(j)+(delr/2))*deltay(nodmat(m+ncol,2))
            kn=props(nodmat(m-ncol,2),1)
            cpn=props(nodmat(m-ncol,2),2)
            rhon=props(nodmat(m-ncol,2),3)
            ks=props(nodmat(m+ncol,2),1)
            cps=props(nodmat(m+ncol,2),2)
            rhos=props(nodmat(m+ncol,2),3)
        A=(rhon*cpn*deltay(nodmat(m-ncol,2)))+
+       (rhos*cps*deltay(nodmat(m+ncol,2)))
            B=((delt)/(pi*r(j)*delr))*
+       ((delr/(Arain*kn)+delr/(Arbin*ks))/
+       ((delr/(Arain*kn))*(delr/(Arbin*ks))))
            C=((delt)/(pi*r(j)*delr))*
```

```
+       ((delr/(Araout*kn)+delr/(Arbout*ks))/
+       ((delr/(Araout*kn))*(delr/(Arbout*ks))))
                D=(2*ks*delt)/deltay(nodmat(m+ncol,2))
                E=(2*kn*delt)/deltay(nodmat(m-ncol,2))
                F=((kn*deltay(nodmat(m+ncol,2))+
      +           ks*deltay(nodmat(m-ncol,2))*2*delt)/
      +           (2*r(j)*delth(nodmat(m+ncol,2))))
                G=((kn*deltay(nodmat(m+ncol,2))+
      +           ks*deltay(nodmat(m-ncol,2))*2*delt)/
      +           (2*r(j)*delth(nodmat(m+ncol,2))))
                Tinp(m)=Tout(m)+
      +         (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
+     (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                If(adiabats.EQ.1)Then
                        If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G)/A
      Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(m.EQ.j+(adialow-1)*ncol+(k-1)*nrow*ncol)Then
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+F+G)/A
      Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                Else
                Endif
                        Endif

                        If(nodint(m,2).EQ.2)Then
                Ayain=pi*(r(j)**2-(r(j)-delr/2)**2)
                Ayaout=Ayain
                Aybin=pi*((r(j)+delr/2)**2-r(j)**2)
                Aybout=Aybin
                kw=props(nodmat(m-1,2),1)
                cpw=props(nodmat(m-1,2),2)
                rhow=props(nodmat(m-1,2),3)
                ke=props(nodmat(m+1,2),1)
                cpe=props(nodmat(m+1,2),2)
                rhoe=props(nodmat(m+1,2),3)
                A=(rhow*cpw*(r(j)-(delr/4))*deltay(nodmat(m-1,2)))+
+       (rhoe*cpe*(r(j)+(delr/4))*deltay(nodmat(m+1,2)))
                B=(kw*delt*(r(j)-(delr/2))*deltay(nodmat(m-1,2)))/delr**2
                C=(ke*delt*(r(j)+(delr/2))*deltay(nodmat(m+1,2)))/delr**2
                D=((delt)/(2*pi*delr))*
+       ((delr/(Ayain*kw)+delr/(Aybin*ke))/
+       ((delr/(Ayain*kw))*(delr/(Aybin*ke))))
                E=((delt)/(2*pi*delr))*
+       ((delr/(Ayaout*kw)+delr/(Aybout*ke))/
+       ((delr/(Ayaout*kw))*(delr/(Aybout*ke))))
```

```
                Tinp(m)=Tout(m)
                Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
        Coe(m-792,j+(i-1)*ncol+(k-1)*nrow*ncol-792)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-792,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
        +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-792,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-792)=
        +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-792,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-792)=
        +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-792,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-792)=
        +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                        Endif
2702            Continue
2602    Continue
        End

        Subroutine Interior4(Coeff,Tinp,Tout,pprops,r,delr,deltayy,delt,
    +ppdim,nodmati,nodmat,nodint,adiabats,
    +nrow,ncol,neq,mats,Coe,delth)
        Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
        Double Precision r(ncol),pprops(15,6),delr,deltayy(15),delt
        Double Precision A,B,C,D,E,F,G,pi,Coe(396,396),delth(neq)
     Double Precision Aya,Ayb,Ayc,Ayd,Ara,Arb,Arc,Ard
        Double Precision knw,kne,ksw,kse,cpnw,cpne,cpsw,cpse
        Double Precision rhonw,rhone,rhosw,rhose
        Double Precision Arain,Araout,Arbin,Arbout
        Double Precision Ayain,Ayaout,Aybin,Aybout
        Double Precision kn,ks,cpn,cps,rhon,rhos
        Double Precision kw,ke,cpw,cpe,rhow,rhoe
        Integer mnw,mne,msw,mse
        Integer ppdim(15,6),nodmati(neq,2),nodmat(neq,2)
        Integer nrow,ncol,neq,mats,nodint(neq,2)
        Integer adiabats
        pi=3.14159265
        Do 2003 ii=1,mats
                Do 2103 i=ppdim(ii,1)+1,ppdim(ii,3)-1
                        Do 2203 j=ppdim(ii,2)+1,ppdim(ii,4)-1
                        k=4
                                m=j+(i-1)*ncol+(k-1)*nrow*ncol
                                A=pprops(ii,2)*pprops(ii,3)
                                B=(pprops(ii,1)*delt*(r(j)-delr/2))/
    +                            (delr**2*r(j))
                                C=(pprops(ii,1)*delt*(r(j)+delr/2))/
    +                            (delr**2*r(j))
                                D=(pprops(ii,1)*delt)/
        +                             (deltayy(nodmat(m,2))**2)
                                E=(pprops(ii,1)*delt)/
        +                             (deltayy(nodmat(m,2))**2)
                                F=(pprops(ii,1)*delt)/((r(j)*delth(m))**2)
                                G=(pprops(ii,1)*delt)/((r(j)*delth(m))**2)
                                Tinp(m)=Tout(m)+
        +                               (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
        +                               (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
                                Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                                Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                                Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
                                Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
        Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
        +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
        Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
```

```
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
c                       ----------------------------------
                        If(adiabats.EQ.1)Then
                                If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+F+G)/A
            Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
            Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
            +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                                Endif
                        Else
                        Endif
c                       ----------------------------------
2203                            Continue
2103                    Continue
2003    Continue

            Do 2603 i=2,nrow-1
                        Do 2703 j=2,ncol-1
                        k=4
                                m=j+(i-1)*ncol+(k-1)*nrow*ncol

                                If(m.NE.nodmati(m,1))Then
    mnw=(j-1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
            knw=pprops(nodmat(mnw,2),1)
            cpnw=pprops(nodmat(mnw,2),2)
            rhonw=pprops(nodmat(mnw,2),3)
    mne=(j+1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
            kne=pprops(nodmat(mne,2),1)
            cpne=pprops(nodmat(mne,2),2)
            rhone=pprops(nodmat(mne,2),3)
    msw=(j-1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
            ksw=pprops(nodmat(msw,2),1)
            cpsw=pprops(nodmat(msw,2),2)
            rhosw=pprops(nodmat(msw,2),3)
            mse=(j+1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
            kse=pprops(nodmat(mse,2),1)
            cpse=pprops(nodmat(mse,2),2)
            rhose=pprops(nodmat(mse,2),3)
            Aya=pi*(r(j)**2-(r(j)-delr/2)**2)
            Ayc=Aya
            Ayb=pi*((r(j)+delr/2)**2-r(j)**2)
            Ayd=Ayb
            Ara=2*pi*(r(j)-delr/2)*deltayy(nodmat(mnw,2))
            Arc=2*pi*(r(j)-delr/2)*deltayy(nodmat(msw,2))
            Arb=2*pi*(r(j)+delr/2)*deltayy(nodmat(mne,2))
            Ard=2*pi*(r(j)+delr/2)*deltayy(nodmat(mse,2))
            A=(rhonw*cpnw*(r(j)-delr/4)*deltayy(nodmat(mnw,2)))+
       +  (rhone*cpne*(r(j)+delr/4)*deltayy(nodmat(mne,2)))+
       +  (rhosw*cpsw*(r(j)-delr/4)*deltayy(nodmat(msw,2)))+
       +  (rhose*cpse*(r(j)+delr/4)*deltayy(nodmat(mse,2)))
            B=((2*delt)/(pi*delr))*
       +  ((delr/(Ara*knw)+delr/(Arc*ksw))/
       +  ((delr/(Ara*knw))*(delr/(Arc*ksw))))
            C=((2*delt)/(pi*delr))*
       +  ((delr/(Arb*kne)+delr/(Ard*kse))/
       +  ((delr/(Arb*kne))*(delr/(Ard*kse))))
            D=((2*delt)/(pi*delr))*
       +  (((deltayy(nodmat(msw,2))/(ksw*Ayc))+
       +  (deltayy(nodmat(mse,2))/(kse*Ayd)))/
       +  ((deltayy(nodmat(msw,2))/(ksw*Ayc))*
       +  (deltayy(nodmat(mse,2))/(kse*Ayd))))
            E=((2*delt)/(pi*delr))*
       +  (((deltayy(nodmat(mnw,2))/(knw*Aya))+
       +  (deltayy(nodmat(mne,2))/(kne*Ayb)))/
       +  ((deltayy(nodmat(mnw,2))/(knw*Aya))*
       +  (deltayy(nodmat(mne,2))/(kne*Ayb))))
```

```
      F=((knw*deltayy(nodmat(m+ncol,2))+
      +           ksw*deltayy(nodmat(m-ncol,2))*delt)/
      +           (r(j)*delth(nodmat(m+ncol,2)))**2)
      G=((knw*deltayy(nodmat(m+ncol,2))+
      +           ksw*deltayy(nodmat(m-ncol,2))*delt)/
      +           (r(j)*delth(nodmat(m+ncol,2)))**2)
      Tinp(m)=Tout(m)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
+(G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                        If(nodint(m,2).EQ.1)Then
            Arain=2*pi*(r(j)-(delr/2))*deltayy(nodmat(m-ncol,2))
            Araout=2*pi*(r(j)+(delr/2))*deltayy(nodmat(m-ncol,2))
            Arbin=2*pi*(r(j)-(delr/2))*deltayy(nodmat(m+ncol,2))
            Arbout=2*pi*(r(j)+(delr/2))*deltayy(nodmat(m+ncol,2))
            kn=pprops(nodmat(m-ncol,2),1)
            cpn=pprops(nodmat(m-ncol,2),2)
            rhon=pprops(nodmat(m-ncol,2),3)
            ks=pprops(nodmat(m+ncol,2),1)
            cps=pprops(nodmat(m+ncol,2),2)
            rhos=pprops(nodmat(m+ncol,2),3)
        A=(rhon*cpn*deltayy(nodmat(m-ncol,2)))+
+       (rhos*cps*deltayy(nodmat(m+ncol,2)))
            B=((delt)/(pi*r(j)*delr))*
+       ((delr/(Arain*kn)+delr/(Arbin*ks))/
+       ((delr/(Arain*kn))*(delr/(Arbin*ks))))
            C=((delt)/(pi*r(j)*delr))*
+       ((delr/(Araout*kn)+delr/(Arbout*ks))/
+       ((delr/(Araout*kn))*(delr/(Arbout*ks))))
            D=(2*ks*delt)/deltayy(nodmat(m+ncol,2))
            E=(2*kn*delt)/deltayy(nodmat(m-ncol,2))
            F=((kn*deltayy(nodmat(m+ncol,2))+
      +           ks*deltayy(nodmat(m-ncol,2))*2*delt)/
      +           (2*r(j)*delth(nodmat(m+ncol,2))))
            G=((kn*deltayy(nodmat(m+ncol,2))+
      +           ks*deltayy(nodmat(m-ncol,2))*2*delt)/
      +           (2*r(j)*delth(nodmat(m+ncol,2))))
            Tinp(m)=Tout(m)+
      +           (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)+
+       (G/A)*Tout(j+(i-1)*ncol+((k-1)+1)*nrow*ncol)
            Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F+G)/A
            Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
            Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
            Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
            Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
```

```
c                       ------------------------------------
                        If(adiabats.EQ.1)Then
                                If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F+G)/A
              Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
              Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                                Endif
                        Else
                        Endif
                                Endif
                                If(nodint(m,2).EQ.2)Then
                        Ayain=pi*(r(j)**2-(r(j)-delr/2)**2)
                        Ayaout=Ayain
                        Aybin=pi*((r(j)+delr/2)**2-r(j)**2)
                        Aybout=Aybin
                        kw=pprops(nodmat(m-1,2),1)
                        cpw=pprops(nodmat(m-1,2),2)
                        rhow=pprops(nodmat(m-1,2),3)
                        ke=pprops(nodmat(m+1,2),1)
                        cpe=pprops(nodmat(m+1,2),2)
                        rhoe=pprops(nodmat(m+1,2),3)
                        A=(rhow*cpw*(r(j)-(delr/4))*deltayy(nodmat(m-1,2)))+
     +          (rhoe*cpe*(r(j)+(delr/4))*deltayy(nodmat(m+1,2)))
                        B=(kw*delt*(r(j)-(delr/2))*deltayy(nodmat(m-1,2)))/delr**2
                        C=(ke*delt*(r(j)+(delr/2))*deltayy(nodmat(m+1,2)))/delr**2
                        D=((delt)/(2*pi*delr))*
     +          ((delr/(Ayain*kw)+delr/(Aybin*ke))/
     +          ((delr/(Ayain*kw))*(delr/(Aybin*ke))))
                        E=((delt)/(2*pi*delr))*
     +          ((delr/(Ayaout*kw)+delr/(Aybout*ke))/
     +          ((delr/(Ayaout*kw))*(delr/(Aybout*ke))))
                        Tinp(m)=Tout(m)
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
              Coe(m-1188,j+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
              Coe(m-1188,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
              Coe(m-1188,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
              Coe(m-1188,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
              Coe(m-1188,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1188)=
              +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                                Endif
2703                    Continue
2603            Continue
                End

                Subroutine Interior5(Coeff,Tinp,Tout,pprops,r,delr,deltayy,delt,
     +ppdim,nodmati,nodmat,nodint,adiabats,
     +nrow,ncol,neq,mats,Coe,delth)
                Double Precision Coeff(neq,neq),Tinp(neq),Tout(neq)
                Double Precision r(ncol),pprops(15,6),delr,deltayy(15),delt
                Double Precision A,B,C,D,E,F,pi,Coe(396,396),delth(neq)
       Double Precision Aya,Ayb,Ayc,Ayd,Ara,Arb,Arc,Ard
                Double Precision knw,kne,ksw,kse,cpnw,cpne,cpsw,cpse
                Double Precision rhonw,rhone,rhosw,rhose
                Double Precision Arain,Araout,Arbin,Arbout
                Double Precision Ayain,Ayaout,Aybin,Aybout
                Double Precision kn,ks,cpn,cps,rhon,rhos
                Double Precision kw,ke,cpw,cpe,rhow,rhoe
                Integer mnw,mne,msw,mse
                Integer ppdim(15,6),nodmati(neq,2),nodmat(neq,2)
```

```
      Integer nrow,ncol,neq,mats,nodint(neq,2)
      Integer adiabats
      pi=3.14159265
      Do 2004 ii=1,mats
            Do 2104 i=ppdim(ii,1)+1,ppdim(ii,3)-1
                  Do 2204 j=ppdim(ii,2)+1,ppdim(ii,4)-1
                  k=5
                        m=j+(i-1)*ncol+(k-1)*nrow*ncol
                        A=pprops(ii,2)*pprops(ii,3)
                        B=(pprops(ii,1)*delt*(r(j)-delr/2))/
     +                   (delr**2*r(j))
                        C=(pprops(ii,1)*delt*(r(j)+delr/2))/
     +                   (delr**2*r(j))
                        D=(pprops(ii,1)*delt)/
     +                     (deltayy(nodmat(m,2))**2)
                        E=(pprops(ii,1)*delt)/
     +                     (deltayy(nodmat(m,2))**2)
                        F=(4*pprops(ii,1)*delt)/((r(j)*delth(m))**2)
                        Tinp(m)=Tout(m)+
     +                  (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
                        Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
                        Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                        Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                        Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
                        Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
c                 ------------------------------------
                  If(adiabats.EQ.1)Then
                        If(m.EQ.j+((5+1)-1)*ncol+(k-1)*nrow*ncol)Then
                              Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=0.
                              Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+F)/A
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                        Endif
                  Else
                  Endif
c                 ------------------------------------
2204                    Continue
2104              Continue
2004      Continue
      Do 2604 i=2,nrow-1
            Do 2704 j=2,ncol-1
            k=5
                  m=j+(i-1)*ncol+(k-1)*nrow*ncol
                  If(m.NE.nodmati(m,1))Then
    mnw=(j-1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
      knw=pprops(nodmat(mnw,2),1)
      cpnw=pprops(nodmat(mnw,2),2)
      rhonw=pprops(nodmat(mnw,2),3)
    mne=(j+1)+((i-1)-1)*ncol+(k-1)*nrow*ncol
      kne=pprops(nodmat(mne,2),1)
      cpne=pprops(nodmat(mne,2),2)
      rhone=pprops(nodmat(mne,2),3)
    msw=(j-1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
      ksw=pprops(nodmat(msw,2),1)
      cpsw=pprops(nodmat(msw,2),2)
      rhosw=pprops(nodmat(msw,2),3)
      mse=(j+1)+((i+1)-1)*ncol+(k-1)*nrow*ncol
      kse=pprops(nodmat(mse,2),1)
```

```
      cpse=pprops(nodmat(mse,2),2)
      rhose=pprops(nodmat(mse,2),3)
      Aya=pi*(r(j)**2-(r(j)-delr/2)**2)
      Ayc=Aya
      Ayb=pi*((r(j)+delr/2)**2-r(j)**2)
      Ayd=Ayb
      Ara=2*pi*(r(j)-delr/2)*deltayy(nodmat(mnw,2))
      Arc=2*pi*(r(j)-delr/2)*deltayy(nodmat(msw,2))
      Arb=2*pi*(r(j)+delr/2)*deltayy(nodmat(mne,2))
      Ard=2*pi*(r(j)+delr/2)*deltayy(nodmat(mse,2))
      A=(rhonw*cpnw*(r(j)-delr/4)*deltayy(nodmat(mnw,2)))+
+  (rhone*cpne*(r(j)+delr/4)*deltayy(nodmat(mne,2)))+
+  (rhosw*cpsw*(r(j)-delr/4)*deltayy(nodmat(msw,2)))+
+  (rhose*cpse*(r(j)+delr/4)*deltayy(nodmat(mse,2)))
      B=((2*delt)/(pi*delr))*
+  ((delr/(Ara*knw)+delr/(Arc*ksw))/
+  ((delr/(Ara*knw))*(delr/(Arc*ksw))))
      C=((2*delt)/(pi*delr))*
+  ((delr/(Arb*kne)+delr/(Ard*kse))/
+  ((delr/(Arb*kne))*(delr/(Ard*kse))))
      D=((2*delt)/(pi*delr))*
+  (((deltayy(nodmat(msw,2))/(ksw*Ayc))+
+  (deltayy(nodmat(mse,2))/(kse*Ayd)))/
+  ((deltayy(nodmat(msw,2))/(ksw*Ayc))*
+  (deltayy(nodmat(mse,2))/(kse*Ayd))))
      E=((2*delt)/(pi*delr))*
+  (((deltayy(nodmat(mnw,2))/(knw*Aya))+
+  (deltayy(nodmat(mne,2))/(kne*Ayb)))/
+  ((deltayy(nodmat(mnw,2))/(knw*Aya))*
+  (deltayy(nodmat(mne,2))/(kne*Ayb))))
      F=((knw*deltayy(nodmat(m+ncol,2))+
      +          ksw*deltayy(nodmat(m-ncol,2))*4*delt)/
      +          (r(j)*delth(nodmat(m+ncol,2)))**2)
      Tinp(m)=Tout(m)+
      +(F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
      Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
      Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
      Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
      Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
      Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
      Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
      Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
      +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                  Endif

                  If(nodint(m,2).EQ.1)Then
            Arain=2*pi*(r(j)-(delr/2))*deltayy(nodmat(m-ncol,2))
            Araout=2*pi*(r(j)+(delr/2))*deltayy(nodmat(m-ncol,2))
            Arbin=2*pi*(r(j)-(delr/2))*deltayy(nodmat(m+ncol,2))
            Arbout=2*pi*(r(j)+(delr/2))*deltayy(nodmat(m+ncol,2))
            kn=pprops(nodmat(m-ncol,2),1)
            cpn=pprops(nodmat(m-ncol,2),2)
            rhon=pprops(nodmat(m-ncol,2),3)
            ks=pprops(nodmat(m+ncol,2),1)
            cps=pprops(nodmat(m+ncol,2),2)
            rhos=pprops(nodmat(m+ncol,2),3)
        A=(rhon*cpn*deltayy(nodmat(m-ncol,2)))+
+       (rhos*cps*deltayy(nodmat(m+ncol,2)))
            B=((delt)/(pi*r(j)*delr))*
+       ((delr/(Arain*kn)+delr/(Arbin*ks))/
+       ((delr/(Arain*kn))*(delr/(Arbin*ks))))
            C=((delt)/(pi*r(j)*delr))*
+       ((delr/(Araout*kn)+delr/(Arbout*ks))/
```

```
     +       ((delr/(Araout*kn))*(delr/(Arbout*ks))))
                     D=(2*ks*delt)/deltayy(nodmat(m+ncol,2))
                     E=(2*kn*delt)/deltayy(nodmat(m-ncol,2))
                     F=((kn*deltayy(nodmat(m+ncol,2))+
             +       ks*deltayy(nodmat(m-ncol,2))*2*delt)/
             +       (r(j)*delth(nodmat(m+ncol,2))))
                     Tinp(m)=Tout(m)+
             +       (F/A)*Tout(j+(i-1)*ncol+((k-1)-1)*nrow*ncol)
                     Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E+F)/A
                     Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                     Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                     Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                     Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
             Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
     c               ------------------------------------
                     If(adiabats.EQ.1)Then
                             If(m.EQ.j+(5-1)*ncol+(k-1)*nrow*ncol)Then
                                     Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=0.
                                     Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+E+F)/A
             Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
                             Endif
                     Else
                     Endif
     c               ------------------------------------
                             Endif

                             If(nodint(m,2).EQ.2)Then
                     Ayain=pi*(r(j)**2-(r(j)-delr/2)**2)
                     Ayaout=Ayain
                     Aybin=pi*((r(j)+delr/2)**2-r(j)**2)
                     Aybout=Aybin
                     kw=pprops(nodmat(m-1,2),1)
                     cpw=pprops(nodmat(m-1,2),2)
                     rhow=pprops(nodmat(m-1,2),3)
                     ke=pprops(nodmat(m+1,2),1)
                     cpe=pprops(nodmat(m+1,2),2)
                     rhoe=pprops(nodmat(m+1,2),3)
                     A=(rhow*cpw*(r(j)-(delr/4))*deltayy(nodmat(m-1,2)))+
     +       (rhoe*cpe*(r(j)+(delr/4))*deltayy(nodmat(m+1,2)))
                     B=(kw*delt*(r(j)-(delr/2))*deltayy(nodmat(m-1,2)))/delr**2
                     C=(ke*delt*(r(j)+(delr/2))*deltayy(nodmat(m+1,2)))/delr**2
                     D=((delt)/(2*pi*delr))*
     +       ((delr/(Ayain*kw)+delr/(Aybin*ke))/
     +       ((delr/(Ayain*kw))*(delr/(Aybin*ke))))
                     E=((delt)/(2*pi*delr))*
     +       ((delr/(Ayaout*kw)+delr/(Aybout*ke))/
     +       ((delr/(Ayaout*kw))*(delr/(Aybout*ke))))
                     Tinp(m)=Tout(m)
                     Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)=(A+B+C+D+E)/A
                     Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)=-B/A
                     Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)=-C/A
                     Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)=-D/A
                     Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)=-E/A
             Coe(m-1584,j+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,j+(i-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
             +Coeff(m,(j-1)+(i-1)*ncol+(k-1)*nrow*ncol)
             Coe(m-1584,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol-1584)=
```

```
     +Coeff(m,(j+1)+(i-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i+1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i+1)-1)*ncol+(k-1)*nrow*ncol)
     Coe(m-1584,j+((i-1)-1)*ncol+(k-1)*nrow*ncol-1584)=
     +Coeff(m,j+((i-1)-1)*ncol+(k-1)*nrow*ncol)
                    Endif
2704            Continue
2604    Continue
     End

     Subroutine Intinterf(Tinp,Tout,props,deltay,delt,
    +nodmat,qinupp,qoutlow,ncol,nrow,neq)
     Double Precision Tinp(neq),Tout(neq)
     Double Precision props(15,4),deltay(15),delt
     Double Precision rhon,cpn,rhos,cps
     Double Precision qinupp(neq),qoutlow(neq)
     Integer nodmat(neq,2)
     Integer ncol,neq,nrow
     pi=3.14159265
     Do 3500 i=7,12,5
            Do 3700 j=2,17
                    k=1
                    m=j+(i-1)*ncol+(k-1)*nrow*ncol
            cpn=props(nodmat(m-ncol,2),2)
                    rhon=props(nodmat(m-ncol,2),3)
                    cps=props(nodmat(m+ncol,2),2)
                    rhos=props(nodmat(m+ncol,2),3)
                    A=(rhon*cpn*deltay(nodmat(m-ncol,2)))+
    +                (rhos*cps*deltay(nodmat(m+ncol,2)))
                    If(i.EQ.7)Then
                            Tinp(m)=Tout(m)+(qinupp(m)*delt)/A
                    Else
                            Tinp(m)=Tout(m)-(qoutlow(m)*delt)/A
                    Endif
3700            Continue
3500    Continue
     End

     Subroutine Intinterf2(Tinp,Tout,props,deltay,delt,
    +nodmat,qinupp,qoutlow,ncol,nrow,neq)
     Double Precision Tinp(neq),Tout(neq)
     Double Precision props(15,4),deltay(15),delt
     Double Precision rhon,cpn,rhos,cps
     Double Precision qinupp(neq),qoutlow(neq)
     Integer nodmat(neq,2)
     Integer ncol,neq,nrow
     pi=3.14159265
     Do 3501 i=7,12,5
            Do 3701 j=2,17
            k=2
                    m=j+(i-1)*ncol+(k-1)*nrow*ncol
                    cpn=props(nodmat(m-ncol,2),2)
                    rhon=props(nodmat(m-ncol,2),3)
                    cps=props(nodmat(m+ncol,2),2)
                    rhos=props(nodmat(m+ncol,2),3)
                    A=(rhon*cpn*deltay(nodmat(m-ncol,2)))+
    +                (rhos*cps*deltay(nodmat(m+ncol,2)))
                    If(i.EQ.7)Then
                            Tinp(m)=Tout(m)+(qinupp(m)*delt)/A
                    Else
                            Tinp(m)=Tout(m)-(qoutlow(m)*delt)/A
                    Endif
3701            Continue
3501    Continue
     End

     Subroutine Intinterf3(Tinp,Tout,props,deltay,delt,
    +nodmat,qinupp,qoutlow,ncol,nrow,neq)
     Double Precision Tinp(neq),Tout(neq)
     Double Precision props(15,4),deltay(15),delt
```

```
      Double Precision rhon,cpn,rhos,cps
      Double Precision qinupp(neq),qoutlow(neq)
      Integer nodmat(neq,2)
      Integer ncol,neq,nrow
      pi=3.14159265
      Do 3502 i=7,12,5
            Do 3702 j=2,17
                  k=3
                  m=j+(i-1)*ncol+(k-1)*nrow*ncol
                  cpn=props(nodmat(m-ncol,2),2)
                  rhon=props(nodmat(m-ncol,2),3)
                  cps=props(nodmat(m+ncol,2),2)
                  rhos=props(nodmat(m+ncol,2),3)
                  A=(rhon*cpn*deltay(nodmat(m-ncol,2)))+
     +             (rhos*cps*deltay(nodmat(m+ncol,2)))
                  If(i.EQ.7)Then
                        Tinp(m)=Tout(m)+(qinupp(m)*delt)/A
                  Else
                        Tinp(m)=Tout(m)-(qoutlow(m)*delt)/A
                  Endif
3702        Continue
3502  Continue
      End

      Subroutine Intinterf4(Tinp,Tout,pprops,deltayy,delt,
     +nodmat,qinupp,ncol,nrow,neq)
      Double Precision Tinp(neq),Tout(neq)
      Double Precision props(15,4),deltay(15),delt
      Double Precision rhon,cpn,rhos,cps
      Double Precision qinupp(neq)
      Integer nodmat(neq,2)
      Integer ncol,neq,nrow
      pi=3.14159265
      Do 3503 i=7,12,5
            Do 3703 j=2,17
                  k=4
                  m=j+(i-1)*ncol+(k-1)*nrow*ncol
                  cpn=pprops(nodmat(m-ncol,2),2)
                  rhon=pprops(nodmat(m-ncol,2),3)
                  cps=pprops(nodmat(m+ncol,2),2)
                  rhos=pprops(nodmat(m+ncol,2),3)
                  A=(rhon*cpn*deltayy(nodmat(m-ncol,2)))+
     +             (rhos*cps*deltayy(nodmat(m+ncol,2)))
                  If(i.EQ.7)Then
                        Tinp(m)=Tout(m)+(qinupp(m)*delt)/A
                  Else
                  Endif
3703        Continue
3503  Continue
      End

      Subroutine Intinterf5(Tinp,Tout,pprops,deltayy,delt,
     +nodmat,qinupp,ncol,nrow,neq)
      Double Precision Tinp(neq),Tout(neq)
      Double Precision props(15,4),deltay(15),delt
      Double Precision rhon,cpn,rhos,cps
      Double Precision qinupp(neq)
      Integer nodmat(neq,2)
      Integer ncol,neq,nrow
      pi=3.14159265
      Do 3504 i=7,12,5
            Do 3704 j=2,17
                  k=5
                  m=j+(i-1)*ncol+(k-1)*nrow*ncol
                  cpn=pprops(nodmat(m-ncol,2),2)
                  rhon=pprops(nodmat(m-ncol,2),3)
                  cps=pprops(nodmat(m+ncol,2),2)
                  rhos=pprops(nodmat(m+ncol,2),3)
                  A=(rhon*cpn*deltayy(nodmat(m-ncol,2)))+
     +             (rhos*cps*deltayy(nodmat(m+ncol,2)))
```

```
                        If(i.EQ.7)Then
                                Tinp(m)=Tout(m)+(qinupp(m)*delt)/A
                        Else
                        Endif
3704            Continue
3504    Continue
        End

        Subroutine Energy(Est,Einp,props,Qfix,Tout,Tavg,Ti,mnet,time,hfg,
     +tstep,pulse,r,delr,delt,Nfix,pdim,thmat,heffe,Te,Ebulkstep,Emats,
     +Ebulksum,Elost,nrow,ncol,neq,nsteps,mats,k,jj)
        Double Precision Est(nsteps+1,mats+3),Einp(nsteps+1),Qfix(10)
        Double Precision Tout(neq),props(15,4),Tavg(nsteps+1,mats)
        Double Precision mnet(nsteps+1),time(nsteps+1),hfg,tstep,pulse
        Double Precision r(ncol),delr,delt,vol,ro,ri,Am,Em,pi
        Double Precision Tsum,Ti,thmat(15),heffe(15),Te,sum1
        Double Precision Ebulkstep(mats,nsteps+1),Ebulksum(mats,2)
        Double Precision Elost(nsteps+1),Emats(nsteps+1)
        Integer Nfix(10),pdim(15,4),nrow,ncol,neq,nsteps,mats,k,jj
        pi=3.14159265
        If(tstep.LE.pulse)Then
                a=a+1
        Do i=1,ncol-11
                        m=Nfix(i)
                        j=m-(nrow-1)*ncol
                        ro=r(j)+(delr/2)
                        ri=r(j)-(delr/2)
                        Am=pi*(ro**2-ri**2)
                        Em=Qfix(i)*Am*time(jj+1)
                        Einp(jj+1)=Einp(jj+1)+Em
        Enddo
        Do i=17,(ncol-11)*2
                        m=Nfix(i)-396
                        j=m-(nrow-1)*ncol
                        ro=r(j)+(delr/2)
                        ri=r(j)-(delr/2)
                        Am=pi*(ro**2-ri**2)
                        Em=Qfix(i)*Am*time(jj+1)
                        Einp(jj+1)=Einp(jj+1)+Em
        Enddo
        Do i=33,(ncol-11)*3
                        m=Nfix(i)-792
                        j=m-(nrow-1)*ncol
                        ro=r(j)+(delr/2)
                        ri=r(j)-(delr/2)
                        Am=pi*(ro**2-ri**2)
                        Em=Qfix(i)*Am*time(jj+1)
                        Einp(jj+1)=Einp(jj+1)+Em
        Enddo
        Else
                Einp(jj+1)=Einp(a+1)
        Endif

c       If(tstep.LE.0.05)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                       !Column where flux is applied
c                       ro=r(j)+(delr/2)                        !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                        !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)                     !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)        !Energy in at each element                   [J]
c                       Einp(jj+1)=Einp(jj+1)+Em        !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                       !Column where flux is applied
c                       ro=r(j)+(delr/2)                        !Outer radius of input nodal element [m]
```

```
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Else
c       If(tstep.LE.0.1)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else

c       If(tstep.LE.0.15)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Else
c       If(tstep.LE.0.2)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.0.25)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
```

```
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Else
c       If(tstep.LE.0.3)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.0.35)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Else
c       If(tstep.LE.0.4)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.0.45)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
```

```
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Else
c       If(tstep.LE.0.5)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.0.55)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Else
c       If(tstep.LE.0.6)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
```

```
c       If(tstep.LE.0.65)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Else
c       If(tstep.LE.0.7)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse   [J]
c       Else
c       If(tstep.LE.0.75)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
```

```
c       Enddo

c          Else
c       If(tstep.LE.0.8)Then
c                 Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c                 Else
c       If(tstep.LE.0.85)Then
c                 a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                              !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                 !Column where flux is applied
c                       ro=r(j)+(delr/2)                  !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                  !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)               !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                 [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                  !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                 !Column where flux is applied
c                       ro=r(j)+(delr/2)                  !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                  !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)               !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                 [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                  !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                 !Column where flux is applied
c                       ro=r(j)+(delr/2)                  !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                  !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)               !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                 [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c          Else
c       If(tstep.LE.0.9)Then
c                 Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c                 Else
c       If(tstep.LE.0.95)Then
c                 a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                              !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                 !Column where flux is applied
c                       ro=r(j)+(delr/2)                  !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                  !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)               !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                 [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                  !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                 !Column where flux is applied
c                       ro=r(j)+(delr/2)                  !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                  !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)               !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                 [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                  !Nodes where heat flux in
applied
```

```
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                   [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c         Else
c       If(tstep.LE.1.0)Then
c               Einp(jj+1)=Einp(a+1)                     !Total stored energy after pulse    [J]
c               Else

c       If(tstep.LE.1.05)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                                !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                   [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                   [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                   [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c     Else
c       If(tstep.LE.1.1)Then
c               Einp(jj+1)=Einp(a+1)                     !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.1.15)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                                !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                   [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
```

```
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c   Else
c       If(tstep.LE.1.2)Then
c               Einp(jj+1)=Einp(a+1)                !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.1.25)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c         Else
c       If(tstep.LE.1.3)Then
c               Einp(jj+1)=Einp(a+1)                !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.1.35)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
```

```
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c               m=Nfix(i)-792                                !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c       Else
c       If(tstep.LE.1.4)Then
c               Einp(jj+1)=Einp(a+1)                         !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.1.45)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                            !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c               Am=pi*(ro**2-ri**2)                  !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Else
c       If(tstep.LE.1.5)Then
c               Einp(jj+1)=Einp(a+1)                         !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.1.55)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                            !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
```

```
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Else
c       If(tstep.LE.1.6)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.1.65)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Else
c       If(tstep.LE.1.7)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.1.75)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
```

```
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c   Else
c       If(tstep.LE.1.8)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.1.85)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                           !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element   [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo

c       Else
c       If(tstep.LE.1.9)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
```

```
c       If(tstep.LE.1.95)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo
c   Else
c       If(tstep.LE.2.0)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse  [J]
c       Else
c
c       If(tstep.LE.2.05)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)  !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em  !Summed energy in for all elements  [J]
c       Enddo
```

```
c       Else
c       If(tstep.LE.2.1)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.2.15)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo

c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c       Else
c       If(tstep.LE.2.2)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c       Else
c       If(tstep.LE.2.25)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
```

```
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Else
c       If(tstep.LE.2.3)Then
c               Einp(jj+1)=Einp(a+1)                !Total stored energy after pulse    [J]
c               Else
c
c       If(tstep.LE.2.35)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c         Else
c       If(tstep.LE.2.4)Then
c               Einp(jj+1)=Einp(a+1)                !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.2.45)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                       !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol           !Column where flux is applied
c                       ro=r(j)+(delr/2)            !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)            !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)         !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
```

```
c
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c
c         Else
c       If(tstep.LE.2.5)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse  [J]
c               Else
c       If(tstep.LE.2.55)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Else
c       If(tstep.LE.2.6)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse  [J]
c       Else
c       If(tstep.LE.2.65)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements  [J]
c       Enddo
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element  [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                  [J]
```

```
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo
c
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo
c       Else
c       If(tstep.LE.2.7)Then
c               Einp(jj+1)=Einp(a+1)                     !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.2.75)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                                !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo

c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo
c       Else
c       If(tstep.LE.2.8)Then
c               Einp(jj+1)=Einp(a+1)                     !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.2.85)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                                !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)              !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)     !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em     !Summed energy in for all elements   [J]
c       Enddo
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                    !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol                !Column where flux is applied
c                       ro=r(j)+(delr/2)                 !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                 !Inner radius of input nodal element [m]
```

```
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Else
c       If(tstep.LE.2.9)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c               Else
c       If(tstep.LE.2.95)Then
c               a=a+1
c       Do i=1,ncol-2
c                       m=Nfix(i)                               !Nodes where heat flux in applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c       Do i=17,(ncol-2)*2
c                       m=Nfix(i)-396                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c       Do i=33,(ncol-2)*3
c                       m=Nfix(i)-792                                   !Nodes where heat flux in
applied
c                       j=m-(nrow-1)*ncol               !Column where flux is applied
c                       ro=r(j)+(delr/2)                !Outer radius of input nodal element [m]
c                       ri=r(j)-(delr/2)                !Inner radius of input nodal element [m]
c                       Am=pi*(ro**2-ri**2)             !Area of each input nodal element    [m^2]
c                       Em=Qfix(i)*Am*time(jj+1)    !Energy in at each element                [J]
c                       Einp(jj+1)=Einp(jj+1)+Em    !Summed energy in for all elements   [J]
c       Enddo
c
c       Else
c       If(tstep.LE.3.0)Then
c               Einp(jj+1)=Einp(a+1)                    !Total stored energy after pulse    [J]
c               Else
c   Endif
c       Endif
c   Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
c       Endif
```

```
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif

c     Endif
c           Endif
c     Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c
c     Endif
c           Endif
c     Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif
c           Endif


            Do 800 ii=1,mats
                  ro=r(pdim(ii,4))
                  ri=r(pdim(ii,2))
                  vol=pi*(ro**2-ri**2)*thmat(ii)
                  Tsum=0.
                  kk=0
            Do k=1,3
                  Do i=pdim(ii,1),pdim(ii,3)
                        Do j=pdim(ii,2),pdim(ii,4)
                              kk=kk+1
                              m=j+(i-1)*ncol+(k-1)*nrow*ncol
                              Tsum=Tsum+Tout(m)
                        Enddo
                  Enddo
            Enddo
                  Tavg(jj+1,ii)=Tsum/kk
                  Est(jj+1,ii)=props(ii,2)*props(ii,3)*
     +      vol*(Tavg(jj+1,ii)-Ti)*1000
800         Continue
c           stop
            Est(jj+1,mats+3)=Einp(jj+1)*1000
```

```
 Est(jj+1,mats+1)=mnet(jj+1)*hfg*1000
        Esum=0.
        Do i=1,mats+1
                Esum=Est(jj+1,i)+Esum
        Enddo
        Emats(jj+1)=esum

        heffe(1)=135000
        heffe(2)=135000
        heffe(3)=0
        heffe(4)=0
        heffe(5)=0
        heffe(6)=135000
        heffe(7)=135000
        heffe(8)=0
        heffe(9)=0
        heffe(10)=0
        heffe(11)=0
        heffe(12)=0
        heffe(13)=0
        heffe(14)=0
        heffe(15)=0
        kk=0
        Do ii=1,mats
                If(heffe(ii).NE.0.)Then
                        kk=kk+1
                        Do k=1,3
                        m=ncol+(((pdim(ii,1)+pdim(ii,3))/2)-1)*ncol+(k-1)*
        +nrow*ncol
                        Ebulkstep(kk,jj+1)=heffe(ii)*2.*pi*r(ncol)*thmat(ii)*
+               (Tout(m)-Te)*delt
                        Enddo
                Endif
        Enddo
        Do nn=1,kk
                sum1=0.
                Do ii=1,jj+1
                        sum1=sum1+Ebulkstep(nn,ii)
                Enddo
                Ebulksum(nn,1)=nn
                Ebulksum(nn,2)=sum1
        Enddo

        sum1=0.
        Do nn=1,kk
                sum1=sum1+Ebulksum(nn,2)
        Enddo
        Elost(jj+1)=sum1*1000
        Est(jj+1,mats+2)=Elost(jj+1)
        If(Esum+Est(jj+1,mats+2).LE.Est(jj+1,mats+3))Then
                Ediff=Est(jj+1,mats+3)-(Esum+Est(jj+1,mats+2))
                Do ii=1,mats+2
                        Est(jj+1,ii)=Est(jj+1,ii)+Ediff*(Est(jj+1,ii)/
        +          (Emats(jj+1)+Est(jj+1,mats+2)))
                Enddo
        Endif
 End

        Subroutine Output1(Coeff,Tnodal,Tout,Temp,Est,time,deltay,r,
+pressg,deflg,volg,Tgasav,nrow,ncol,neq,nsteps,mats,skip,massgain)
        Double Precision Coeff(neq,neq),Tnodal(nrow,ncol),Tout(neq)
        Double Precision massgain(nsteps+1)
        Double Precision Temp(neq,neq),Est(nsteps+1,mats+3),time(nsteps+1)
        Double Precision pressg(nsteps+1),deflg(nsteps+1),volg(nsteps+1)
        Double Precision Tgasav(nsteps+1),deltay(15),r(ncol)
        Integer params(4),nrow,ncol,neq,nsteps,mats,skip
        k=1
        Do i=1,nrow
                Do j=1,ncol
                        m=j+(i-1)*ncol
```

```
                        Tnodal(i,j)=Tout(m)
                    Enddo
            Enddo

c           Do i=1,nrow
c                   Write(2,4200) (Tnodal(i,j)-273.,j=1,ncol)
c           Enddo
4200        Format(10000F16.4)

c           Do i=1,neq
c                   Write(3,4400) (Temp(i,j)-273.,j=1,nsteps+1,skip)
c           write(1013,4400) i,Tout(717),Tout(718)
c           write(1014,4400) i,Tout(719)
c           Enddo
4400        Format(10000F16.4)

C           Writes some parameters to an export file
            params(1)=neq
            params(2)=nrow
            params(3)=ncol
            params(4)=nsteps
            params(5)=nanagle
c           Do i=1,4
c                   Write(4,*) params(i)
c           Enddo

C           Writes time step and ideal gas information to a file
            Do i=1,nsteps+1,skip
c                   Write(5,4550) time(i)                           ![s]
c                   write(6,4500) pressg(i)/1000.   ![kPa]
c                   write(7,4500) Tgasav(i)-273.    ![C]
c                   write(8,4500) deflg(i)                          ![um]
c                   write(9,4550) volg(i)                   ![m^3]
            write(1012,4550) Temp(715,i),Temp(716,i)
            write(1013,4550) Temp(717,i),Temp(718,i)
            write(1014,4550) Temp(719,i)
            write(1016,4550) massgain(i)
            Enddo
4500        Format(10000F16.4)
4550        Format(10000E16.7E2)

C           Writes deltay for each material to a file
c           Do i=1,mats
c                   Write(10,4600) deltay(i)                ![m^3]
c           Enddo
4600        Format(10000E16.7E2)

C           Writes radial distance to a file
c           Do i=1,ncol
c                   Write(11,4700) r(i)                             ![m]
c           Enddo
4700        Format(10000E16.7E2)

c           Do i=1,nsteps+1,skip
c                   Write(12,4800) (Est(i,j),j=1,mats+3)
c           Enddo
4800        Format(10000E16.7E2)
            End

C           ------------------------------------------------------------------------------
C           CALCULATES MASS TRANSFER IN CAVITY
            Subroutine Evaporation(Pressavg,Tempavg,Pcav,Tcav,mgen,mtot,
     +mnet,mfluxlow,mevaplow,mfluxupp,mcondupp,mevapsum,mevaptot,
     +mcondsum,mcondtot,Tlvlow,Tlvupp,qoutlow,qinupp,
     +props,r,Tout,thmat,mol,hfg,delr,delt,delth,
            +minil,rhol,Liqlow,Liqupp,mremlow,mremupp,edgel,
            +Pgage,Deflupp,Volupp,volcoeff,evpenrg,conenrg,
     +matll,adialow,mats,distr,hcell,
            +Defllow,Vollow,hcellupp,hcelllow,sum5,deltaP,
     +ii1,ii2,ii3,ncol,nrow,neq,nsteps,jj,memtype)
```

```
      Double Precision Pressavg(nsteps+1),Tempavg(nsteps+1)
      Double Precision Pcav(ncol,nsteps+1),Tcav(ncol,nsteps+1)
      Double Precision mgen(ncol,nsteps+1),mtot(ncol,nsteps+1)
      Double Precision mnet(nsteps+1),minil(ncol),deltaP(ncol,nsteps+1)
      Double Precision mfluxlow(ncol,nsteps+1),mevaplow(ncol,nsteps+1)
      Double Precision mfluxupp(ncol,nsteps+1),mcondupp(ncol,nsteps+1)
      Double Precision mevapsum(nsteps+1),mevaptot(ncol,nsteps+1)
      Double Precision mcondsum(nsteps+1),mcondtot(ncol,nsteps+1)
      Double Precision Tlvlow(ncol,nsteps+1),Tlvupp(ncol,nsteps+1)
      Double Precision qoutlow(neq),qinupp(neq),props(15,4)
      Double Precision r(ncol),Tout(neq),thmat(mats)
      Double Precision mol,hfg,delr,delt,delth(neq)
      Double Precision sum1,sum2,sum3,sum4,rhol,Aliql,Aliqu
      Double Precision Liqlow(nsteps+1),Liqupp(nsteps+1)
      Double Precision mremlow(nsteps+1),mremupp(nsteps+1)
      Double Precision Titer(100000),pi,Ac,Fxa,Fxb,sump,sumt
      Double Precision Pgage(nsteps+1),Deflupp(nsteps+1)
      Double Precision Diter(100000),Volupp(nsteps+1)
      Double Precision evpenrg(nsteps+1),conenrg(nsteps+1)
      Double Precision volcoeff,edgel,sum5(ncol,nsteps+1)
      Double Precision distr,xy,hcell(ncol,nsteps+1)
      Double Precision hcellupp(ncol,nsteps+1),hcelllow(ncol,nsteps+1)
      Double Precision Defllow(nsteps+1),Vollow(nsteps+1)
      Integer matll,adialow,mats
      Integer ii1,ii2,ii3,ncol,neq,nsteps,jj,memtype,nrow

C     EVAPORATION AT LOWER SURFACE **************************
      pi=3.14159265
      sum2=0.
      sum3=0.
      sum4=0.
      ii1=0
      i=12
      memtype=1
      Do j=2,17
            sum1=0.
            ii1=ii1+1
            m=j+(i-1)*ncol
            Ac=2.*pi*r(j)*delr*0.5
        sum5(ii1,jj+1)=((0.12)/(8*delth(m)))*(8*delth(m))**2
            deltaP(ii1,jj+1)=(sum5(ii1,jj+1)-sum5(ii1+1,jj+1))*(Ac)**2/
     +      (0.0018*delr)
            Titer(1)=Tcav(ii1,jj)-10
            Titer(2)=Tcav(ii1,jj)+10

            Do kk=2,100000
                  Fxa=((sqrt(mol/(2.*pi*8.314)))*((((1.101E-6*(Titer(kk)-
     +      273)**4+2.942E-5*(Titer(kk)-273)**3+.01648*(Titer(kk)-
     +      273)**2+.3891*(Titer(kk)-273)+8.966)*1000.)/sqrt(Titer(kk)
     +      ))-(Pcav(ii1,jj)/sqrt(Tcav(ii1,jj))))*hfg*Ac*1.)+
     +      (props(matll,1)*Ac*(Titer(kk)-
     +      Tout(j+(adialow-1)*ncol)))/thmat(matll)
                  Fxb=((sqrt(mol/(2.*pi*8.314)))*((((1.101E-6*(Titer(kk-1)-
     +      273)**4+2.942E-5*(Titer(kk-1)-273)**3+.01648*(Titer(kk-1)-
     +      273)**2+.3891*(Titer(kk-1)-273)+8.966)*1000.)/
     +      sqrt(Titer(kk-1)))-(Pcav(ii1,jj)/sqrt(Tcav(ii1,jj))))*
     +      hfg*Ac*1.)+(props(matll,1)*Ac*(Titer(kk-1)-
     +      Tout(j+(adialow-1)*ncol)))/thmat(matll)
                  Titer(kk+1)=Titer(kk)-
     +      ((Fxa*(Titer(kk)-Titer(kk-1)))/(Fxa-Fxb))
                        If(ABS(Titer(kk+1)-Titer(kk)).LT..00001) Goto 5
            Enddo
5           Continue
                  Tlvlow(ii1,jj+1)=Titer(kk+1)
            mfluxlow(ii1,jj+1)=((sqrt(mol/(2.*pi*8.314)))*((((1.101E-6*
     +      (Titer(kk+1)-273)**4+2.942E-5*(Titer(kk+1)-273)**3+.01648*
     +  (Titer(kk+1)-273)**2+.3891*(Titer(kk+1)-273)+8.966)*1000.)/
     +  sqrt(Titer(kk+1)))-(Pcav(ii1,jj)/sqrt(Tcav(ii1,jj)))))
            mevaplow(ii1,jj+1)=mfluxlow(ii1,jj+1)*Ac*delt
            sum5(ii1,jj+1)=(mevaplow(ii1,jj+1)*8.314*Tcav(ii1,jj))
```

```
     +          /(mol*Ac*hcell(ii1,jj))
                deltaP(ii1,jj+1)=(sum5(ii1,jj+1)-sum5(ii1+1,jj+1))*(Ac)**2/
     +          (0.0018*delr)
                Do ii=1,jj+1
                        sum1=sum1+mevaplow(ii1,ii)
                Enddo
                mevaptot(ii1,jj+1)=sum1
                If(mevaptot(ii1,jj+1).GE.minil(ii1))Then
                        mevaptot(ii1,jj+1)=minil(ii1)
                        mevaplow(ii1,jj+1)=0.
                        qoutlow(m)=0.
                Else
                        qoutlow(m)=mfluxlow(ii1,jj+1)*hfg
                Endif
                sum2=sum2+mevaptot(ii1,jj+1)
                mevapsum(jj+1)=sum2
          Enddo
          evpenrg(jj+1)=(mevapsum(jj+1)*hfg)       ![J]
          Do ii=2,17
          Enddo
          Do ii=1,ii1
                sum3=sum3+mevaplow(ii,jj+1)
          Enddo
          mremlow(jj+1)=mremlow(jj)-sum3
          Aliql=pi*((r(17)+delr/2)**2-(r(2)-delr/2)**2)
          thmat(matll)=(mremlow(jj+1))/(rhol*Aliql)
          Liqlow(jj+1)=thmat(matll)
          sum2=0.
          sum3=0.
          ii2=0
          i=7
          Do j=2,17
                sum1=0.
                ii2=ii2+1
                m=j+(i-1)*ncol

                Ac=2.*pi*r(j)*delr
                Titer(1)=Tcav(ii2,jj)-10
                Titer(2)=Tcav(ii2,jj)+10
                Do kk=2,100000
                        Fxa=((sqrt(mol/(2.*pi*8.314)))*((Pcav(ii2,jj)/
     +          sqrt(Tcav(ii2,jj)))-(((1.101E-6*(Titer(kk)-
     +          273)**4+2.942E-5*(Titer(kk)-273)**3+.01648*(Titer(kk)-
     +          273)**2+.3891*(Titer(kk)-273)+8.966)*1000.)/sqrt(Titer(kk)
     +          )))*hfg*Ac*1.)+(props(3,1)*Ac*
     +          (Tout(j+(5-1)*ncol)-Titer(kk)))/thmat(3)
                        Fxb=((sqrt(mol/(2.*pi*8.314)))*((Pcav(ii2,jj)/
     +          sqrt(Tcav(ii2,jj)))-(((1.101E-6*(Titer(kk-1)-
     +          273)**4+2.942E-5*(Titer(kk-1)-273)**3+.01648*(Titer(kk-1)-
     +          273)**2+.3891*(Titer(kk-1)-273)+8.966)*1000.)/
     +          sqrt(Titer(kk-1))))*hfg*Ac*1.)+(props(3,1)*Ac*
     +          (Tout(j+(5-1)*ncol)-Titer(kk-1)))/
     +          thmat(3)
                        Titer(kk+1)=Titer(kk)-
     +          ((Fxa*(Titer(kk)-Titer(kk-1)))/(Fxa-Fxb))
                If(ABS(Titer(kk+1)-Titer(kk)).LT..00001) Goto 6000
                Enddo
6000            Continue
                Tlvupp(ii2,jj+1)=Titer(kk+1)
                mfluxupp(ii2,jj+1)=((sqrt(mol/(2.*pi*8.314)))*((Pcav(ii2,jj)/
     +          sqrt(Tcav(ii2,jj)))-(((1.101E-6*(Titer(kk+1)-
     +          273)**4+2.942E-5*(Titer(kk+1)-273)**3+.01648*(Titer(kk+1)-
     +          273)**2+.3891*(Titer(kk+1)-273)+8.966)*1000.)/
     +          sqrt(Titer(kk+1)))))
                mcondupp(ii2,jj+1)=mfluxupp(ii2,jj+1)*Ac*delt
                Do ii=1,jj+1
                        sum1=sum1+mcondupp(ii2,ii)
                Enddo
                mcondtot(ii2,jj+1)=sum1
                If(mcondtot(ii2,jj+1).GE.minil(ii2))Then
```

```
                        mcondtot(ii2,jj+1)=minil(ii2)
                        mcondupp(ii2,jj+1)=0.
                        qinupp(m)=0.
                Else
                        qinupp(m)=mfluxupp(ii2,jj+1)*hfg
                Endif
                sum2=sum2+mcondtot(ii2,jj+1)
                mcondsum(jj+1)=sum2
        Enddo
        conenrg(jj+1)=mcondsum(jj+1)*hfg
        Do ii=1,ii2
                sum3=sum3+mcondupp(ii,jj+1)
        Enddo
        mremupp(jj+1)=mremupp(jj)+sum3
        Aliqu=pi*((r(17)+delr/2)**2-(r(2)-delr/2)**2)
        thmat(3)=(mremupp(jj+1))/(rhol*Aliqu)
        Liqupp(jj+1)=thmat(3)
        ii=0
        Do i=2,17
                ii=ii+1
                mgen(ii,jj+1)=mevaptot(ii,jj+1)-mcondtot(ii,jj+1)
        Enddo
        ii=0
        Do i=2,17
                ii=ii+1
                mtot(ii,jj+1)=mgen(ii,jj+1)+mgen(ii,1)
        Enddo
        mnet(jj+1)=mevapsum(jj+1)-mcondsum(jj+1)
        ii3=0
        Do j=2,17
                ii3=ii3+1
                Ac=2.*pi*r(j)*delr
                Titer(1)=Tcav(ii3,jj)-10
                Titer(2)=Tcav(ii3,jj)+10
                Do kk=2,100000
                        Fxa=((1.101E-6*(Titer(kk)-273)**4+2.942E-5*
     +          (Titer(kk)-273)**3+.01648*(Titer(kk)-273)**2+
     +          .3891*(Titer(kk)-273)+8.966)*1000.)-((mtot(ii3,jj+1)*
     +          8.314*Tcav(ii3,jj))/(mol*Ac*hcell(ii,jj)))
                        Fxb=((1.101E-6*(Titer(kk-1)-273)**4+2.942E-5*
     +          (Titer(kk-1)-273)**3+.01648*(Titer(kk-1)-273)**2+
     +          .3891*(Titer(kk-1)-273)+8.966)*1000.)-((mtot(ii3,jj+1)*
     +          8.314*Tcav(ii3,jj))/(mol*Ac*hcell(ii,jj)))
                        Titer(kk+1)=Titer(kk)-
     +          ((Fxa*(Titer(kk)-Titer(kk-1)))/(Fxa-Fxb))
                        If(ABS(Titer(kk+1)-Titer(kk)).LT..00001) Goto 6200
                Enddo
6200            Continue
                Pcav(ii3,jj+1)=(mtot(ii3,jj+1)*8.314*Tcav(ii3,jj))/
     +          (mol*Ac*hcell(ii,jj))
                Tcav(ii3,jj+1)=Titer(kk+1)
        Enddo
        ii=0
        sump=0.
        sumt=0.
        Do i=2,17
                ii=ii+1
                sump=sump+Pcav(ii,jj+1)
                sumt=sumt+Tcav(ii,jj+1)
        Enddo
        Pressavg(jj+1)=sump/16                  ![abs Pa]
        Tempavg(jj+1)=sumt/16                   ![K]
        Pgage(jj+1)=(Pressavg(jj+1)-Pressavg(1)+Pgage(1))
                Diter(1)=Deflupp(jj)
                Diter(2)=Deflupp(jj)+10
                Do kk=2,100000
                        If(edgel.EQ.3.E-3)Then
                                If(memtype.EQ.1)Then
                        Fxa=(1.31699E-4*Diter(kk)**3-4.05487E-5*Diter(kk)**2+
     +                  .00723331*Diter(kk))-((Pgage(jj+1))/1000)
```

```
                    Fxb=(1.31699E-4*Diter(kk-1)**3-4.05487E-5*Diter(kk-1)**2+
     +              .00723331*Diter(kk-1))-((Pgage(jj+1))/1000)
                        Endif
                    If(memtype.EQ.2)Then
                    Fxa=(2.45945E-5*Diter(kk)**3-1.78599E-5*Diter(kk)**2+
     +          .00526537*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(2.45945E-5*Diter(kk-1)**3-1.78599E-5*Diter(kk-1)**2+
     +          .00526537*Diter(kk-1))-((Pgage(jj+1))/1000)
                        Endif
                    If(memtype.EQ.3)Then
                    Fxa=(1.39505E-4*Diter(kk)**3-1.30452E-4*Diter(kk)**2+
     +          .00526537*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(1.39505E-4*Diter(kk-1)**3-1.30452E-4*Diter(kk-1)**2+
     +          .00526537*Diter(kk-1))-((Pgage(jj+1))/1000)
                        Endif
                    Endif
                    If(edgel.EQ.4.E-3)Then
                    Fxa=(4.51435E-5*Diter(kk)**3-2.6698E-4*Diter(kk)**2+
     +              .13029*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(4.51435E-5*Diter(kk-1)**3-2.6698E-4*Diter(kk-1)**2+
     +              .13029*Diter(kk-1))-((Pgage(jj+1))/1000)
                    Endif
                    If(edgel.EQ.5.E-3)Then
                    Fxa=(1.59507E-5*Diter(kk)**3-8.63774E-6*Diter(kk)**2+
     +              .0300243*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(1.59507E-5*Diter(kk-1)**3-8.63774E-6*Diter(kk-1)**2+
     +              .0300243*Diter(kk-1))-((Pgage(jj+1))/1000)
                    Endif
                    Diter(kk+1)=Diter(kk)-
     +          ((Fxa*(Diter(kk)-Diter(kk-1)))/(Fxa-Fxb))

                    If(ABS(Diter(kk+1)-Diter(kk)).LT..00001) Goto 7000
                Enddo
7000            Continue
                Deflupp(jj+1)=Diter(kk+1)
        Volupp(jj+1)=volcoeff*Deflupp(jj+1)
                Diter(1)=Defllow(jj)
                Diter(2)=Defllow(jj)+10                         ]
                Do kk=2,100000
                    If(edgel.EQ.3.E-3)Then
                    Fxa=(1.39505E-4*Diter(kk)**3-1.30452E-4*Diter(kk)**2+
     +          .02525*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(1.39505E-4*Diter(kk-1)**3-1.30452E-4*Diter(kk-1)**2+
     +          .02525*Diter(kk-1))-((Pgage(jj+1))/1000)
                    Endif
                    If(edgel.EQ.4.E-3)Then
                    Fxa=(4.51435E-5*Diter(kk)**3-2.6698E-4*Diter(kk)**2+
     +              .13029*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(4.51435E-5*Diter(kk-1)**3-2.6698E-4*Diter(kk-1)**2+
     +              .13029*Diter(kk-1))-((Pgage(jj+1))/1000)
                    Endif
                    If(edgel.EQ.5.E-3)Then
                    Fxa=(1.59507E-5*Diter(kk)**3-8.63774E-6*Diter(kk)**2+
     +              .0300243*Diter(kk))-((Pgage(jj+1))/1000)
                    Fxb=(1.59507E-5*Diter(kk-1)**3-8.63774E-6*Diter(kk-1)**2+
     +              .0300243*Diter(kk-1))-((Pgage(jj+1))/1000)
                    Endif
                    Diter(kk+1)=Diter(kk)-
     +          ((Fxa*(Diter(kk)-Diter(kk-1)))/(Fxa-Fxb))

                    If(ABS(Diter(kk+1)-Diter(kk)).LT..00001) Goto 7100
                Enddo
7100            Continue
                Defllow(jj+1)=Diter(kk+1)

        Vollow(jj+1)=volcoeff*Defllow(jj+1)
        ii=0
        Do j=2,17
                ii=ii+1
                xy=r(j)
```

```
                    hcellupp(ii,jj+1)=(1.E-6*Deflupp(jj+1)*(1/(distr)**4)*
+       (((distr)**2)-xy**2)*(((distr)**2)-xy**2)*
+       ((1+(.34/(distr)**2)*(xy**2+xy**2))))
                    hcelllow(ii,jj+1)=1.E-6*Defllow(jj+1)*(1/(distr)**4)*
+       (((distr)**2)-xy**2)*(((distr)**2)-xy**2)*
+       ((1+(.34/(distr)**2)*(xy**2+xy**2)))
                    hcell(ii,jj+1)=hcellupp(ii,jj+1)+hcelllow(ii,jj+1)+
+                                           thmat(4)
 Enddo
 End
```