Subject: Intermediate Milestone: FLOW-3D YMUZ2 Version 1.0 Users Manual (IM 06002.01.262.770)

Dear Dr. Rubenstone:

This letter transmits a final report titled, "FLOW-3D YMUZ2 Version 1.0 Users Manual." All Nuclear Regulatory Commission (NRC) staff comments have been addressed. The previous draft report of the subject deliverable (IM 06002.01.262.770) was reviewed and accepted by NRC (Ticket CNWRA 2007 212). We have removed the draft designation from each page and are transmitting the report in final form.

Please call me at (210) 522-6418 or Dr. K. Das at (210) 522-4269 if you have any questions about the report.

Sincerely,

Robert Lenhard, Ph.D.
Manager, Hydrology

cc:
DHLWRS
D. DeMarco
S. Kim
B. Meehan
L. Kokajko
J. Davis
A. Mohseni
T. McCartin
B. Hill
M. Shah
J. Gutmann
S. Whaley
E. Peters
K. Stablein
M. Wong
D. Brooks
R. Fedors
J. Pohle
P. Justus
GED/CNWRA
W. Patrick
B. Sagar
C. Manepally
K. Das
SwRI
Letter only:
GED Directors and Managers
L. Gutierrez
Record Copy B, IQS

Washington Office • Twinbrook Metro Plaza #210
12300 Twinbrook Parkway • Rockville, Maryland 20852-1606
Ticket CNWRA 2007-212 is closed, phase 2

The report titled "FLOW-3D YMUZ2 Version 1 Users Guide" was revised to address minor NRC comments. His version of the report will not be made public because CNWRA designated the report as "Draft."

The next phase in this closed ticket is for CNWRA to provide a version of the file with the word "Draft" removed from each page.

-Randy Fedors
JZ2 Project Officer
ABSTRACT

Estimating the performance of a potential high-level nuclear waste repository at Yucca Mountain, Nevada, requires an in-depth knowledge of the surrounding conditions inside the drift that stores the waste packages. The in-drift environment is influenced by complex processes like multimode heat transfer, phase change, air flow, and moisture transport caused by the dissipation of decay heat from the radioactive waste and availability of liquid water in the drift wall. Numerical simulation of this coupled flow and heat transfer process requires a robust multi component flow solver that accurately represents the in-drift flow. The general purpose computational fluid dynamics code *FLOW-3D*® is a widely used and validated tool in the engineering community. It has been used successfully to study various engineered and natural systems including single phase free convection flows. The standard flow solver *FLOW-3D* Version 9.0, however, does not have the capability to model radiation heat transfer or moisture transport that is required to simulate the in-drift environment in the repository. A number of customized subroutines were required to build two new modules for simulating the radiation heat transfer and moisture transport. The *FLOW-3D* solver with this enhanced capability to model multimode heat transfer with radiation and moisture transport is named *FLOW-3D YMUZ2* Version 1.0. Understanding moisture transport and condensation is necessary to estimate the quantity and chemistry of water that may lead to localized corrosion of the engineered barrier system. The solver has been validated for a number of benchmark problems and was used to perform simulations for in-drift heat transfer and multiphase flows. This Users Manual describes the Moisture Transport and Radiation Modules and the relevant input files, parameters necessary to use these modules, example problems, and installation procedures. The manual also delineates the theory used to model the radiation and moisture transport processes followed by brief guidelines to program the modules using standard FORTRAN 77 language.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>FIGURES</td>
<td>v</td>
</tr>
<tr>
<td>TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vii</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td>1.1 Overview of <em>FLOW-3D</em> Simulation Package</td>
<td>1-2</td>
</tr>
<tr>
<td>1.2 Moisture Transport Module</td>
<td>1-3</td>
</tr>
<tr>
<td>1.2.1 Assumptions of the Moisture Transport Module</td>
<td>1-3</td>
</tr>
<tr>
<td>1.2.2 Limitations of the Moisture Transport Module</td>
<td>1-3</td>
</tr>
<tr>
<td>1.3 Radiation Module</td>
<td>1-4</td>
</tr>
<tr>
<td>1.3.1 Assumptions of the Radiation Module</td>
<td>1-4</td>
</tr>
<tr>
<td>1.3.2 Limitations of the Radiation Module</td>
<td>1-4</td>
</tr>
<tr>
<td>2 INSTALLATION AND EXECUTION</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1 Hardware Requirements and Operating System for <em>FLOW-3D YMUZ2</em></td>
<td>2-1</td>
</tr>
<tr>
<td>2.2 Software Requirements for <em>FLOW-3D YMUZ2</em></td>
<td>2-1</td>
</tr>
<tr>
<td>2.3 Microsoft Windows® Installation Procedure</td>
<td>2-1</td>
</tr>
<tr>
<td>2.4 Code Execution</td>
<td>2-2</td>
</tr>
<tr>
<td>3 INPUT REQUIREMENTS</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1 Input Requirements for Standard <em>FLOW-3D</em></td>
<td>3-1</td>
</tr>
<tr>
<td>3.2 Input Requirements for <em>FLOW-3D YMUZ2</em></td>
<td>3-1</td>
</tr>
<tr>
<td>3.2.1 Input Requirements for the Moisture Transfer Module in <em>FLOW-3D YMUZ2</em></td>
<td>3-2</td>
</tr>
<tr>
<td>3.2.1.1 Computational Parameters (XPUT)</td>
<td>3-4</td>
</tr>
<tr>
<td>3.2.1.2 Fluid Properties (PROPS)</td>
<td>3-5</td>
</tr>
<tr>
<td>3.2.1.3 User-Defined Scalars (SCALAR)</td>
<td>3-5</td>
</tr>
<tr>
<td>3.2.1.3.1 Scalar-1</td>
<td>3-6</td>
</tr>
<tr>
<td>3.2.1.3.2 Scalar-2</td>
<td>3-7</td>
</tr>
<tr>
<td>3.2.1.3.3 Scalar-3</td>
<td>3-7</td>
</tr>
<tr>
<td>3.2.1.3.4 Scalar-4</td>
<td>3-8</td>
</tr>
<tr>
<td>3.2.1.3.5 Scalar-5</td>
<td>3-9</td>
</tr>
<tr>
<td>3.2.1.3.6 Scalar-6</td>
<td>3-9</td>
</tr>
<tr>
<td>3.2.1.3.7 Scalar-7</td>
<td>3-10</td>
</tr>
<tr>
<td>3.2.1.3.8 Scalar-8</td>
<td>3-11</td>
</tr>
<tr>
<td>3.2.1.4 Flow Field Initialization (FL)</td>
<td>3-11</td>
</tr>
</tbody>
</table>
## CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1.5 Temperature Field Initialization (TEMP)</td>
<td>3-12</td>
</tr>
<tr>
<td>3.2.1.6 User-Defined Parameters (USRDAT)</td>
<td>3-13</td>
</tr>
<tr>
<td>3.2.2 Input Requirements for Radiation Module in <em>FLOW-3D YMUZ2</em></td>
<td>3-14</td>
</tr>
<tr>
<td>3.2.2.1 Computational Parameters (XPUT)</td>
<td>3-14</td>
</tr>
<tr>
<td>3.2.2.2 User-Defined Parameters (USRDAT)</td>
<td>3-15</td>
</tr>
<tr>
<td>3.3 Output from <em>FLOW-3D YMUZ2</em></td>
<td>3-15</td>
</tr>
<tr>
<td>3.3.1 Output of Moisture Transport Module</td>
<td>3-17</td>
</tr>
<tr>
<td>3.3.2 Output of Radiation Module</td>
<td>3-17</td>
</tr>
<tr>
<td>4 SAMPLE PROBLEM</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1 Problem Description</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2 Test Cases</td>
<td>4-1</td>
</tr>
<tr>
<td>4.3 Input File for Convection With Moisture Transport and Radiation</td>
<td>4-3</td>
</tr>
<tr>
<td>4.4 Results</td>
<td>4-6</td>
</tr>
<tr>
<td>5 REFERENCES</td>
<td>5-1</td>
</tr>
</tbody>
</table>

**APPENDIX A** THEORETICAL BASIS OF MOISTURE TRANSPORT AND RADIATION MODULES

**APPENDIX B** PROGRAMMING GUIDE

**APPENDIX C** ADDITIONAL SAMPLE INPUT FILES
# FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Schematic for an Arrangement of Solid Bodies With Radiating Surfaces</td>
<td>3-15</td>
</tr>
<tr>
<td>4-1</td>
<td>Schematic for Convection, Radiation, and Mass Transfer in a Two-Dimensional Enclosure</td>
<td>4-2</td>
</tr>
<tr>
<td>4-2</td>
<td><em>FLOW-3D YMUZ2</em> Predictions for Temperature and Flow for Combined Convection, Radiation, and Moisture Transport</td>
<td>4-7</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2-1</td>
<td>Directory Contents of the Installation CD for <strong>FLOW-3D YMUZ2</strong></td>
<td>2-3</td>
</tr>
<tr>
<td>3-1</td>
<td>Input Namelist Blocks for <strong>FLOW-3D</strong> and <strong>FLOW-3D YMUZ2</strong></td>
<td>3-2</td>
</tr>
<tr>
<td>3-2</td>
<td>Computational Parameters for the Moisture Transport Module</td>
<td>3-4</td>
</tr>
<tr>
<td>3-3</td>
<td>Fluid Properties for the Moisture Transport Module</td>
<td>3-5</td>
</tr>
<tr>
<td>3-4</td>
<td>General Scalar Parameters for the Moisture Transport Module</td>
<td>3-6</td>
</tr>
<tr>
<td>3-5</td>
<td>Parameters for Scalar-1 in the Moisture Transport Module</td>
<td>3-6</td>
</tr>
<tr>
<td>3-6</td>
<td>Parameters for Scalar-2 in the Moisture Transport Module</td>
<td>3-7</td>
</tr>
<tr>
<td>3-7</td>
<td>Parameters for Scalar-3 in the Moisture Transport Module</td>
<td>3-8</td>
</tr>
<tr>
<td>3-8</td>
<td>Parameters for Scalar-4 in the Moisture Transport Module</td>
<td>3-8</td>
</tr>
<tr>
<td>3-9</td>
<td>Parameters for Scalar-5 in the Moisture Transport Module</td>
<td>3-9</td>
</tr>
<tr>
<td>3-10</td>
<td>Parameters for Scalar-6 in the Moisture Transport Module</td>
<td>3-10</td>
</tr>
<tr>
<td>3-11</td>
<td>Parameters for Scalar-7 in the Moisture Transport Module</td>
<td>3-10</td>
</tr>
<tr>
<td>3-12</td>
<td>Parameters for Scalar-8 in the Moisture Transport Module</td>
<td>3-11</td>
</tr>
<tr>
<td>3-13</td>
<td>Initialization Parameters for the Moisture Transport Module</td>
<td>3-12</td>
</tr>
<tr>
<td>3-14</td>
<td>Temperature Field Initialization for the Moisture Transport Module</td>
<td>3-13</td>
</tr>
<tr>
<td>3-15</td>
<td>User-Defined Parameters for the Moisture Transport Module</td>
<td>3-13</td>
</tr>
<tr>
<td>3-16</td>
<td>Computational Parameters for the Radiation Module</td>
<td>3-14</td>
</tr>
<tr>
<td>3-17</td>
<td>User-Defined Parameters for the Radiation Module</td>
<td>3-16</td>
</tr>
<tr>
<td>4-1</td>
<td>Fluid and Wall Properties for the Combined Heat and Moisture Transport Problem in an Enclosure</td>
<td>4-2</td>
</tr>
<tr>
<td>4-2</td>
<td>Moisture Transport Model Parameters Used in the Combined Heat and Moisture Transport Problem in an Enclosure</td>
<td>4-3</td>
</tr>
<tr>
<td>4-3</td>
<td><strong>FLOW-3D YMUZ2</strong> Results for 2-D Enclosure Heat and Mass Transfer Problem</td>
<td>4-8</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

This Users Manual documents work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the U.S. Nuclear Regulatory Commission (NRC) under Contract No. NRC–02–02–012. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards, Division of High-Level Waste Repository Safety. This manual is an independent product of the CNWRA and does not necessarily reflect the view or regulatory position of the NRC.

The authors acknowledge the technical review of D. Basu, the editorial review of L. Mulverhill, the programmatic review of G. Wittmeyer, and the assistance of J. Simpson in preparing this report.

QUALITY OF DATA, ANALYSES, AND CODE DEVELOPMENT

DATA: No data is presented in this report.

ANALYSES AND CODES: The FLOW-3D® Version 9.0 (Flow Sciences, Inc. 2005) fluid dynamics simulation code was used to develop the code FLOW-3D YMUZ2 under the software development procedures described in Geosciences and Engineering Division Technical Operating Procedure TOP–018. This controlled FLOW-3D YMUZ2 Version 1.0 is the initial release of this software. Information on development of this code is available in Scientific Notebook #536E and the companion Software Development plan. FLOW-3D Version 9.0 will be referred to simply as FLOW-3D, and FLOW-3D YMUZ2 Version 1.0 will be referred to as FLOW-3D YMUZ2.

REFERENCE:

1 INTRODUCTION

The in-drift thermal environment may affect long-term performance of the potential repository at Yucca Mountain, Nevada. The in-drift physical processes are characterized by complex heat transfer, phase change, and moisture redistribution processes. The decay heat of the radioactive waste inside the drift results in high waste package temperatures leading to multimode heat transfer, including conduction, convection, and radiation. This also causes a redistribution of liquid water (seepage) that could be present inside the drift. Liquid water evaporates at hotter locations, is carried in the vapor phase by convective air flow, and condenses at cooler locations, which is known as the cold trap process.

In-drift convection may significantly affect the gradients for temperature, relative humidity, and moisture redistribution. These conditions, in turn, affect the chemistry of water contacting the drip shield and waste package, corrosion of the engineered barrier system, and transport of radionuclides through the invert to the unsaturated zone below the drifts. Hence, identifying the source, distribution, and magnitude of water potentially contacting the engineered barrier system is important for assessing the repository performance.

Validated computational fluid dynamics (CFD) solvers provide a useful tool for simulating the in-drift convection and thermal processes. **FLOW-3D®** Version 9.0 (Flow Sciences, Inc., 2005) is a general purpose CFD simulation software package founded on the algorithms for simulating fluid flow that were developed at Los Alamos National Laboratory in the 1960s and 1970s. It has been widely used to solve technical problems ranging from basic hydraulics to micro-electro-mechanical devices. The standard version of **FLOW-3D**, however, does not have the capability to model radiative heat transfer or in-drift moisture redistribution processes. Additionally, the estimated range of waste package temperatures warrants inclusion of (Bechtel SAIC Company, LLC; 2005, Manepally, et. al., 2004) radiation as an important component of heat transfer. At the same time, the presence of liquid water on the drift wall will cause phase change and moisture transport that will affect both the temperature field and the distribution of moisture.

A robust and accurate flow simulation of the potential repository drift should account for these physical processes. To effectively use the standard **FLOW-3D** package for the numerical simulation of in-drift thermal and transport processes, new modules were developed to incorporate radiative heat transfer and moisture transport processes to specifically simulate the Yucca Mountain in-drift convection and heat transfer problem. The modified version of the **FLOW-3D** simulation package with these modules is called **FLOW-3D YMUZ2** Version 1.0.

The modified flow solver could be used as a tool to independently assess the approaches DOE is currently considering for calculating in-drift heat transfer and moisture redistribution in its performance assessment model. It could also be used to support, verify, and assess the near-field environment computations in the Total-system Performance Assessment (TPA) code. The **FLOW-3D YMUZ2** flow solver could also be used to carry out parametric studies to develop insights for the uncertainties in the near-field environment and in-drift physical processes.

---

1 Computational fluid dynamics (CFD) is referenced frequently throughout this report; consequently, the acronym CFD will be used.
This document details the Moisture Transport and Radiation Modules in FLOW-3D YMUZ2. The installation and execution procedure for FLOW-3D YMUZ2 is described in Chapter 2. Chapter 3 discusses the input parameters required for these modules, and Chapter 4 illustrates the use of FLOW-3D YMUZ2 through an example problem. The mathematical foundation of the FLOW-3D code modifications for including moisture transport and radiation heat transfer is presented in Appendix A. The coding methodology for implementing the mathematical models into the overall FLOW-3D program logic flow is described in Appendix B. Appendix C provides sample input files for a number of test cases.

1.1 Overview of the FLOW-3D Simulation Package

The governing equations for the standard version of FLOW-3D solver are the three-dimensional incompressible Navier-Stokes equations (Flow Sciences, Inc., 2005). FLOW-3D uses an ordered grid scheme that is oriented along a Cartesian or a polar cylindrical coordinate system. Fluid flow and heat transfer boundary conditions are applied at the six orthogonal mesh limit surfaces. The code uses the Fractional Area/Volume Obstacle Representation (FAVOR)™ method developed by Flow Sciences, Inc., to incorporate solid surfaces into the mesh structure and the computing equations. Three-dimensional solid objects are modeled as collections of blocked volumes and surfaces, to retain the advantages of solving the different equations on an orthogonal, structured grid.

Several spatial and temporal discretization schemes are available in FLOW-3D, and most of the terms in the Navier-Stokes equations are evaluated explicitly. The governing equations are approximated using a finite volume approach in an Eulerian framework describing the conservation of mass, momentum, and energy in a fluid. The code is capable of simulating two-fluid problems that include combinations of incompressible, compressible flow, laminar, and turbulent flows. The code has many auxiliary models for simulating non-Newtonian fluids, noninertial reference frames, porous media flows, casting processes surface tension effects, and thermo-elastic behavior. Several turbulence models are already available in FLOW-3D including the conventional k-ε model, the Renormalization Group (RNG), k-ε model. In general, all k-ε models use two transport equations. The RNG k-ε model is an improvement over the standard k-ε approach because it has an analytical expression for the turbulent Prandtl number, and it includes low Reynolds effects through an effective viscosity formulation. The standard version of FLOW-3D includes large eddy simulation (LES) for turbulence. In addition, it has a number of models to represent non-Newtonian fluid viscosity.

The code includes the Boussinesq approach to modeling buoyant fluids in an otherwise incompressible flow regime. The Boussinesq approximation neglects the effect of fluid density dependence on the pressure of the air phase, but includes the density dependence on temperature. This approach will be used extensively to represent in-drift air flow and heat transfer processes at Yucca Mountain.

Standard FLOW-3D has the option of customizing the solver to meet the unique requirements of a simulation. Solver customization could be done by modifying the source subroutines provided with the standard distribution. It is also possible to create new subroutines and link them with the solver. Both these techniques were used to develop the customized Moisture Transport and Radiation Modules of FLOW-3D YMUZ2.
1.2 Moisture Transport Module

The simulation of moisture transport in FLOW-3D requires that evaporation and condensation processes be modeled under high-humidity conditions. The key assumptions for the Moisture Transport Module are discussed in Section 1.2.1, and the limitations of this module are described in Section 1.2.2. The equations associated with the Moisture Transport Module along with formulations for energy equations and density evaluations are described in Appendix A. The method adopted to program this formulation is described in Appendix B.

1.2.1 Assumptions of the Moisture Transport Module

The assumptions used to represent high-humidity conditions in the Moisture Transport model are

- Air and water vapor are assumed to act as ideal gases with temperature-dependent density. This allows the use of the ideal gas equation of state to compute the mixture density as a function of temperature and composition for simulating the buoyancy effects on the fluid.

- A Boussinesq-like assumption is used where the density is assumed to be constant in the energy equation.

- Water can enter and exit the flow domain only at walls specified as sources or sinks.

- Walls not specified as sources or sinks can have condensate formation on them. This condensed water is available for reevaporation.

- The walls act as the source or sink of energy for the evaporation and condensation processes.

- Condensed “fog” acts as a mist that diffuses and advects like water vapor. When the relative humidity is limited to 100 percent, water can condense in the bulk of the flow domain. This water is not allowed to coalesce and “rain” out.

1.2.2 Limitations of the Moisture Transport Module

There are some limitations regarding the applicability of the Moisture Transport Module.

- The Moisture Transport Module has been formulated and tested only for high-humidity conditions near the wall. Its applicability for near wall low-humidity conditions has not been assessed.

- The water vapor and air phases are linked through an energy equation, as discussed in Appendix A, Section A1.3. The phase momentum equations are not solved separately. If the secondary phase has a high volume ratio, this assumption may lead to substantial deviation from the actual process.

- The liquid water in condensed form is treated as a continuous media and interdroplet interactions like breakup and coalescence are not considered. Hence, for processes like
sprayers and atomizers, where droplet—droplet interaction is significant, this module may only provide approximate results.

1.3 Radiation Module

The estimated waste package temperatures are expected to be above boiling for long durations (Bechtel SAIC LLC., 2005, Manepally et. al. 2004) because of emplaced high-level radioactive waste. At this temperature, thermal radiation significantly affects the overall heat transfer process, and inclusion of radiation is necessary to accurately estimate the in-drift convection and heat transfer. The radiation heat transfer pattern encountered in the drift is similar to the radiative exchange between surfaces in an enclosure with radiation frequency in the infrared range of the electromagnetic spectrum. A number of well-established techniques are available to solve radiation problems in an enclosure. The \textit{FLOW-3D YMUZ2} adopts the standard technique to solve radiation exchange between gray diffuse surfaces using a net radiation method.

1.3.1 Assumptions of the Radiation Module

The assumptions for the Radiation Module in \textit{FLOW-3D YMUZ2} are the following.

- All surfaces are diffuse and gray. The spectral characteristics of the surfaces are ignored by assuming that the surface emissivity is uniform and at a constant value. Each surface can, however, have a different value of emissivity.

- The gas is transparent to radiation and does not take part in the radiative heat transfer process.

1.3.2 Limitations of the Radiation Module

The limitations of the Radiation Module are the following

- The radiation module assumes the fluid to be nonparticipating in the radiative process. This is a reasonable assumption for the in-drift heat transfer in the repository. However, if the user intends to use it in other situations where the fluid media is not transparent to radiation, this module will tend to overpredict surface temperature and underpredict fluid temperature.

- The radiative heat transfer formulation assumes an enclosed surface with uniform radiosity. The user should use proper judgment when using this module for problems that do not have enclosed surfaces.

The Moisture Transport and Radiation Modules have been validated for natural convection flows (Green, 2006). The intended use of \textit{FLOW-3D YMUZ2} is for simulating the in-drift convection and heat transfer processes at Yucca Mountain. Users should use caution and judgment if they intend to apply these modules to other problems.
2 INSTALLATION AND EXECUTION

This chapter describes the installation and execution procedure of *FLOW-3D YMUZ2* in Microsoft® Windows XP® (Microsoft Corp., 2002a) operating system based on standard installation of *FLOW-3D*.

2.1 Hardware Requirements and Operating System for *FLOW-3D YMUZ2*

The standard *FLOW-3D* program is designed to be run on computers running the Windows or Linux/UNIX operating systems as described in the *FLOW-3D* manual (Flow Sciences, Inc., 2005). The hardware requirements for *FLOW-3D YMUZ2* are similar to *FLOW-3D*.

2.2 Software Requirements for *FLOW-3D YMUZ2*

*FLOW-3D* can be installed to operate in a WINDOWS or Linux/UNIX environment. The choice of operating system is made at the time the software is installed. It is understood that users of *FLOW-3D YMUZ2* will have the standard version installed on the platform of their choice. Intended users of *FLOW-3D YMUZ2* should have the administrative privileges to modify the installation directory of *FLOW-3D*.

The software described here was developed in accordance with the guidelines provided by *FLOW-3D* for modifying existing code modifications and adding new subroutines (Green, 2006). Pertinent guidelines are included in the description of each subroutine in the following sections. The software was written for use on a Windows XP platform using Compaq Visual FORTRAN Version 6.6c in accordance with the *FLOW-3D* guidelines. *FLOW-3D YMUZ2* is compatible with platforms and compilers that are supported by standard *FLOW-3D* with only two exceptions. First, an array that uses dynamic memory allocation was added. The declarations for dynamic memory allocation are compiler specific. Second, some features of the OPEN statements used to handle the radiation heat transfer output file are also compiler specific. These features will require modification based on the choice of compiler and platform. Other code modifications could be necessary if the software is implemented on other platforms or with other compilers. The *FLOW-3D YMUZ2* solver was extensively tested and validated on the Windows operating system with Compaq Visual FORTRAN Version 6.6c.

2.3 Microsoft Windows® Installation Procedure

The building procedure for *FLOW-3D YMUZ2* is analogous to building any customized routine in the standard version of *FLOW-3D*. It requires a basic understanding of the installation directory structure of standard *FLOW-3D*. The directory structure is detailed in Flow Sciences, Inc., (2005). Standard installation of *FLOW-3D* Version 9.0 provides both single and double precision object files and modules for customization purposes. The Moisture Transport and Radiation Modules use the double precision object files for building the customized *FLOW-3D YMUZ2* solver to maintain a high degree of accuracy.
The contents of the installation CD for the Moisture Transport and Radiation Modules should have the directory structure described in Table 2-1.

The custom modules were written with Compaq Visual FORTRAN Version 6.6C and were linked to the **FLOW-3D** object files provided with the standard installation. The installation process includes the following steps:

1. Install **FLOW-3D** Version 9.0 standard release.

2. Copy/Overwrite the directories as follows:
   a. From `QA_Control\prehyd\hydr3d` to `FLOW3D\prehyd\hydr3d`
   b. From `QA_Control\prehyd\prep3d` to `FLOW3D\prehyd\prep3d`
   c. From `QA_Control\source\comdeck` to `FLOW3D\source\comdeck`
   d. From `QA_Control\source\utility` to `FLOW3D\source\utility`
   e. From `QA_Control\source\prep3d` to `FLOW3D\source\prep3d`
   f. From `QA_Control\source\hydr3d` to `FLOW3D\source\hydr3d`

3. Initiate the Compaq Visual FORTRAN development environment.

4. Open the workspace: `FLOW3D\prehyd\hydr3d\hydr3d_rad`.

5. Build PREP3D.

6. Build HYDR3D.

7. Copy the executables as follows:
   a. From `FLOW3D\prehyd\hydr3d\release\hydr3d.exe` to `FLOW3D\double\hydr3d\release\hydr3d.exe`
   b. From `FLOW3D\prehyd\prep3d\release\prep3d.exe` to `FLOW3D\double\prep3d\release\prep3d.exe`

8. The flow solver **FLOW-3D YMUZ2** is now ready for operation.

Users can simulate some of the validation tests (Green, 2006) to ensure the solver has installed correctly. An example problem is described in Chapter 4.

### 2.4 Code Execution

**FLOW-3D YMUZ2** is executed in the same way as the standard version. Users have the option of using either the graphical user interface or the standard console. The detailed execution method is described in the **FLOW-3D** Version 9.0 Users Manual (Flow Sciences, Inc., 2005). The user should use the special input directives and keywords necessary to invoke the Moisture Transport and the Radiation Module. These special input keywords are described in Chapter 3.
<table>
<thead>
<tr>
<th>Directory</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA_Control\prehyd\hydr3d</td>
<td>hydr3d.dsp</td>
</tr>
<tr>
<td></td>
<td>hydr3d.dsw</td>
</tr>
<tr>
<td></td>
<td>hydr3d.ncb</td>
</tr>
<tr>
<td></td>
<td>hydr3d.opt</td>
</tr>
<tr>
<td></td>
<td>hydr3d.plg</td>
</tr>
<tr>
<td></td>
<td>hydr3d_rad.dsw</td>
</tr>
<tr>
<td></td>
<td>hydr3d_rad.opt</td>
</tr>
<tr>
<td>QA_Control\rehyd\prep3d</td>
<td>prep3d.dsp</td>
</tr>
<tr>
<td></td>
<td>prep3d.dsw</td>
</tr>
<tr>
<td></td>
<td>prep3d.ncb</td>
</tr>
<tr>
<td></td>
<td>prep3d.opt</td>
</tr>
<tr>
<td></td>
<td>prep3d.plg</td>
</tr>
<tr>
<td>QA_Control\source\comdeck</td>
<td>cbusr.f</td>
</tr>
<tr>
<td></td>
<td>pcg_rad_module.f</td>
</tr>
<tr>
<td></td>
<td>usrdat.f</td>
</tr>
<tr>
<td>QA_Control\source\utility</td>
<td>e1cal_stg.f</td>
</tr>
<tr>
<td></td>
<td>rhocal_stg.F</td>
</tr>
<tr>
<td>QA_Control\source\prep3d</td>
<td>prusrd_stg.F</td>
</tr>
<tr>
<td>QA_Control\source\hydr3d</td>
<td>matinv_stg.f</td>
</tr>
<tr>
<td></td>
<td>pcg_init_calc.f</td>
</tr>
<tr>
<td></td>
<td>qsadd_rad_pcg.f</td>
</tr>
<tr>
<td></td>
<td>rad_init_calc.f</td>
</tr>
<tr>
<td></td>
<td>rusrd_stg.F</td>
</tr>
<tr>
<td></td>
<td>teval_stg.f</td>
</tr>
</tbody>
</table>
3 INPUT REQUIREMENTS

This section describes the input entries required for FLOW-3D YMUZ2. The Moisture Transport and Radiation Modules of FLOW-3D YMUZ2 can either be used independently or together. FLOW-3D YMUZ2 requires a basic input file, prepin.*, used in the standard FLOW-3D package as a starting point. This basic input file could be developed using the FLOW-3D graphical user interface (GUI)\(^1\) or any available text editor. Subsequently, this input file has to be modified to include parameters related to the Moisture Transport and Radiation Modules of FLOW-3D YMUZ2. A very brief description of the FLOW-3D input specification is provided in Section 3.1 to help users start building the basic prepin file. Section 3.2 describes the input parameters for FLOW-3D YMUZ2 in detail. FLOW-3D YMUZ2 requires the user to be familiar with standard FLOW-3D Version 9.0. and have a degree of familiarity and understanding of the standard FLOW-3D. A brief description of the standard FLOW-3D input block will be provided with the same nomenclature and input blocks used for FLOW-3D YMUZ2 input.

3.1 Input Requirements for Standard FLOW-3D

The FLOW-3D preprocessor reads the problem definition from the project file, prepin.*, which contains a title for the simulation and a series of namelist blocks that define the problem setup and initial conditions. Most of the input keywords are specified through the standard GUI; however, some keyword specification cannot be done using the GUI and has to be typed using a text editor. The namelist blocks are separated into logical divisions, (e.g., boundary condition data is specified in namelist BCDATA and fluid properties are defined in namelist PROPS). The namelist blocks are summarized in the order required by the project file. Some of these blocks are optional. The namelist blocks are usually preceded by a couple of title lines. A brief description of the namelist blocks is provided in Table 3-1.

It can be seen that among the optional input blocks in standard FLOW-3D, the SCALAR and USRDAT namelist are required for FLOW-3D YMUZ2 as the customization makes extensive use of scalars.

3.2 Input Requirements for FLOW-3D YMUZ2

This section describes the keywords used for specifying inputs in FLOW-3D YMUZ2 that should be used in addition to the standard input parameters. Unlike the standard version, FLOW-3D YMUZ2 inputs cannot be specified using the GUI. The user has the option of using any standard text editor or a text editor supplied as an addendum with the GUI. The input requirement for Moisture Transport and Radiation Modules is discussed in two separate subsections. The input keywords for each module are listed and described under the namelist blocks of standard FLOW-3D.

\(^1\)Graphical user interface (GUI) is referenced throughout this report; consequently, the acronym GUI will be used.
3.2.1 Input Requirements for the Moisture Transfer Module in *FLOW-3D YMUZ2*

The Moisture Transport Module relies extensively on scalars and user-defined parameters. Table 3-1 indicates the required namelist blocks for the input files in *FLOW-3D YMUZ2*. The input keywords and the namelist blocks that are relevant for *FLOW-3D YMUZ2* are discussed in the following section. Detailed descriptions of all other input blocks are available in Flow Sciences, Inc., (2005).

<table>
<thead>
<tr>
<th>Namelist Block</th>
<th>Description</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUT</td>
<td>Contains the computational parameters as physical modeling options; selection of method for solving the equations are specified. A number of <em>FLOW-3D YMUZ2</em> parameters are specified in this namelist block.</td>
<td>Required</td>
</tr>
<tr>
<td>LIMITS</td>
<td>This namelist block lists the computational and output limits including screen prints.</td>
<td>Required</td>
</tr>
<tr>
<td>PROPS</td>
<td>Contains properties of all the fluids used in a problem.</td>
<td>Required</td>
</tr>
<tr>
<td>SCALAR</td>
<td>Specifies the scalar species parameters in standard <em>FLOW-3D</em>. The Moisture Transport and Radiation module extensively uses this namelist block.</td>
<td>Optional in standard version; required for <em>FLOW-3D YMUZ2</em></td>
</tr>
<tr>
<td>CHM</td>
<td>Used to specify chemistry parameters.</td>
<td>Optional in standard version; not required by <em>FLOW-3D YMUZ2</em></td>
</tr>
<tr>
<td>PCAP</td>
<td>Capillary pressure data for unsaturated porous media model are read using this optional namelist block.</td>
<td>Optional in standard version; not required by <em>FLOW-3D YMUZ2</em></td>
</tr>
<tr>
<td>RBDATA</td>
<td>Specifies coupled rigid body dynamics data.</td>
<td>Optional in standard version; not required by <em>FLOW-3D YMUZ2</em></td>
</tr>
<tr>
<td>Namelist Block</td>
<td>Description</td>
<td>Requirement</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BCDATA</td>
<td>Lists the boundary condition data. Needs to be specified for each mesh block separately.</td>
<td>Required</td>
</tr>
<tr>
<td>MESH</td>
<td>Specifies the mesh data. The computational domain could comprise multiple mesh blocks. The mesh data needs to be specified for each unit.</td>
<td>Required</td>
</tr>
<tr>
<td>OBS</td>
<td>Specifies the obstructions or geometries involved in the problem.</td>
<td>Required</td>
</tr>
<tr>
<td>FL</td>
<td>Specifies the fluid initial conditions for the problem. There could be multiple fluid zones in a problem, and each zone needs to be initialized separately.</td>
<td>Required</td>
</tr>
</tbody>
</table>
Table 3-1. Input Namelist Blocks for *FLOW-3D* (continued)

<table>
<thead>
<tr>
<th>Namelist Block</th>
<th>Description</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>Sets up the baffle.</td>
<td>Required</td>
</tr>
<tr>
<td>TEMP</td>
<td>Specifies initial temperature distribution for every fluid zone.</td>
<td>Required</td>
</tr>
<tr>
<td>MOTN</td>
<td>Noninertial reference frame motion description.</td>
<td>Optional in standard version; not required by <em>FLOW-3D YMUZ2</em></td>
</tr>
<tr>
<td>GRAFIC</td>
<td>Specifies the user request for graphical output.</td>
<td>Required</td>
</tr>
<tr>
<td>PART</td>
<td>Used for setting the marker and mass particles that are used for tracking the flow.</td>
<td>Required</td>
</tr>
<tr>
<td>USRDAT</td>
<td>Contains the user-specified data. If the variable ISURD is specified as equal to one in XPUT, this namelist block is read at the end of the input file. Otherwise, if the variable IUSRD has a value of two, it is read just after RBDATA.</td>
<td>Optional in standard version; required for <em>FLOW-3D YMUZ2</em>.</td>
</tr>
</tbody>
</table>

3.2.1.1 Computational Parameters (XPUT)

The input keywords for the XPUT namelist block are described in Table 3-2.

Table 3-2. Computational Parameters for the Moisture Transport Module (XPUT namelist block)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isurd</td>
<td>1</td>
<td>Unitless</td>
<td>User should always set the value equal to 1 when using <em>FLOW-3D YMUZ2</em>. Any other value will result in an error.</td>
<td>Indicates that the USRDAT namelist block must be created and read by the solver.</td>
</tr>
<tr>
<td>rmrhol</td>
<td>1</td>
<td>Unitless</td>
<td>= 0 for pure diffusion problems only = 1 all other cases</td>
<td>Dictates how mass diffusion affects the conservation equations.</td>
</tr>
<tr>
<td>rmrhoe</td>
<td>1</td>
<td>Unitless</td>
<td>= 0 for pure diffusion problems only = 1 all other cases</td>
<td>Dictates how mass diffusion affects the energy equations.</td>
</tr>
</tbody>
</table>
3.2.1.2 Fluid Properties (PROPS)

The input keywords for the PROPS namelist block describing the fluid properties are described in Table 3-3.

Table 3-3. Fluid Properties for the Moisture Transport Module (PROPS namelist block)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>rhof</td>
<td>0.8463</td>
<td>kg/m$^3$ [0.062 lb/ft$^3$]</td>
<td>Nominal mixture density for energy equation</td>
<td>If the simulation shows that the mixture temperature is not consistent with the specified nominal density, a better estimation may be required. The default may be changed based on the requirement.</td>
</tr>
<tr>
<td>mu1</td>
<td>2.3543 ×10$^{-5}$</td>
<td>N-s/m$^2$ [4.2 × 10$^{-7}$ lbf-s/ft$^2$]</td>
<td>Nominal mixture dynamic viscosity</td>
<td>If the simulation shows that the mixture temperature is not consistent with the specified nominal viscosity, a better estimation may be required. The default may be changed based on the requirement.</td>
</tr>
<tr>
<td>thc1</td>
<td>0.03484</td>
<td>W/m-K [0.02 BTU/hr-ft-°F]</td>
<td>Nominal mixture thermal conductivity</td>
<td>If the simulation shows that the mixture temperature is not consistent with the specified nominal thermal conductivity, a better estimation may be required. The default may be changed based on the requirement.</td>
</tr>
<tr>
<td>cv1</td>
<td>717 at 20 °C [0.172 at 68°F]</td>
<td>J/kg-K [BTU/lb-°F]</td>
<td>Constant volume-specific heat for dry air</td>
<td>This is a function of dry air temperature. The default may be changed based on the requirement.</td>
</tr>
</tbody>
</table>

3.2.1.3 User-Defined Scalars (SCALAR)

A SCALAR variable in FLOW-3D is a physical scalar quantity that is used to perform a number of functions ranging from storing a variable to solving for a flow parameter associated with the principal flow. A number of scalars, known as active scalars, are used by the solver for models
that come as addendum to standard *FLOW-3D* as defect tracking, air entrainment and others. Passive scalars are parameters that are available to users for customization. The Moisture Transport Module in *FLOW-3D YMUZ2* extensively uses passive scalar variables to solve and track variables related to moisture transport phenomena.

In the scalar namelist, eight scalars are defined for the Moisture Transport Module. The general input keywords associated with all the scalars are described in Table 3-4.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsc</td>
<td>8</td>
<td>Unitless</td>
<td>Number of scalars for use in <em>FLOW3D-YMUZ2</em></td>
<td>The value is fixed, and any change is likely to provide undesirable output.</td>
</tr>
<tr>
<td>rmsc</td>
<td>0</td>
<td>Unitless</td>
<td>Schmidt number for diffusion in turbulent flows</td>
<td></td>
</tr>
</tbody>
</table>

Each scalar needs at least three parameters to completely define it. The purpose for using these scalars, their functionality, and parameters related to each scalar are described in the following section.

3.2.1.3.1 Scalar-1

Scalar-1 accounts for the total water mass fraction in the domain. This scalar accounts for the diffusion and advection of liquid water. It is assumed that any liquid water in the bulk flow exists as a fog and moves as if it was vapor. Also, liquid water in the bulk flow does not coalesce or rains out due to gravity from the flow domain. The parameters associated with Scalar-1 are described in Table 3-5.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(1)</td>
<td>3</td>
<td>Unitless</td>
<td>Second order advection of the scalar in Fluid 1</td>
<td>This parameter describes the advection option for Scalar-1. A value of 1 signifies that liquid water in the air will advect with fluid-1, which is air for the in-drift Moisture Transport Module. For other problems, it could be set to 1 or 5 depending upon the requirement.</td>
</tr>
</tbody>
</table>
Table 3-5. Parameters for Scalar-1 in the Moisture Transport Module (continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmsc(1)</td>
<td>$0.26 \times 10^{-4}$ [0.17 \times 10^{-4}]</td>
<td>kg/m-s [lb/ft-s]</td>
<td>Molecular diffusion coefficient for Scalar-1</td>
<td>The default value is the diffusion coefficient of liquid water in Fluid-1. Fluid-1 is air for studies related to FLOW-3D YMUZ2.</td>
</tr>
<tr>
<td>scltit(1)</td>
<td>Tot.Water</td>
<td>Unitless</td>
<td>Title of Scalar-1</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.3.2 Scalar-2

Scalar-2 accounts for the mass flux at the wall. This scalar is used for extracting a field variable for postprocessing. This variable holds the current water mass flux at each wall surface cell for recording to the output data file so that the user can view/process the information. For example, total evaporation rate or condensation rate at a wall can be computed from data extracted from FLOW-3D YMUZ2 and transferred to a spreadsheet. The scalar is only defined in the boundary surfaces that have an interface with a solid wall. The input keywords associated with Scalar-2 are discussed in Table 3-6.

Table 3-6. Parameters for Scalar-2 in the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(2)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>This scalar is for extracting field variable, liquid water flux that is required for post processing. It is not used in the solver. This parameter value should not be changed.</td>
</tr>
<tr>
<td>cmsc(2)</td>
<td>0</td>
<td>kg/m-s [lb/ft-s]</td>
<td>Molecular diffusion coefficient for Scalar-2</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(2)</td>
<td>&quot;Liq.Flux&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-2</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.3.3 Scalar-3

Scalar-3 accounts for the relative humidity in the domain. This variable holds the current value of relative humidity at all flow domains and boundary cells. This value is intended for
visualization of the humidity during a simulation and is extracted from the flow field. The input keywords associated with Scalar-3 are discussed in Table 3-7.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(3)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>This scalar is for extracting field variable relative humidity required for post processing; it is not used in the solver. This parameter value should not be changed.</td>
</tr>
<tr>
<td>cmsc(3)</td>
<td>0 [0]</td>
<td>kg/m-s [lb/ft-s]</td>
<td>Molecular diffusion coefficient for Scalar-3</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(3)</td>
<td>&quot;Rel.Hum&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-3</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.3.4 Scalar-4

Scalar-4 accounts for the net water mass transferred at the wall, and like Scalar-2 and Scalar-3, it is also used to store variables for visualization. This variable accounts for the total mass of water that has changed phase at a wall. Initial transient values are also stored in this scalar. As a result, the initial values of Scalar-4 will tend to show nonphysical oscillations. To use this variable, it is suggested that a user perform a restart once the start-up transients are purged out and the flow stabilizes. The input keywords associated with Scalar-4 are discussed in Table 3-8.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(4)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>This scalar is for extracting field variable total mass of water changing phase at the wall. It is not used in the solver. This parameter value should not be changed.</td>
</tr>
</tbody>
</table>
Table 3-8. Parameters for Scalar-4 in the Moisture Transport Module (continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmsc(4)</td>
<td>0</td>
<td>kg/m-s [lb/ft-s]</td>
<td>Molecular diffusion coefficient for Scalar-4</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(4)</td>
<td>&quot;Net.Liq&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-4</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.3.5 Scalar-5

Scalar-5 saves the water vapor mass fraction in each cell of the computational domain and does not computationally interact with the solver. The initial value of water vapor mass fraction (i.e., Scalar-5) should be specified at the beginning of the simulation.

Table 3-9. Parameters for Scalar-5 in the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(5)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>This scalar accounts for the water vapor mass fraction in the domain. It is not used in the solver. This parameter value should not be changed.</td>
</tr>
<tr>
<td>cmsc(5)</td>
<td>0 [0]</td>
<td>kg/m-s [lb/ft-s]</td>
<td>Molecular diffusion coefficient for Scalar-5</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(5)</td>
<td>&quot;Vap.Wat&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-5</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.3.6 Scalar-6

Scalar-6 is similar in nature to Scalar-5, but it stores liquid water mass fraction in the computational domain. It is also meant for accounting a field variable and post processing. Scalar-6 needs to be initialized, which will provide the solver with a value of liquid water mass fraction at every grid point at the beginning of the solution.
### Table 3-10. Parameters for Scalar-6 in the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(6)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>This scalar accounts for the liquid water mass fraction in the domain. It is not used in the solver. This parameter value should not be changed.</td>
</tr>
<tr>
<td>cmsc(6)</td>
<td>0</td>
<td>kg/m-s</td>
<td>Molecular diffusion coefficient for Scalar-7</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(6)</td>
<td>&quot;Liq.Wat&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-7</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.3.7 Scalar-7

Scalar-7 keeps track of the wall phase change calculation iterations and can be used for debugging and troubleshooting. This scalar is defined only for wall surface cells. This variable counts the number of iterations required to have a converged solution of vapor concentration and wall temperature at each wall surface cell. This scalar is also useful for troubleshooting and can be used to set the relaxation factor \( \text{vaprlx} \).

### Table 3-11. Parameters for Scalar-7 in the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(7)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>It keeps track of the number of iterations required to have a converged solution of wall temperature and vapor concentration at the surface. This parameter value should not be changed.</td>
</tr>
<tr>
<td>cmsc(7)</td>
<td>0</td>
<td>kg/m-s</td>
<td>Molecular diffusion coefficient for Scalar-7</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(7)</td>
<td>&quot;Itr.Wall&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-7</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>
3.2.1.3.8 Scalar-8

Scalar-8 keeps track of the number of iterations required to adjust the bulk flow relative humidity. It is similar in nature to Scalar-7 as it is also used for debugging and troubleshooting. The main function of Scalar-8 is to hold the current number of iterations required to have a converged solution of vapor concentration and fluid temperature at each flow domain cell to satisfy the energy–humidity constraint. Like Scalar-7, this variable is also useful for troubleshooting and can be used to set the relaxation factor $vaprlx$.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isclr(8)</td>
<td>0</td>
<td>Unitless</td>
<td>No advection of the scalar</td>
<td>It keeps track of the number of iterations required to have a converged solution of fluid temperature and vapor concentration in the flow domain. This parameter value should not be changed.</td>
</tr>
<tr>
<td>cmsc(8)</td>
<td>0</td>
<td>kg/m-s/ [lb/ft-s]</td>
<td>Molecular diffusion coefficient for Scalar-8</td>
<td>The molecular diffusion value is set to zero as this scalar is only for accounting a field variable. This parameter value should not be changed.</td>
</tr>
<tr>
<td>scltit(8)</td>
<td>&quot;Itr.Mesh&quot;</td>
<td>Unitless</td>
<td>Title of Scalar-8</td>
<td>The string input should be between the quotation marks.</td>
</tr>
</tbody>
</table>

3.2.1.4 Flow Field Initialization (FL)

In the standard version of FLOW-3D, initialization of pressure is not mandatory. It is, however, a requirement for FLOW-3D YMUZ2 as a number of flow parameters are calculated based on the initial pressure values. In addition, a number of user-defined scalars have to be initialized. It is possible to specify spatially varying initial values for these parameters as described in the standard FLOW-3D Users Manual (Flow Sciences, Inc., 2005). Table 3-13 describes the input keywords related to the namelist block FL.
### Table 3-13. Initialization Parameters for the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>presi</td>
<td>99379 [2075.57]</td>
<td>Pa [lbf/ft²]</td>
<td>Initial uniform pressure in the flow domain</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This parameter must be specified for FLOW-3D YMUZ2. The regular FLOW-3D</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>solver can simulate flow without any initial uniform pressure specification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>but initial uniform pressure must be specified in FLOW-3D YMUZ2 to</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>calculate mixture parameters. The simulation will provide an erroneous</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>result without an initial pressure specification.</td>
<td></td>
</tr>
<tr>
<td>sclri(1)</td>
<td>0.022</td>
<td>Unitless</td>
<td>Initial value of total water mass fraction in the flow domain (Scalar-1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default value is for 100% relative humidity at a pressure of 99.379 kPa</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[2075.57 lbf/ft²] and a temperature of 299.7 K [79.79 °F].</td>
<td></td>
</tr>
<tr>
<td>sclri(5)</td>
<td>0.022</td>
<td>Unitless</td>
<td>Initial value of water vapor mass fraction in the flow domain (Scalar-5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default value is for 100% relative humidity at a pressure of 99.379 kPa</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[2075.57 lbf/ft²] and a temperature of 299.7 K [79.79 °F].</td>
<td></td>
</tr>
<tr>
<td>sclri(6)</td>
<td>0</td>
<td>Unitless</td>
<td>Initial value of liquid water mass fraction in the flow domain (Scalar-6)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default value is for 100% relative humidity at a pressure of 99.379 kPa</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[2075.57 lbf/ft²] and a temperature of 299.7 K [79.79 °F].</td>
<td></td>
</tr>
</tbody>
</table>

### 3.2.1.5 Temperature Field Initialization (TEMP)

The input parameters for the namelist block TEMP are described in Table 3-14.

### Table 3-14. Temperature Field Initialization for the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntmp</td>
<td>1</td>
<td>Unitless</td>
<td>Number of zones created to specify initial temperature. Each zone could</td>
<td>This parameter depends upon the number of zones created to specify initial temperatures. The typical value would be 1 for most of the in-drift heat transfer simulations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>start with a distinct temperature.</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-14. Temperature Field Initialization for the Moisture Transport Module (continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>tempi</td>
<td>299.7 [79.79]</td>
<td>K [°F]</td>
<td>Initial uniform temperature in the domain</td>
<td>If multiple initial temperature zones are created, each of them needs to be initialized with the keyword TREG.</td>
</tr>
</tbody>
</table>

3.2.1.6 User-Defined Parameters (USRDAT)

User-defined parameters used in *FLOW-3D YMUZ2* are described in Table 3-15.

Table 3-15. User-Defined Parameters for the Moisture Transport Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>istwtf_stg</td>
<td>&quot;tw&quot;</td>
<td>Unitless</td>
<td>&quot;tw&quot; for wall source &quot;tf&quot; for fluid source</td>
<td></td>
</tr>
<tr>
<td>imoist_stg(n)</td>
<td>-n</td>
<td>Unitless</td>
<td>This parameter defines obstacles that can condense and evaporate without limit. These are essentially saturated porous surfaces. This is an array input and should be defined for every condensing or evaporating surfaces.</td>
<td>This parameter needs to be specified for any surface that participates in the condensation-evaporation process.</td>
</tr>
<tr>
<td>ilqonly_stg(n)</td>
<td>-n</td>
<td>Unitless</td>
<td>This parameter defines obstacles that can condense without limit. These surfaces can also reevaporate the condensed water as long as it is available.</td>
<td>This parameter needs to be specified for any surface that participates in the condensation-evaporation process.</td>
</tr>
<tr>
<td>isvap_stg</td>
<td>1</td>
<td>Unitless</td>
<td>Defines scalar index for water concentration.</td>
<td>The default value should not be changed.</td>
</tr>
<tr>
<td>isliq_stg</td>
<td>2</td>
<td>Unitless</td>
<td>Defines scalar index for liquid flux.</td>
<td>The default value should not be changed.</td>
</tr>
<tr>
<td>isrh_stg</td>
<td>3</td>
<td>Unitless</td>
<td>Defines scalar index for relative humidity.</td>
<td>The default value should not be changed.</td>
</tr>
<tr>
<td>istlq_stg</td>
<td>4</td>
<td>Unitless</td>
<td>Defines scalar index for total liquid accumulation.</td>
<td>The default value should not be changed.</td>
</tr>
</tbody>
</table>
### Table 3-15. User Defined Parameters for the Moisture Transport Module (continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isywv_stg</td>
<td>5</td>
<td>Unitless</td>
<td>Defines scalar index for liquid water (mist).</td>
<td>The default value should not be changed.</td>
</tr>
<tr>
<td>cvvap_stg</td>
<td>1370 [0.3272]</td>
<td>J/kg-K [BTU/lb-°F]</td>
<td>Constant volume specific heat for water vapor.</td>
<td>Parameter value depends on the specific range of temperature in the simulation.</td>
</tr>
<tr>
<td>rvap_stg</td>
<td>416 [0.01]</td>
<td>J/kg-K [BTU/lb-°F]</td>
<td>Gas constant for water vapor.</td>
<td>Parameter value depends on the specific range of temperature in the simulation.</td>
</tr>
<tr>
<td>vaprlx_stg</td>
<td>0.8</td>
<td>Unitless</td>
<td>Relaxation factor for phase change iterations.</td>
<td>Parameter value depends on stability and other numerical considerations.</td>
</tr>
<tr>
<td>rhlim_stg</td>
<td>&quot;y&quot;</td>
<td>Unitless</td>
<td>y (Limit relative humidity to 100%)</td>
<td>&quot;y&quot; option sets a limit of relative humidity to 100%, and any excess water vapor will condense and advect as fog beyond saturation limit.</td>
</tr>
</tbody>
</table>

### 3.2.2 Input Requirements for Radiation Module in FLOW-3D YMUZ2

The Radiation Module of FLOW-3D YMUZ2 requires a number of input keywords that are described in this section. Some of the keywords are common to the Moisture Transport Module. The Radiation module could, however, be used independent of the Moisture Transport Module.

#### 3.2.2.1 Computational Parameters (XPUT)

The radiation module input parameters for the namelist block XPUT are provided in Table 3-16.

### Table 3-16. Computational Parameters for the Radiation Module

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>isurd</td>
<td>1</td>
<td>Unitless</td>
<td>User should always set the value equal to 1 when using FLOW-3D YMUZ2. Any other value will result in an error.</td>
<td>Indicates that the USRDAT namelist block must be created and read by the solver.</td>
</tr>
</tbody>
</table>
3.2.2.2 User-Defined Parameters (USRDAT)

Almost all the user-defined parameters in the Radiation Module deal with the geometry and orientation of the solid objects and surfaces participating in the radiation heat transfer process. A generalized schematic is shown in Figure 3-1 to explain the input keywords related to the user-defined parameters in the Radiation Module. The schematic shows a number of solid objects placed side by side. These objects can have any arbitrary orientation in space. Figure 3-1 has line orientation relevant to Yucca Mountain disposal plans currently considered by DOE. Each object is identified with an index "m" and contains a number of radiating surfaces identified with an index "n". It is not necessary for every surface of an object to participate in radiation. Only those surfaces that take part in radiation are identified with the index "n". For example, only surface n = 4 participates in the radiation heat transfer on the object m = 2. None of the surfaces in object m = 4 take part in the radiation heat transfer process. It is also possible to subdivide a continuous surface with some subdivisions participating in radiation while others stay inert. Table 3-17 provides the input parameters for the namelist block USRDAT.

3.3 Output from FLOW-3D YMUZ2

The FLOW-3D GUI is used to graphically display the computational domain, geometrical parameters, and flow field variables. Values such as temperature, pressure, water vapor concentration, and relative humidity can be presented in contour plots. The use of the standard flow visualization GUI tool has certain limitations with respect to the Moisture Transport and Radiation Modules. The standard visualization tool cannot compute the heat transfer due to phase change or radiation. The user needs to perform some additional calculations to extract and visualize these parameters. In the case of the Radiation Module, a special file is created to record the radiation-specific input parameters and to record the values of the radiation heat transfer rates from each radiation surface. These heat transfer rates can be used to compute quantities such as total heat transfer rate.

Figure 3-1. Schematic for an Arrangement of Solid Bodies With Radiating Surfaces
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>srsrf_stg</td>
<td>1</td>
<td>Unitless</td>
<td>Number of radiating surfaces</td>
<td>This parameter is only used when the code calculates the configuration factors.</td>
</tr>
<tr>
<td>eps_stg(n)</td>
<td>0.9</td>
<td>Unitless</td>
<td>Emissivity of surface number &quot;n&quot;</td>
<td>If the emissivity of surface n = 1 is 0.9 and that of surface n = 2 is 0.9, then eps_stg (1) = 0.9 and eps_stg (2) = 0.95.</td>
</tr>
<tr>
<td>rad_stg(1,n)</td>
<td></td>
<td>Unitless</td>
<td>The obstacle number that has surface &quot;n&quot;</td>
<td>As shown in Figure 3-1, object m = 3 contains surface n = 5; hence .rad_stg (1,5) = 3.</td>
</tr>
<tr>
<td>rad_stg(2,n)</td>
<td>m</td>
<td>Unitless</td>
<td>Lower x-limit of cells on radiation surface &quot;n&quot;</td>
<td>For surface 5 rad_stg, (2,5) will be the minimum x coordinate value of the surface.</td>
</tr>
<tr>
<td>rad_stg(3,n)</td>
<td>m</td>
<td>Unitless</td>
<td>Upper x-limit of cells on radiation surface &quot;n&quot;</td>
<td>For surface 5 rad_stg, (2,5) will be the maximum x coordinate value of the surface.</td>
</tr>
<tr>
<td>rad_stg(4,n)</td>
<td>m</td>
<td>Unitless</td>
<td>Lower y-limit of cells on radiation surface &quot;n&quot;</td>
<td>For surface 5 rad_stg, (2,5) will be the minimum y coordinate value of the surface.</td>
</tr>
<tr>
<td>rad_stg(5,n)</td>
<td>m</td>
<td>Unitless</td>
<td>Upper y-limit of cells on radiation surface &quot;n&quot;</td>
<td>For surface 5 rad_stg, (2,5) will be the maximum y coordinate value of the surface.</td>
</tr>
<tr>
<td>rad_stg(6,n)</td>
<td>m</td>
<td>Unitless</td>
<td>Lower z-limit of cells on radiation surface &quot;n&quot;</td>
<td>For surface 5 rad_stg, (2,5) will be the minimum z coordinate value of the surface.</td>
</tr>
<tr>
<td>rad_stg(7,n)</td>
<td>m</td>
<td>Unitless</td>
<td>Upper z-limit of cells on radiation surface &quot;n&quot;</td>
<td>For surface 5 rad_stg (2,5) will be the maximum z coordinate value of the surface.</td>
</tr>
<tr>
<td>iusrcf_stg</td>
<td>0</td>
<td>Unitless</td>
<td>= 0 for configuration factors are computed by the code = 1 for configuration factors are specified by the user</td>
<td>Provides user a choice of either reading the configuration factors that were calculated externally and provided as input or calculate it within FLOW-3D YMUZ2.</td>
</tr>
</tbody>
</table>
Table 3-17. User Defined Parameters for the Radiation Module (continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Units</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf_stg(i,j)</td>
<td></td>
<td>Unitless</td>
<td>User-specified value of radiation configuration factor, $F_{nm}$</td>
<td>This option is only functional if iusrcf_stg = 0. These configuration factors could be calculated externally and supplied as an input. If the view configuration factor between surfaces 3 and 4 is 0.2 then cf_stg (3,4) = 0.2.</td>
</tr>
</tbody>
</table>

3.3.1 Output from the Moisture Transport Module

All of the scalar variables defined in the "$Scalar" namelist block can be viewed as contour plots in the FLOW-3D graphics package. These values can also be extracted into text files for postprocessing. To compute the energy transfer via mass transfer, the user must extract the following values for the surfaces in question:

- Fluid temperature
- Wall temperature
- Heat transfer coefficient
- Liquid flux (variable specific to Moisture Transport Module)

Conductive and convective heat flux at each cell are computed by the FLOW-3D YMUZ2 postprocessor. The energy transfer via phase change has to be computed externally such as by Microsoft® Excel® (Microsoft Corporation, 2002b) using the values listed previously obtained from the FLOW-3D YMUZ2 solutions from the total energy values at each cell.

3.3.2 Output from the Radiation Module

Subroutines rad_init and rad_calc were modified to record the pertinent output information from the Radiation Module. These modifications are given in the file check_rad.* where * stands for the problem being executed as is the case with the other problem-specific FLOW-3D files. The configuration factor matrix terms and the time history of the radiation heat transfer rates of the surfaces that are activated for radiation information are recorded in this file.
4 SAMPLE PROBLEM

This section provides an example for the use of the Moisture Transport and Radiation Modules. This sample problem is identical to one of the test cases described in Green (2006). It is intended for demonstration purposes, not for testing all code capabilities.

4.1 Problem Description

The objective of this test is to estimate the heat transfer rate due to conduction, convection, and radiation along with the mass transport rate due to phase change in a heated enclosure. The vertical walls of the enclosure participate in the radiative heat transfer and provide the source of water. The horizontal surfaces are treated as adiabatic hydrophobic walls.

A two-dimensional enclosure measuring 0.1 × 0.1 m [0.32 × 0.32 ft] is depicted in Figure 4-1. The left vertical wall is 0.025 m [8.2 × 10⁻² ft] thick and has an internal heat generation rate such that the heat flux at the inner surface is 200 W/m² [63.4 BTU/hr-ft²]. The outer surface of this wall is adiabatic. The right vertical wall is 0.025 m [8.2 × 10⁻² ft] thick, and its outer surface is held constant at 300 K [80.3 °F]. The emissivity of both the left and right walls is 0.9. The vertical walls provide for the evaporation and condensation of water as needed under the local temperature and concentration conditions in the flow. The upper and lower walls are adiabatic and do not exchange heat with the vertical walls. These walls are assumed to be transparent to radiation and, therefore, do not interact with the other walls via this mode. These walls are furthermore assumed to not be a source or a sink for water. The only interaction of these walls in the problem is to bound the flow and provide for viscous drag. The acceleration due to gravity is assumed to be only 0.001 × g (g is the gravitational constant) so that the flow field for these geometric and thermal conditions will be laminar.

The objective here is to compare the effects of the radiation and moisture transport processes to accurately model a turbulent flow scenario. The fluid and wall properties used in this simulation are listed in Table 4-1.

The density value listed above is used as the nominal density in the conservation of energy equation. The incompressible ideal gas model is used for this test case for the temperature- and concentration-dependent density that is used for the momentum equation. The Moisture Transport Module parameters used in this study are described in Table 4-2.

4.2 Test Cases

The following four cases will be investigated:

1. Convection only
2. Convection with radiation
3. Convection with moisture transport
Figure 4-1. Schematic for Convection, Radiation, and Mass Transfer in a Two-Dimensional Enclosure

\[ 1 \text{ m} = 3.28 \text{ ft}; \, 1 \text{ W/m}^3 = 9.66 \times 10^{-2} \text{ BTU/hr/ft}^3; \, ^\circ\text{F} = 1.8 \times T \text{ K} - 459.67 \]

Table 4-1. Fluid and Wall Properties for the Combined Heat and Moisture Transport Problem in an Enclosure

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viscosity ((\mu))</td>
<td>(2 \times 10^{-5} \text{ Pa-sec} ) [(4.16 \times 10^{-7} \text{ lb-s/ft}^2)]</td>
</tr>
<tr>
<td>Fluid thermal conductivity ((k_{\text{air}}))</td>
<td>(0.026 \text{ W/m-K} ) [(0.015 \text{ BTU/hr-ft-}^\circ\text{F})]</td>
</tr>
<tr>
<td>Air/Vapor diffusivity (D)</td>
<td>(2.6 \times 10^{-5} \text{ m}^2/\text{sec} ) [(26.5 \times 10^{-5} \text{ ft}^2/\text{sec})]</td>
</tr>
<tr>
<td>Nominal Density ((\rho))</td>
<td>(1.169 \text{ kg/m}^3 ) [(0.073 \text{ lb/ft}^3)]</td>
</tr>
<tr>
<td>Wall thermal conductivity ((k_w))</td>
<td>(1 \text{ W/m-K} ) [(0.58 \text{ BTU/hr-ft-}^\circ\text{F})]</td>
</tr>
</tbody>
</table>
### Table 4-2. Moisture Transport Module Parameters Used in the Combined Heat and Moisture Transport Problem in an Enclosure

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heat of vaporization of water ($u_{fg}$)</td>
<td>2,304,900 J/kg [990.94 BTU/lb]</td>
</tr>
<tr>
<td>Water vapor-specific heat ($C_{vv}$)</td>
<td>1370 J/kg-K [0.33 BTU/lb-°F]</td>
</tr>
<tr>
<td>Liquid water-specific heat ($C_{vl}$)</td>
<td>$2.6 \times 10^{-5}$ m$^2$/sec [26.5 $\times 10^{-5}$ ft$^2$/sec]</td>
</tr>
<tr>
<td>Water vapor gas constant ($R_v$)</td>
<td>416 J/kg-K [$1 \times 10^{-2}$ BTU/lb-°F]</td>
</tr>
<tr>
<td>Air gas constant ($R_a$)</td>
<td>289 J/kg-K [$6.9 \times 10^{-2}$ BTU/lb-°F]</td>
</tr>
</tbody>
</table>

The input file for Case 4, which simulates the enclosure for convection, radiation, and moisture transport, is provided here. Input parameters that are specific to the Moisture Transport and Radiation Modules are highlighted and enclosed in boxes. The input file for the other three test cases is provided in Appendix C. Users can use these input files to test the installation of 

**FLOW-3D YMUZ2.**

### 4.3 Input File for Convection With Moisture Transport and Radiation

Heat transfer in 2-D Sq. Box w/ natural convection, radiation, moisture  
Ra ~ 6000 for natural convection only  
Box is 0.1 m x 0.1 m long,  
Adiabatic at top and bottom,  
20 W generated in left wall,  
300 K at outer face of right wall.  
Both walls have $k=1$ w/(m*K), $rcobs=1000$ joule/(m$^3$K)  
Moisture at left and right walls if modeled  
Radiation at left and right walls if modeled

```plaintext
$xput
remark='units are SI',
$set
remark='Sharp interface not required',
ifvis=0, remark='Laminar',
ifenrg=2, remark='Solve energy eq., first order',
ifrho=1, remark='Temp. dependent density',
gz=-9.8e-03, remark='Gravity',
ipdis=1, remark='Hydrostatic pressure distribution in z-direction',
ihtc=2, remark='Evaluate heat transfer with solid conduction',
iwsh=1, remark='Wall shear active',
delt=1.e-4, remark='',
twfin=4000., remark=''
rmrhoe=1., remark='Density diffusion term coefficient',
rmrho=1., remark='Energy diffusion term coefficient',
remark=' RMRHO, RMRHOE REQUIRED FOR VAPOR TRANSPORT MODEL',
iusrd=1,
$end
$limits
$end
$props
```
units="si",

rhof=1.1687, remark='Bulk nominal density at 310 K for energy equation',
muf=2.e-05, remark='Bulk nominal dynamic viscosity density for momentum equation',
cv1=717., remark='Const. Vol. Specific heat of dry air',

thc1=0.026, remark='Bulk nominal thermal conductivity for energy equation',
thexf1=0.003226, remark='Approximate thermal expansion coefficient',
tstar = 300., remark='Reference temperature for thermal expansion coefficient',
remark='THEXF1, TSTAR not needed for vapor tranport model',
remark='but are used in some parts of code to set up simulation',

nsc=8,
isclr(1)=3, cmsc(1)=0.26e-04, scltit(1)='Tot.Water', rmsc=0.,
isclr(2)=0, cmsc(2)=0., scltit(2)='Liq.Flux',
isclr(3)=0, cmsc(3)=0., scltit(3)='Rel.Hum',
isclr(4)=0, cmsc(4)=0., scltit(4)='Net.Liq',
isclr(5)=0, cmsc(5)=0., scltit(5)='Vap.Wat',
isclr(6)=0, cmsc(6)=0., scltit(6)='Liq.Wat',
isclr(7)=0, cmsc(7)=0., scltit(7)='Itr.Wall',
isclr(8)=0, cmsc(8)=0., scltit(8)='Itr.Mesh',

remark='Water vapor scalars not used for this problem but need to be defined to use the ',
remark=' radiation and water transport modules together',
remark='Scalar 1 is the diffusing/adlecting water',
remark='Scalar 2 (non-diffusing) is for storing the surface phase change flux values',
remark='Scalar 3 (non-diffusing) is for storing the relative humidity values',
remark='Scalar 4 (non-diffusing) is for storing the net surface liquid accumulation',
remark='Scalar 5 (non-diffusing) is for storing the vapor water concentration',
remark='Scalar 6 (non-diffusing) is for storing the liquid water (mist) concentration',
remark='Scalar 7 (non-diffusing) is for storing the calculation iterations for the',
remark=' wall mass flux',
remark='Scalar 8 (non-diffusing) is for storing the calculation iterations for the',
remark=' mist phase change in the fluid interior',
remark='Scalars 7 and 8 are helpful in tuning the value of vaprflx if necessary',

wfl=1,           remark='Symmetry at left to make all the heat go into the inner face',
wfr=2,           remark='Constant temperature wall at right',
tbc(2)=300.,
wfl=1, wbk=1,    remark=' Symmetry at front and back',
wbf=2,           remark=' Insulated wall',
wt=2,           remark=' Insulated wall',

px(1) =-0.025, py(1) =0.0, pz(1) = 0.0,
px(2) = 0.0, py(2) =1.0, pz(2) = 0.1,
px(3) = 0.1, px(4) = 0.125,
xncell(1)=5,
xncell(2)=20,
nxcell(3)=5, nsclri= 30, nycelt=1, nzcelt=30,

$end

$obs
  avrck=-3.1, nobss = 2, tobs(1)=0., tobs(2)=1000.,
  remark='Obstacle 1. Left channel wall. Total heat output 50W',
  xl(1)=-0.025, xh(1)=0.,
  kobs(1)= 1., rcobs(1)=10000.,
  twobs(1,1)=300.,
  pobs(1,1)=20., pobs(2,1)=20.,
  remark='Obstacle 2. Right channel wall. Conduction to constant temp',
  xl(2)=0.1,  xh(2)=0.125,
  kobs(2)= 1., rcobs(2)=10000.,
  twobs(1,2)=300.,
$end

$fl

remark=' No vapor in fluid',
  sclri(1)=0.0,
  sclri(5)=0.0,
  sclri(6)=0.0,
  presi=101325.,

$end

$bf
$end

$temp

ntmp=1, tempi=300.,
  presi=101325.,

$end

$motn
$end
4.4 Results

All of the cases were conducted using a 20 × 30-cell mesh in the flow field. Five cells were used to represent the wall at the left and right boundaries so the entire grid was 30 × 30 cells.

A sample of the FLOW-3D YMUZ2 results for the air flow vectors and temperature contours is shown in Figure 4-2 for the input file described previously. The velocity vectors show that the flow in the enclosure is laminar. The temperature contours are as expected with maximum values in the solid and fluid in the upper right corner of their respective regions. The
temperature minima are in the lower left corners of the fluid region and solid region, respectively.

The *FLOW-3D YMUZ2* predictions for the four scenarios are summarized in Table 4-3. Results show that *FLOW-3D YMUZ2* is capable of simulating conduction, convection, and radiation along with moisture transport. Details of this simulation and comparison with analytical results are documented in Green, S. (2006).

![Figure 4-2. FLOW-3D YMUZ2 Predictions for Temperature (K) (Color Contours) and Air Flow (Vectors) for Combined Convection Radiation, and Moisture Transport](image)

\[1 \text{ m} = 3.28 \text{ ft} ; \ T^o \text{F} = 1.8 \times T \text{ K} - 459.67\]
<table>
<thead>
<tr>
<th>Scenario</th>
<th>$T_h$, Hot Surface Temperature Average (K)*</th>
<th>$T_c$, Cold Surface Temperature Average (K)*</th>
<th>Heat Transfer Rate</th>
<th>Mass Transfer (kg/m-s)‡</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Convection (W/m)†</td>
<td>Radiation (W/m)†</td>
</tr>
<tr>
<td>Convection Only</td>
<td>646.2</td>
<td>304.5</td>
<td>20</td>
<td>n/a</td>
</tr>
<tr>
<td>Convection + Radiation</td>
<td>362.7</td>
<td>304.5</td>
<td>2.3</td>
<td>17.7</td>
</tr>
<tr>
<td>Convection + Moisture Transport</td>
<td>340.9</td>
<td>304.5</td>
<td>1.8</td>
<td>n/a</td>
</tr>
<tr>
<td>Convection + Radiation + Moisture Transport</td>
<td>333.8</td>
<td>304.5</td>
<td>1.3</td>
<td>7.9</td>
</tr>
</tbody>
</table>

*° F = 1.8 × T K−459.67
†W/m = 1.04 BTU/hr-ft
‡kg/m-s = 0.67 lb/ft-s
5 REFERENCES


THEORETICAL BASIS OF THE MOISTURE TRANSPORT AND RADIATION MODULES

Appendix A presents the theoretical basis for the Moisture Transport and Radiation Module. This discussion includes the evaluation techniques for different flow variables in the FLOW-3D YMUZ2 solver, proof of concept calculations, and solution schemes.

A1 MOISTURE TRANSPORT MODULE

The theory used to represent processes associated with the moisture transport in FLOW-3D YMUZ2 is described in this section. The equations for estimating the fluid density are discussed in Section A1.1 followed by the methodology to estimate the total energy in Section A1.2. Section A1.3 describes the method to calculate temperature. The governing equations for the mass transfer and heat transfer rates associated with the Moisture Transport Modules are presented in Section A1.4.

A1.1 Density Evaluation

The bulk density of the air/vapor mixture is obtained from engineering psychometrics (e.g., ASHRAE, 1977) with the modification that allows for some water to be condensed as a mist in the mixture. The species diffusion equation solved in FLOW-3D YMUZ2 is in terms of the mass fraction of the air and water. Hence, the density property in FLOW-3D YMUZ2 must be defined in terms of the mass fractions of these constituents.

The bulk density is defined as

$$\rho = \frac{m_{\text{tot}}}{V} = \frac{m_a + m_{\text{av}} + m_{\text{wl}}}{V}$$  \hspace{1cm} (A–1)

where

- $m_{\text{tot}}$ = total mass of all species in sample volume [kg]
- $m_a$ = mass of air [kg]
- $m_{\text{av}}$ = mass of water vapor [kg]
- $m_{\text{wl}}$ = mass of water mist [kg]
- $V$ = sample volume [m$^3$]
The mass of air is computed from the ideal gas equation of state along with the approximation that the gas-only volume nearly equals the total volume

\[ m_a = \frac{P_a V_g}{R_a T} \approx \frac{P_a V}{R_a T} x_{ag} \quad (A-2) \]

where

- \( P_a \) = partial pressure of air [Pa]
- \( P \) = total static pressure [Pa]
- \( R_a \) = ideal gas constant for air = 287 J/kg-K [6.85 \times 10^{-2} BTU/lb-°F]
- \( x_{ag} \) = mole fraction of air in the gas/vapor portion of the volume
- \( V_g \) = volume of gas only in control volume [m³]
- \( T \) = temperature [K]

Likewise, the mass of water vapor is

\[ m_{wv} = \frac{P_{wv} V_g}{R_v T} \approx \frac{P_{wv} V}{R_v T} x_{wvg} \quad (A-3) \]

where

- \( P_{wv} \) = partial pressure of water vapor [Pa]
- \( R_v \) = ideal gas constant for water vapor {462 J/(kg-K) [0.11 BTU/lb/°F]}
- \( x_{wvg} \) = mole fraction of air in the gas/vapor portion of the volume

The mass fractions of the vapor water and liquid water are computed as part of the humidity calculations and are known inputs to the density evaluation algorithm. The mass fractions of water vapor and air in the gas phase can now be defined using Eqs. (A–4a) and (A–4b), respectively

\[ c_{wvg} = \frac{c_{wv}}{c_w + c_a} = \frac{c_{wv}}{c_{wv} + (1 - c_w)} \quad (A-4a) \]

and

\[ c_{ag} = 1 - c_{wvg} \quad (A-4b) \]

where

- \( c_a \) = mass fraction of air
- \( c_w \) = mass fraction of water
- \( c_{wv} \) = mass fraction of water vapor
- \( c_{ag} \) = mass fraction of air vapor in gas
- \( c_{wvg} \) = mass fraction of water vapor in gas
The respective mole fractions for water vapor and air in only the gas space can now be defined using Eqs. (A–5a) and (A–5b), respectively

\[ x_{wvg} = \frac{c_{wvg} \frac{M_a}{M_w}}{1 - c_{wvg} \left(1 - \frac{M_a}{M_w}\right)} \quad (A–5a) \]

and

\[ x_{ag} = 1 - x_{wvg} \quad (A–5b) \]

where

\[ M_a = \text{molecular weight of air} \]
\[ M_w = \text{molecular weight of water} \]

Substituting Eq. (A–2) – Eq. (A–5b) in Eq. (A–1) gives

\[ \rho = \frac{1}{V} \left[ \frac{PV}{R_a T} x_{ag} + \frac{PV}{R_w T} x_{wvg} + c_{wl} m_{tot} \right] = \frac{P}{T} \left( \frac{x_{ag}}{R_a} + \frac{x_{wvg}}{R_w} \right) + c_{wl} \rho \quad (A–6) \]

which can be reduced to

\[ \rho = \frac{P}{T(1 - c_{wl})} \left( \frac{x_{ag}}{R_a} + \frac{x_{wvg}}{R_w} \right) \quad (A–7) \]

Eqs. (A–7) and (A–6) serve as the basis for evaluating the bulk density in the Moisture Transport Module. The inputs to this function are the local pressure, temperature, water vapor mass fraction, and liquid water (as mist) mass fraction.

**A1.2 Air/Water Vapor Energy**

The energy of an air/water mixture is evaluated following the basic procedures of standard FLOW-3D Version 9.0 (Flow Sciences, Inc., 2005). In this case, the mass-specific energy (e.g., J/kg) is a function of the local temperature and constituent concentrations

\[ e = c_a C_{va} T + c_{wg} C_{vv} T - c_{wl} h_{vap} \quad (A–8) \]

where

\[ c_{wl} = \text{mass fraction of liquid water (as mist)} \]
\[ T = \text{temperature (K)} \]
\[ C_{va} = \text{specific heat of air at constant volume (J/kg-K)} \]
\[ C_{vv} = \text{specific heat of water vapor at constant volume (J/kg-K)} \]
\[ h_{vap} = \text{energy of vaporization of water (J/kg)} \]

### A1.3 Temperature Evaluation Method

The conservation of energy equation used in FLOW-3D is based on the volume-specific energy defined as the product of the nominal fluid density and the mass-specific energy

\[
\left( \rho \right) = \rho_{nom} e
\]
(A–9)

where

\( \left( \rho \right) \) — volume-specific energy
\( \rho_{nom} \) — constant nominal density provided by the user in the problem setup
\( e \) — mass-specific energy for the air/water mixture

The \( \left( \rho \right) \) product is the conserved quantity, and the temperature is calculated from this parameter for use in various property functions and output files. Consequently, the temperature evaluation function in FLOW-3D must be modified when using the Moisture Transport Model in which the mass-specific energy is different from the standard FLOW-3D installation. The mass-specific energy equation, Eq. (A–8), can be rearranged as

\[
T = \frac{\frac{\left( \rho \right)}{\rho_{nom} C_{va}} + \frac{c_{v} h_{vap}}{c_{a} C_{va} + c_{wv} C_{vv}}}{c_{a} C_{va} + c_{wv} C_{vv}}
\]
(A–10)

### A1.4 Air/Water Vapor Energy

The Moisture Transport Module will be based on the well-established approach of the conservation of mass of multiple chemical species in nonreacting mixtures (Bird, et al., 1960). A conservation equation for the transport of water vapor can be written as

\[
\rho \left( \frac{\partial c_{w}}{\partial t} + \frac{\partial \mathbf{u}}{\partial x} \right) + \frac{\partial}{\partial y} \left( \rho D_{va} c_{w} \right) = \nabla \cdot \left( \rho D_{va} c_{w} \right) + \dot{m}_{source}
\]
(A–11)

where

\( \rho \) — bulk density of the air/water mixture (kg/m³)
\( u, v, w \) — vector components of bulk fluid flow (m/s)
\( x, y, z \) — coordinate directions (m)
\( D_{va} \) — diffusion coefficient for air/vapor (m²/s)
\( \dot{m}_{source} \) — volume specific rate of vapor generation (kg/m³-s)
The FLOW-3D software solves this equation in conjunction with the equations for the conservation of mass and momentum of the air/vapor mixture. The Moisture Transport Module does not directly specify the source term in Eq. (A–11) for the mass transfer rate at walls. Rather, the module essentially provides the boundary conditions for Eq. (A–11) at wall surfaces. The methodology is described next.

A1.4.1 Phase Change at Walls

The scenario for mass transfer at a wall is shown in Figure A–1.

The conservation of energy is applied to the solid wall and the liquid film on the wall over a single timestep in the calculation. Evaporation or condensation will take place as necessary to satisfy the energy change of the solid material. Any heat transfer to or from the air/vapor mixture takes place via conduction/convection at the wall; the air/vapor mixture next to the wall does not exchange energy directly with the water that is undergoing the phase change.
Based on these assumptions, the conservation of energy can be expressed as

\[
m_s \frac{e_{s,2} - e_{s,1}}{\Delta t} = \dot{m}_w h_{vap}
\]

\[
\rho_s C_s V_s \frac{T_{s,2} - T_{s,1}}{\Delta t} = \dot{m}_w h_{vap}
\]

where

- \( m_s \) — mass of solid wall inside the control volume (kg)
- \( \dot{m}_w \) — mass flow rate of water evaporating/condensing (kg/s)
- \( e_{s,1} \) — mass-specific energy of solid at the beginning of the timestep (J/kg)
- \( e_{s,2} \) — mass-specific energy of solid at the end of the timestep (J/kg)
- \( \rho_s \) — solid material density (kg/m³)
- \( C_s \) — solid material specific heat (J/(kg-k))
- \( V_s \) — solid material volume (m³)
- \( h_{vap} \) — heat of vaporization (J/kg)
- \( \Delta t \) — timestep (s)
- \( T_{s,1} \) — wall temperature at beginning of timestep
- \( T_{s,2} \) — wall temperature at end of timestep

There are two unknowns in Eq. (A–12); namely, \( T_{s,2} \) and \( \dot{m}_w \). The mass flow rate of water could be obtained in two different ways:

1. It can be estimated from the diffusion flux of water vapor from the wall into the control volume adjacent to the wall. The diffusion flow of vapor is estimated from an approximation to the Fickian diffusion law (Bird, et al., 1960)

\[
\dot{m}_{w,d} = \dot{\rho} D_{va} A_w \frac{dc}{dx} \approx \dot{\rho} D_{va} (c_{w,sat} - c_w) \frac{A_w^2}{V_f}
\]

where

- \( \dot{m}_{w,d} \) — mass flow via diffusion (kg/s)
- \( c_{w,sat} \) — mass fraction of water vapor at saturation
- \( c_w \) — current mass fraction of water vapor
- \( A_w \) — wall surface area inside the cell (m²)
- \( V_f \) — volume of fluid in the cell (m³)
- \( D_{va} \) — diffusion coefficient for air/vapor (m²/s)
- \( \rho \) — bulk density of mixture (kg/m³)

2. \( \dot{m}_w \) could also be quantified as the mass flow of water that will create a saturated air/vapor mixture in the control volume at the corresponding fluid temperature. The mass transfer rate is estimated from the change in water content required to bring the vapor concentration to the saturation value. Based on the law of conservation of mass, the flow
rate required to saturate the air/vapor mixture adjacent to the wall in a single computational timestep is estimated as

\[ \dot{m}_{w,s} = \left( \rho_{\text{sat}} c_{w,\text{sat}} - \rho c_w \right) \frac{V_f}{\Delta t} \quad (A-14) \]

where

- \( \dot{m}_{w,s} \) — mass flow required to saturate grid cell (kg/s)
- \( \rho_{\text{sat}} \) — bulk density at saturation condition (kg/m\(^3\))
- \( c_{w,\text{sat}} \) — mass fraction of water at saturation condition (kg/m\(^3\))
- \( \rho \) — current bulk density of mixture (kg/m\(^3\))
- \( c_w \) — current mass concentration of water vapor (kg/m\(^3\))
- \( V_f \) — fluid volume in the grid cell (m\(^3\))
- \( \Delta t \) — timestep(s)

The phase change rates from Eq. (A–13) and (A–14) are compared. The minimum of the two values is used in Eq. (A–12) to solve for \( T_{s,2} \) and \( \dot{m}_w \).

This process allows for the source terms of water mass into and out of the flow domain to be computed for use in the species diffusion equation. This process also provides for the source/sink of energy at the surfaces of solid obstacles through the adjustment of the obstacle temperature in conjunction with the water phase change at the surface.

A1.5.2 Phase Change in Open Volume

During a simulation, the transport of energy, mass, and the diffusion of the water may create a situation in which the relative humidity becomes greater than 100 percent in the grid cells away from the walls. If this occurs, the local temperature and relative portions of water vapor and water mist must be altered to satisfy this constraint. To accomplish the vapor-liquid mass adjustment, it is assumed that the total energy of a control volume (i.e., a FLOW-3D grid cell) is fixed, where the volume specific energy \((\rho_l)\) remains constant over time. This quantity can be expressed as

\[ (\rho_l)_2 = (\rho_l)_1 = \rho_{\text{nom}} \left( c_a C_{va} T + c_{wv} C_{vv} T - c_{wl} h_{vap} \right) \quad (A-15) \]

where

- \( (\rho_l) \) — volume-specific energy (J/m\(^3\))
- \( \rho_{\text{nom}} \) — constant nominal density provided by the user in the problem setup (kg/m\(^3\))
- \( c_a \) — mass fraction of air
- \( c_{wv} \) — mass fraction of water vapor
- \( c_{wl} \) — mass fraction of liquid water
- \( T \) — temperature (K)
- \( C_{va} \) — constant volume-specific heat of air (J/kg-K)
- \( C_{vv} \) — constant volume-specific heat of water vapor (J/kg-K)
- \( h_{vap} \) — energy of vaporization of water (J/kg)
The Subscript "1" refers to the conditions before the adjustment of the water vapor and water liquid mass fraction, and Subscript "2" refers to the conditions after the adjustment during the solver advancement in time. State "2" is one in which the vapor is at the saturation pressure at $T$. The State "1" is assumed to be when the vapor mass fraction is greater than allowed by the water vapor saturation pressure at temperature $T$. If there is no liquid water and the temperature is greater than the dewpoint, no adjustment is necessary. The temperature and concentrations in State 2 must be solved iteratively. A modified Newton-Raphson technique for solving Eq. (A–15) was found to be robust and was used in the Moisture Transport module.

**A2 RADIATION MODULE**

This section describes the theory used to model radiative heat transfer in FLOW-3D YMUZ2. The first subsection will detail the governing equations, and the next section will describe the method to determine the view factors.

**A2.1 Governing Equations for Radiation Module**

Siegel and Howell (1992) show that the heat transfer by radiation between gray, diffuse surfaces in an enclosure with a nonparticipating medium with uniform surface radiosity is given by Eq. A–16, quantifying the heat transfer rate between the surface "n" and all other participating surfaces.

\[
Q_m = \sum_{n=1}^{N} \left( \delta_{nm} - \frac{F_{n-m}}{\varepsilon_m} \right) \left( 1 - \frac{\varepsilon_m}{\varepsilon_n} \right) \frac{Q_n}{A_n} = \sum_{m=1}^{N} \left( \delta_{nm} - F_{n-m} \sigma \right) \varepsilon_n T_n^4
\]  

(A–16)

where

- $m, n$ — index of the irradiating surfaces
- $\sigma$ — Stefan-Boltzmann constant \([5.67 \times 10^{-8} \text{ w/(m}^2\text{K}^4) [1.714 \times 10^{-9} \text{ BTU/hr-ft}^2\text{-°R}^4]]\)
- $\delta_{nm}$ — Kronecker delta
- $\varepsilon_m$ — surface emissivity
- $F_{n-m}$ — configuration factor for surface "n" viewing surface "m"
- $Q_m$ — heat transfer rate from surface via radiation (W)
- $A_n$ — surface area (m$^2$)
- $T_n$ — temperature of surface (K)
- $N$ — number of surfaces active for radiation

The computation of radiation heat transfer involves manipulation of coefficient matrices produced from the heat flux and temperature terms in Eq. (A–16). That is, the heat flux at an individual surface is directly dependent on the temperatures and orientation of the other surfaces involved in radiative heat transfer. FLOW-3D YMUZ2 discretizes surfaces into many elements or cells. Defining each of these discrete elements as a single surface in Eq. (A–16) will lead to large coefficient matrices that are computationally expensive to invert and manipulate. To avoid this computational burden, radiation surface elements are defined as groups of the individual surfaces that result from the discretization process. Grouping of surfaces results in moderate matrix size that is easy to handle. However, discretization and
A–9

The surface discretization procedure and matrix formation method adopts the following method. A cell is a small but finite volume of space defined by the *FLOW-3D* grid discretization scheme. *FLOW-3D* uses the term "obstacle" to refer to solid objects that are in the flow domain. A radiation surface is a group of cells comprising a user-defined obstacle surface or a part of an obstacle surface. The temperature, \( T_m \), is taken to be the average temperature over all the cells comprising the radiation surface \( m \). For incorporating radiation effects into *FLOW-3D*, it is assumed that the surface temperatures are known from previous calculations at the current time step. Eq. (A–16) can be rearranged into the matrix equation representing a system of \( N \) equations with \( N \) unknowns, where \( N \) is the number of radiation surfaces

\[
q'' = G^{-1}HT_r 
\]

where

\( q'' \) — vector of \( N \) unknown heat fluxes

\( T_r \) — vector of \( N \) known surface temperatures raised to the fourth power

\( G \) — \( N \times N \) matrix of heat flux coefficients

\( H \) — \( N \times N \) matrix of temperature coefficients

The elements of the heat flux vector are

\[
q'' = \frac{Q_m}{A_m} \quad (A–18)
\]

The elements of the coefficient matrices are

\[
G_{n,m} = \left( \frac{\delta_{nm}}{\epsilon_m} - F_{n-m} \frac{1 - \epsilon_m}{\epsilon_m} \right) 
\]

\[
H_{n,n} = (\delta_{nm} - F_{n-m})\sigma 
\]

\( \delta_{nm} \) — Kronecker delta

\( \epsilon_m \) — emissivity of surface \( m \)

\( F_{n-m} \) — Fresnel reflectance coefficient

\( \sigma \) — Stefan-Boltzmann constant
The $m^{th}$ element of the $T_r$ vector is simply given by

$$T_m = (T_m)^4$$  \hspace{1cm} \text{(A–21)}

Once the surface heat fluxes are known, they are incorporated into the overall solution procedure to simulate the effects of radiation heat transfer at a surface. There is no method of directly incorporating the radiation heat transfer rates into the overall energy equations that are being solved for the solid and fluid materials. Instead, the surface element temperatures are adjusted prior to the next time step of the calculations. The temperature adjustment is defined as follows

$$\Delta T_m = T_m + q_m' / (\Delta t) \frac{A_m}{\rho_m C_m V_m}$$  \hspace{1cm} \text{(A–22)}

where

- $\Delta t$ — time step (s)
- $A_m$ — area of surface $m$ ($m^2$)
- $\rho_m$ — density of material in surface $m$ ($kg/m^3$)
- $C_m$ — specific heat of material $m$ (J/kg-K)
- $V_m$ — effective volume of the surface $m$ ($m^3$)

The volume, $V_m$, corresponds to a thin layer of material at the surface represented by a portion or a single grid cell at each discretized node on the surface. The temperature adjustment, $\Delta T_m$, is applied to all of the individual cells on the radiation surface.

### A2.2 Configuration Factors

*FLOW-3D YMUZ2* computes radiation configuration factors for the user-specified radiation surfaces by integrating the configuration factor basis equation using the surface geometry definitions within *FLOW-3D*.

Consider the differential areas, $dA_n$ and $dA_m$, on two differential surfaces in three-dimensional space in Figure A–2.

The configuration factors could be calculated using Eq. A–23 (Siegel and Howell, 1992).

$$F_{n-m} = \frac{1}{A_n} \int \int \frac{\cos \theta_n \cos \theta_m}{\pi S^2} dA_m dA_n$$  \hspace{1cm} \text{(A–23)}
Figure A-2. Configuration Factor Definition for 3-D Surfaces

where

\[
\cos \theta_n = \frac{\vec{n}_n \cdot \vec{S}_{nm}}{|\vec{n}_n| |\vec{S}_{nm}|} \tag{A-24}
\]

\[
\cos \theta_m = \frac{\vec{n}_m \cdot \vec{S}_{nm}}{|\vec{n}_m| |\vec{S}_{nm}|} \tag{A-25}
\]

| \vec{S}_{nm} | — distance between differential elements on the surfaces (m) \\
| \vec{n}_n | — unit normal vector for element \(dA_n\) \\
| \vec{n}_m | — unit normal vector for element \(dA_m\) \\
| \vec{S}_{nm} | — vector from element \(dA_n\) to \(dA_m\)
\[ \mathbf{S}_{mn} = \text{vector from element } dA_m \text{ to } dA_n \]

The elemental areas, \( dA_m \) and \( dA_n \), are equated to individual \textit{FLOW-3D} grid cells on each surface, and the integration is carried out by adding across all the cells comprising radiation surfaces. The standard \textit{FLOW-3D} obstacle is used to compute all the geometry and vector parameters required for surface configuration calculation. \textit{FLOW-3D} uses Cartesian geometry for its obstacle definitions. In this framework, the surface normal vector components are given by

\[
\mathbf{n} = \frac{A_{r_{i,j,k}} - A_{t_{i,j,k}}}{A_{i,j,k}} \mathbf{e}_x + \frac{A_{b_{k_{i,j,k}}}}{A_{i,j,k}} \mathbf{e}_y + \frac{A_{y_{i,j,k}} - A_{z_{i,j,k}}}{A_{i,j,k}} \mathbf{e}_z \quad (A-26)
\]

where

\( A_{r_{i,j,k}} \) = projection of obstacle area in cell \((i,j,k)\) in a plane normal to the x-axis
\( A_{b_{k_{i,j,k}}} \) = projection of obstacle area in cell \((i,j,k)\) in a plane normal to the y-axis
\( A_{y_{i,j,k}} \) = projection of obstacle area in cell \((i,j,k)\) in a plane normal to the z-axis
\( A_{i,j,k} \) = obstacle area in cell \((i,j,k)\)
\( \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z \) = unit vectors in the three coordinate directions

\textit{FLOW-3D} uses the nomenclature of "right" or "r" to mean in the positive x-direction, "back" or "bk" to mean in the positive y-direction, and "top" or "t" to mean in the positive z-direction. The element-to-element vector is computed as

\[
\mathbf{S}_{mn} = \left[ (x_{i,j,k})_m - (x_{i,j,k})_n \right] \mathbf{e}_x + \left[ (y_{i,j,k})_m - (y_{i,j,k})_n \right] \mathbf{e}_y + \left[ (z_{i,j,k})_m - (z_{i,j,k})_n \right] \mathbf{e}_z \quad (A-27)
\]

The appropriate vector component and magnitude values are substituted in Eqs. (A-23), Eqs. (A-24), and Eqs. (A-25) to compute the kernel of the double integral. The calculation procedure for the integral kernel includes checks to ensure that the surface elements can actually "see" each other. First, if either of the two cosine terms is less than zero, then the surfaces cannot see each other and the kernel is set to zero for that particular pair of surface cells. Second, if there is an obstacle blocking the view, then the kernel is also zeroed out for that particular pair of cells. This is determined by tracing a ray along the path of the ray. If any cell along this path has more than half its volume occupied by an obstacle, the kernel is zeroed out. The kernel computations are repeated and summed over the lists of cells in each surface 'n' and 'm' to compute the integral. For two-dimensional problems, the configuration integral is different than is shown in Eq. (A-23). Consider the configuration of two infinite differential strips in Figure A–3. The configuration factor between these two differential strips is given by

A–12
\[
dF_{dn-dm} = \frac{1}{2} \cos \theta_m \cos \theta_n = \frac{1}{2} \frac{\cos \theta_m \cos \theta_n}{S} dL_m
\]  

(A–28)

where

\[
S = \left| S_{nm} \right|.
\]

Eq. (A–27) can be integrated to yield the required two-dimensional configuration factor

\[
F_{n-m,2D} = \frac{1}{L_m} \int_{L_n}^{L_m} \frac{\cos \theta_m \cos \theta_n}{2S} dL_m dL_n
\]  

(A–29)

In FLOW-3D, the out-of-plane dimension in two-dimensional problems is given a finite value. Using this, a single expression can be used to compute both the integrals in Eqs. (A–22) and Eq. (A–28)

\[
F_{n-m} = \frac{C_{F2D}}{A_n} \int_{L_n}^{L_m} \frac{\cos \theta_m \cos \theta_n}{2S} dL_m dL_n
\]  

(A–30)

where

\[
C_{F2D} = \begin{cases} 
1 & \text{for three-dimensional problems} \\
\Delta x^{-1} & \text{for two-dimensional in y-z plane} \\
\Delta y^{-1} & \text{for two-dimensional in x-z plane} \\
\Delta z^{-1} & \text{for two-dimensional in x-y plane} 
\end{cases}
\]

\[
\Delta x, \Delta y, \text{ or } \Delta z = \text{out of plane thickness for the respective Two-Dimensional problem setup}
\]

\[
D = \begin{cases} 
\pi S^2 & \text{for three-dimensional problems} \\
2S & \text{for two-dimensional problems} 
\end{cases}
\]

The reciprocity relation is used to eliminate the need for integrating over all the surfaces

\[
F_{n-m} = \frac{A_n}{A_m} F_{m-n}
\]  

(A–31)

Eq. (A–29) is the operational equation that is programmed into the software for radiation configuration factors.
REFERENCES


APPENDIX B
PROGRAMMING GUIDE

To fully implement the Moisture Transport Module, some existing subroutines in FLOW-3D were modified. The new and modified subroutines are described in this appendix. The coding methodology, the description of the modified and newly introduced variables, and subroutines are described. It is intended for programmers who want to access the source code of the FLOW-3D YMUZ2 solver and may want to modify and add features in the future.

The approach and methodology for the new software and the software modifications are summarized below.

• The preprocessor routines are modified to accommodate the necessary user specifications for simulating moisture transport and radiation heat transfer processes.

• The density and energy property subroutines are modified to include the effects of variable moisture content and condensed water on these properties of the fluid mixture.

• The evaporation and condensation rates of water are computed as source terms in the chemical species diffusion equation.

• The energy associated with moisture transport and thermal radiation are computed as source terms in the thermal energy equation.

B1 SOFTWARE IMPLEMENTATION OVERVIEW

This software was developed with Compaq Visual FORTRAN 6.6c. It is expected that this software is compatible with all platforms and compilers with which FLOW-3D can be used with two exceptions. First, there is one new array added here (see subroutine NBR_INIT) that uses dynamic memory allocation. The declarations for dynamic memory allocation are compiler specific. Second, some features of the OPEN statements used to handle the radiation heat transfer output file (see subroutine RAD_CALC) are compiler specific. These lines of code will require alteration for use on other platforms or compilers. There could be other alterations necessary if the software is implemented on other platforms or with other compilers.

B2 MAJOR VARIABLES

Major variables used in the programming are described in Table B–1.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Subroutines</th>
<th>Standard/New</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ijk</td>
<td>All</td>
<td>Standard</td>
<td>Grid cell index and subscript</td>
</tr>
<tr>
<td>kajk</td>
<td>All</td>
<td>Standard</td>
<td>List of cells with a fluid/solid interface</td>
</tr>
<tr>
<td>kvjk</td>
<td>All</td>
<td>Standard</td>
<td>List of cells with a partial or full solid volume</td>
</tr>
<tr>
<td>vf</td>
<td>All</td>
<td>Standard</td>
<td>Fluid volume fraction of each cell</td>
</tr>
<tr>
<td>nbrobs</td>
<td>pcg_init, pcg_calc, rad_init, rad_calc</td>
<td>New</td>
<td>Defines the direction of a cell containing a solid material with respect to the cell indicated by the subscript</td>
</tr>
<tr>
<td>sclr</td>
<td>all</td>
<td>Standard</td>
<td>Holds fluid composition and auxiliary quantities</td>
</tr>
<tr>
<td>rhof</td>
<td>all</td>
<td>Standard</td>
<td>Nominal fluid density for energy equation</td>
</tr>
<tr>
<td>rhoe</td>
<td>all</td>
<td>Standard</td>
<td>Fluid volume-specific energy</td>
</tr>
<tr>
<td>cv1</td>
<td>e1cal, teval,</td>
<td>Standard</td>
<td>Air-specific heat</td>
</tr>
<tr>
<td>hvvap_stg</td>
<td>e1cal, teval, pcg_calc</td>
<td>New</td>
<td>Energy of vaporization of water</td>
</tr>
<tr>
<td>cvvap_stg</td>
<td>e1cal, teval, pcg_calc</td>
<td>New</td>
<td>Water vapor-specific heat</td>
</tr>
<tr>
<td>npsrf_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Number of surfaces active for phase change</td>
</tr>
<tr>
<td>imoist_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Identifier for obstacles that are infinite sources/sinks of water</td>
</tr>
<tr>
<td>ilqonly_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Identifier for obstacles that can condense water and reevaporate only what has condensed</td>
</tr>
<tr>
<td>kajk_pcg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>List of subscripts in the moisture surface area array, waobs_pcg, and moisture surface cell index array, ijk_pcg</td>
</tr>
<tr>
<td>waobs_pcg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Surface area of cells that have moisture surfaces</td>
</tr>
<tr>
<td>wvosbs_pcg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Solid volume of moisture surface cells</td>
</tr>
<tr>
<td>itw_pcg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td><em>FLOW-3D</em> index for the cell with the solid associated with this moisture surface cell</td>
</tr>
<tr>
<td>Variable</td>
<td>Subroutines</td>
<td>Standard/New</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>sclr</td>
<td>All</td>
<td>Standard</td>
<td>Stores the values of the species concentrations also stores bookkeeping values</td>
</tr>
<tr>
<td>isvap_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Total water concentration computed by species diffusion, global value</td>
</tr>
<tr>
<td>isliq_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Moisture Transport mass flux at moisture surface cells Bookkeeping only, local value only</td>
</tr>
<tr>
<td>isrh_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Relative humidity Bookkeeping only, local value only</td>
</tr>
<tr>
<td>istlq_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Total mass transfer from start of simulation Global value</td>
</tr>
<tr>
<td>swv_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Water vapor concentration Computed from total water and saturation value</td>
</tr>
<tr>
<td>iswl_stg</td>
<td>pcg_init, pcg_calc</td>
<td>New</td>
<td>Liquid water as mist; computed from total water and vapor, global value</td>
</tr>
<tr>
<td>tn</td>
<td>All</td>
<td>Standard</td>
<td>Fluid temperature</td>
</tr>
<tr>
<td>tw</td>
<td>All</td>
<td>Standard</td>
<td>Solid temperature</td>
</tr>
<tr>
<td>nrsf_stg</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Number of radiation surfaces</td>
</tr>
<tr>
<td>radl_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Radiation heat flux coefficient matrix and its inverse</td>
</tr>
<tr>
<td>rad_stg</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Information array for identifying radiation surfaces from obstacle cells surfaces</td>
</tr>
<tr>
<td>rcobs_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Density-specific heat product for radiation surfaces</td>
</tr>
<tr>
<td>kajk_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>List of subscripts in the radiation area array, waobs_rad, and radiation surface cell index array, ijk_rad</td>
</tr>
<tr>
<td>waobs_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Surface area of in cells that are part of radiation surfaces.</td>
</tr>
<tr>
<td>ijk_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td><strong>FLOW-3D</strong> ijk index for radiation surface cells</td>
</tr>
</tbody>
</table>
Table B–1. Major Variables used in FLOW-3D YMUZ2 (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Subroutines</th>
<th>Standard/New</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sa_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Area of radiation surface</td>
</tr>
<tr>
<td>sv_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Solid volume of radiation surface</td>
</tr>
<tr>
<td>cf_stg</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Radiation configuration factors</td>
</tr>
<tr>
<td>eps_stg</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Radiation surface emissivity</td>
</tr>
<tr>
<td>qflx_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Radiation heat flux</td>
</tr>
<tr>
<td>radr_rad</td>
<td>rad_init, rad_calc</td>
<td>New</td>
<td>Right side of radiation heat flux equation</td>
</tr>
</tbody>
</table>

B3 SOFTWARE DESCRIPTION

B3.1 Modified FLOW-3D Subroutines

The standard FLOW-3D code allows the user to change some of the existing subroutines and add entire new subroutines for special models or simulation algorithms. The software described here falls in both categories. The modified subroutines are those involving property evaluations for high-humidity air/water mixtures and the preprocessor subroutine for communicating user-provided input specifications to the customized software. The subroutines that fall into this category are described in Table B–2.

Table B–2. Modified Subroutines in FLOW-3D YMUZ2

<table>
<thead>
<tr>
<th>Original File Name</th>
<th>New File Name</th>
<th>Function Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1cal.f</td>
<td>e1cal_stg.f</td>
<td>Calculates fluid mass-specific energy</td>
<td>This subroutine was modified to make the fluid energy a function of temperature and composition</td>
</tr>
<tr>
<td>rhocal.f</td>
<td>rhocal_stg.f</td>
<td>Calculates fluid density</td>
<td>This subroutine required modification to make the fluid density a function of temperature and composition</td>
</tr>
</tbody>
</table>
Table B–2. Modified Subroutines in FLOW-3D YMUZ2 (continued)

<table>
<thead>
<tr>
<th>Original File Name</th>
<th>New File Name</th>
<th>Function</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>prusrd.f</td>
<td>prusrd_stg.f</td>
<td>Preprocessor input file reader</td>
<td>This subroutine was modified to define and initialize the appropriate variable for the Moisture Transport and radiation modules</td>
</tr>
<tr>
<td>rusrd.f</td>
<td>rusrd_stg.f</td>
<td>Simulation code file reader</td>
<td>This subroutine was modified to define and initialize the appropriate variable for the Moisture Transport and radiation modules</td>
</tr>
<tr>
<td>cbusr.f</td>
<td>cbusr.f</td>
<td>&quot;COMMON&quot; block definitions for customized user inputs</td>
<td>The name of this file cannot be changed because it is part of INCLUDE statements throughout the software</td>
</tr>
<tr>
<td>usrdat.f</td>
<td>usrdat.f</td>
<td>&quot;NAMELIST&quot; definition for customized user inputs.</td>
<td>The name of this file cannot be changed because it is part of INCLUDE statements throughout the software</td>
</tr>
</tbody>
</table>

B3.2 Modified and Added FLOW-3D Subroutines

The new subroutines developed for FLOW-3D YMUZ2 are described in this section through the use of pseudocode. Some of the standard customizable subroutines supplied by the standard version of FLOW-3D were extensively modified and are also described in this appendix. Minor modifications made to some of the customizable subroutines are not reported.

B3.2.1 Module pcg_rad_module

This module contains only a specification statement for defining the dynamic memory allocation of the variable nbrosb. This is a two-dimensional array that is sized at $6 \times N_{cell}$, where $N_{cell}$ is the total number of grid cells in the simulation. Dynamic memory allocation allows the dimension of this array to vary depending on the size of the problem and precludes the need to recompile the code if the simulation requires an array larger than that was anticipated for static allocation.
B3.2.2 Subroutine nbr_init

This subroutine initializes and defines the array nbrobs. This subroutine is contained in the file qsadd_pcg_rad.f.

```fortran
subroutine nbr_init
    ! nbr_init pseudocode
    ! No arguments required
    ! Get the total number of grid cells from the vf array
    Allocate the size of nbrobs to (6,nbr_siz)
    Initialize nbrobs(m,ijk) = 0
    For nob = 1 to nobs
        mincel = kvjk(nob)
        maxcel = ijkvob(m)
        for m = mincel to maxcel
            ijk = ijkvob(ijk)
            Neighbor cell indexes from mijk, pijk
            Set nbrobs(1,ijpk), nbrobs(2,imjk).....
            ! Assign nbrobs for the neighbor cells
            nbrobs(2,ijpk), nbrobs(3,ijmk),
            nbrobs(5,ijkp), nbrobs(6,ijkm) = nob
        End loop for m = mincel to maxcel
    End loop for nob = 1 to nobs
    return
```

B3.2.3 Subroutine Teval

This subroutine computes the temperature corresponding to fluid volume-specific energy and fluid composition. The underlying theory for this subroutine is described in Appendix-A (Section A1.3). The code used here must be the reverse of the code used to compute the fluid energy in subroutine e1cal so that there is a one-to-one correspondence between energy and temperature for a specific combination of air concentration, water vapor concentration, and liquid water concentration. The teval object module contained in the standard FLOW-3D environment must be removed so that the object module created by compiling the code listed here replaces the standard version. This subroutine is in the file named teval_stg.f to distinguish it from the standard version and as a reminder that a substitute version is being used.

```fortran
function teval(ijk)
    if (isvap .le. 0) stop
    ywv = sclr(ijk,isywv_stg)
    ywl = sclr(ijk,isywl_stg)
    ya = 1. – ywv – ywl
    rhomix = rhof
    teval = rhoe(ijk)/rhomix + ywl*hvvap_stg
    teval = teval/(ya*cv1+ywv*cvvap_stg)
    return
```

B–7
B3.2.4 Subroutine qsadd

This subroutine is the interface between the standard FLOW-3D code and the software for the Moisture Transport and Radiation Modules and is based on the sample code provided in the standard FLOW-3D installation. This subroutine provides for the initialization requirements of these modules and calls the modules during FLOW-3D execution. This subroutine is in the file named qsadd_pcg_rad.f to distinguish it from the standard version.

```fortran
subroutine qsadd ! No arguments are required
    if (nrsrf_stg and isvap_stg <= 0) return ! Return if phase change
    if (cycle <=0) then ! Call initialization routines for first cycle
        call nbr_init
        if (nrsrf_stg .gt. 0) call rad_init ! Initialization for radiation calculations
        if (isvap_stg .gt. 0) call pcg_init ! Initialization for Moisture Transport calculations
    endif
    if (nrsrf_stg .gt. 0) call rad_calc ! Perform radiation calculations
    if (isvap_stg .gt. 0) call pcg_calc ! Perform Moisture Transport calculations
return
```

B3.2.5 Subroutine pcg_init

There are several information items that are extracted from the FLOW-3D data arrays and stored. These procedures need to be performed only once at the start of the code execution and are separated from the Moisture Transport Module to make the execution more efficient. This subroutine provides for the initialization requirements of the Moisture Transport Module and is in the file named pcg_init_calc.f.

```fortran
subroutine pcg_init ! No arguments are required
    mpcg=0 ! Initialize the surface cell index
    npsrf_stg=0 ! and surface counter
    Do 2000 nob =1 to nobs ! Loop over all the obstacles
        if (imoist_stg(nob),ilqonly(nob) .ne. –nob) goto 2000 ! Skip if this obstacle is not
            ! moisture-active
            npsrf_stg=npsrf_stg+1 ! Increment the surface counter
            rcobs_pcg(npsrf_stg) = rcobs(nob) ! Set the rho-c product for this surface
            imlpcg_stg(npsrf_stg) = 'm' or 'l' ! Define surface as infinite source/sink
            if (ilqonly(nob) .eq. –nob) imlpcg_stg='l' ! or limited-liquid
            mincel=kajk(nob,nbl) ! Get the range of cells on the cell surface
            maxcel=kajk(nob+1,nbl)-1 ! surface ID array for this obstacle
            mmm=0 ! Initialize counter for moisture surface cells
            for m=mincel to maxcel ! Loop over all cells for this obstacle surface
                ijk-ijkobs(m) ! Get the ijk index for this obstacle surf cell
                call inijk ! get the neighbor indexes, imjk, ipjk, etc
                vcell, vwall ! Total cell volume and solid-only volume
                mmm = mmm + 1 ! Increment the moisture surfacecell index
                mpcg = mpcg + 1 ! Increment the moisture cell index limit
                if (mmm=1) ! Assign limits for the moisture cell index
                    kajk_pcg(npsrf_stg) = mpcg
            enddo
        enddo
```

B–8
B3.2.6 Subroutine pcg_calc

This subroutine calculates the mass transfer at the wall and adjusts the wall surface cell temperatures to account for the heat of vaporization effects on the heat transfer in the walls. The basic theory for this process is described in Appendix A in Section A.1.4.1. The mass transfer at walls is effectively the source term in the species diffusion equation. The adjustment of wall surface temperature is done here because there is no way for the user to directly assign the heat source terms in the solid energy equation in FLOW-3D. The pcg_calc subroutine is in the file named pcg_init_calc.f.

```fortran
subroutine pcg_calc
  ! No arguments are required
  ! Initialize the bookkeeping scalars to zero
  ! Loop over all moisture surfaces
  ! Moisture surface cell ID limit this surface
  ! FLOW-3D indexes this fluid and wall cell
  ! Decode the i,j,k from Index ijk
  ! Get the neighbor cell indexes
  ! Trap possible error condition
  ! Get the solid volume and surface area for this
  ! cell on this moisture surface
  ! If wall temp is non-existent, this is error
  ! Initialize cell temp according current energy
  ! Initialize the wall temp to current value
  ! Get current water mass fraction from previous sol'n
  ! Initialize counter for wall calculations
  ! Loop for finding wall temp and water mass flux
  ! Set the temp for humidity properties accordingly

  ! ID array. kajk_pcg is analogous to kajk
  ! Obstacle surface area for moisture surface cells
  ! FLOW-3D index for this moisture surface cell
  ! Assume this cell actually has the surface inside it
  ! Find the cell that has the wall obstacle and get its index. This is used to get right wall temp later
  ! Store the solid volume associated with this surface
  ! FLOW-3D index for solid for with this surface

end subroutine pcg_calc
```
fractions and temps  ! Bulk density at saturation compute delmmax Mass
compute delmdif  ! required to bring entire cell to saturation
delm=min(delmmax,delmdif) ! Mass entering cell if diffusion from wall is limiting
compute new wall temp or fluid temp
Check if we’re out of condensed water
If not converged on new temp, loop again for max of 25 iterations
delmrat=delm/rho(ijk)/vcell ! New water mass as fraction
sclr(ijk,isvap_stg)=
(yvacti+delmrat)/(1+delmrat) ! New value of total water fraction
sclr(ijk,isiq_stg)=
sclr(ijk,isiq_stg)-delm/delt/sa ! Sum mass flux to get values from all wall neighbors
sclr(ijk,istslq_stg)=
sclr(ijk,istslq_stg)-delm ! Total liquid accumulation
Update the FLOW-3D arrays for wall temp or fluid temp
End loop for m=mincel to maxce
End loop for ipsrf = 1 to npsurf_stg
Loop over all real cells in flow domain that are not fully blocked
yw=sclr(ijk,isvap_stg)
nitr_moist2 = 0
if (yw .le. yvsat .or. rhlim_stg .eq. 'n') skip calculations for RH shift
Iterate until converged on new saturation temp for max of 25 iterations
nitr_moist2 = nitr_moist2 + 1
ed0 = rhoe(ijk)/rhof ! Get the current value of mass-specific energy
Get yvsat, saturation and mass fraction
ed1 = current value of energy ! The energy derivative depends on whether
d1edt = derivative of energy w.r.t. temp ! there is liquid present
etrlx= de1dt/ed0 ! Newton-Raphson equation
If not converged on new temp, loop again for max of 25 iterations
Update vapor, liquid fractions, cell temp ! Get new conditions that satisfy constant energy
End loop over all real cells in flow domain that are not fully blocked
return

B3.2.7 Subroutine rad_init

There are several information items that are extracted from the FLOW-3D data arrays and stored for radiation surface areas and volumes. These procedures need to be performed only once at the start of the code execution and are separated from the main part to make the code more efficient. This subroutine also computes the radiation configuration factors and forms and then inverts the heat flux coefficient matrix for use in computing the radiation heat fluxes. The underlying theory for this process is described in Appendix A, Section A.2.2. This subroutine is in the file named rad_init_calc.f.
For nob = 1 to nobs ! Loop over all obstacles to find the one if (rad_stg(1,irsrf) .eq. nob) then ! for this radiation surface
  rcobs_rad(irsrf)=rcobs(nob)
  mincel=kajk(nob,nbl)
  maxcel=kajk(nob+1,nbl)
  mmm=0 ! Initialize radiation surface cell index
  For m=mincel to maxcel ! Loop over all FLOW3D cells
    if (cell within geometry limits) then ! that make up the surface of this obstacle
      vwall= compute obstacle volume
      if cell is empty, search the neighbor
cells for the obstacle volume
      mmm = mmm+1 ! Increment the radiation cell counters
      mrad = mrad + 1
      if (mmm=1) kajk_rad(irsrf)=mrad ! Set the min and max cell
      kajk_rad(irsrf+1)=mrad+1 ! indexes for this radiation surface
      waobs_rad(mrad) = waobs(m) ! Get the surface area for this cell
      a_rad = a_rad+waobs(m) ! Sum to get the total surface area
      v_rad = v_rad + vwall ! Sum to get the total solid volume
      of surface cells
      ijk_rad(mrad)=ijktw ! Save the index for this rad
      surface cell
    endif
  End loop for m=mincel to maxcel
End loop for nob=1 to nobs
sa_rad(irsrf) = a_rad ! Save the total area and volume
sv_rad(irsrf) = v_rad ! radiation surface
End loop for irsrf=1 to nrsrf
Compute the configuration factors
Find the minimum grid cell spacing for defining the ray tracing step
For n=1 to nrsrf ! Loop through all the surfaces
  cf_stg(n,n) ! Assume surface cannot see itself
  min_ns = kajk_rad(n) ! Get max and min indexes for
  max_ns = kajk_rad(n+1)-1 ! cells in radiation surface 'n'
  For m= n+1 to nrsrf ! Loop over surfaces m>n
    min_ms = kajk_rad(m) ! Get max and min indexes for
    max_ms = kajk_rad(m+1)-1 ! cells in radiation surface 'm'
    sum = 0 ! Initialize the CF integral
    For nn= min_ns to max_ns ! Loop over all cells in surface 'n'
      ijk_m=ijk_rad(nn)
      compute all local FLOW-3D indexes
      compute the dAn vector and magnitude ! and magnitude for element in 'n'
      For mm=man_ms to max_ms ! Loop over all cells in surface 'm'
        ijk_n=ijk_rad(nn)
        compute all local FLOW-3D indexes ! Get the cell surface area vector components
        compute the dAm vector and magnitude
        compute the vector Smn
        compute ns_ray, steps for ray tracing
        rqay = 1
        For nss=1 to ns_ray ! Trace along Smn to check for blockage
          if cell volume fraction <10-6, ray=0 ! Set factor to zero if blocked
        End loop for nss = 1 to ns_ray
        costnn = from dot product of dAn and Smn
        costmm = from dot product of dAm and Smn
    End loop for mm=man_ms to max_ms
  End loop for m= n+1 to nrsrf
End loop for n=1 to nrsrf
denom = pi*Smn for 3-D
denom = 2*Smn for 2-D problem
sum = sum + ray*costnn*costmm/denom
End of loop for mn=min_ms to max_ms
End of loop for nn=min_ns to max_ns
cf2d = 1 for 3-D
   cf2d = 1.delx or 1/dely or 1/delz for 2-D
   cf_stg(n,m) = cf2d*sum/sa_rad(n) ! compute the CF for n-m
   cf_stg(m,n) = cf_stg(n,m)*sa_rad(n)/sa_rad(m) ! Use reciprocity for CF of m-n
   End of loop for m=n+1 to nrsrf
End of loop for n=1 to nrsrf
End of loop for n=1 to nrsrf
For n=1 to nrsrf
   For m=1 to nrsrf
      kdelta = 0
      if (m=n) kdelta = 1
      radl_rad(m,n) = kdelta/eps_stg(m)-cf_stg(n,m)*(1-eps_syg(m))/eps_stg(m)
   End of loop for m=1 to nrsrf
   End of loop for n=1 to nrsrf
   call matinv(radl_rad) ! Invert the matrix for calculations later
   record the emissivities and configuration factors for review
end of subroutine rad_init

B3.2.8 Subroutine matinv

This subroutine performs a matrix inversion by modified Gaussian elimination. This is a standard numerical technique and will not be described by pseudocode here. This subroutine is called only by rad_init.

B3.2.9 Subroutine rad_calc

This subroutine computes the radiation heat transfer rate from each radiation-active surface. The routine multiplies the inverted heat flux coefficient by the temperature array to compute radiation heat flux at each surface cell. The mathematical formulation of this subroutine is described in Appendix A, Section A.2.1. The heat fluxes are used to adjust the surface cell temperatures to simulate the effects of the radiation heat flux as if it is a source or sink in the conduction energy equation for the affected cells. This subroutine is called only by qsadd.

subroutine rad_calc
For irsrf = 1 to nrsrf
   ta_rad=0
   mincel=kajk_rad(irsrf)
   maxcel=kajk_rad(irsrf+1)-1
   for m=mincel to maxcel
      ijk=ijk_rad(m)
      compute the local neighbor indexex
      ta_rad=ta_rad+tw(ijk)*waobs_rad(m)
   End loop for m=mincel to maxcel
   tr_rad(irsrf)=ta_rad/sa_rad(irsrf)
End loop for irsrf=1 to nrsrf
For n=1 to nrsrf
   trd_rad(n)=0
   End of loop for n=1 to nrsrf
   call matinv(trd_rad)
   record the emissivities and configuration factors for review
end of subroutine rad_init

B–12
for m=1 to nrsrf
  kdelta=0
  if (m=n) kdelta = 1
  radr_rad(n) = radr(n)+(kdelta-cf_stg(n,m))*tr_rad(m)**4
end loop for m=1 to nrsrf
end loop for n=1 to nrsrf

for n=1 to nrsrf
  qflx_rad(n)=0
  for m=1 to nrsrf
    qflx_rad(n)=qflx(n)
    -radl_rad(n,m)*radr(m) !Heat flux is positive away from surface
  end loop for m=1 to nrsrf
end loop for n=1 to nrsrf

prt_rad = 0 ! Record the heat fluxes for
if (t .ge. tnext) then ! processing later
  prt_rad=1
  tnext = tnext + pltdt ! Save on the FLOW-3D pltdt interval
  write t, qflx_rad
for n=1 to nrsrf !Compute the delta-T for surfaces that are not fixed temp
  dtmp_rad=0
  if (rcobs_rad(n) .gt. 0) then
    dtmp_rad = qflx_rad(n)*sa_rad(n)*delt/
              rcobs_rad(n)/sv_rad(n)
    mincel=kajk_rad(n)
    maxcel=kajk_rad(n+1)-1
    for m=mincel to maxcel
      ijktw=ijk_rad(m)
      tw(ijktw) = tw(ijktw) + dtmp_rad
    end loop for m=mincel to maxcel
  end loop for n=1 to nrsrf
end subroutine rad-calc
APPENDIX C
ADDITIONAL SAMPLE INPUT FILES

Appendix C represents the input files for the two-dimensional enclosed heat and mass transfer problem discussed in Chapter 4. This problem was investigated for four cases. The input files for the following cases are provided in this appendix.

(1) Convection only
(2) Convection with radiation
(3) Convection with moisture transport

The input file has combined convection and radiation heat transfer with moisture transport and has already been discussed in Section 4.2.4.

C1 INPUT FILE FOR CONVECTION-ONLY PROBLEM

Heat transfer in 2-D Sq. Box w/ natural convection, NO radiation, NO moisture
Ra ~ 6000 for natural convection only
Box is 0.1 m x 0.1 m long,
Adiabatic at top and bottom,
20 W generated in left wall,
300 K at outer face of right wall.
Both walls have k=1 w/(m*K), rcobs=1000*joule/(m^3*K)
Moisture at left and right walls if modeled
Radiation at left and right walls if modeled

```
$XPUT
remark='units are SI',
itb=0,      remark='Sharp interface not required',
ifvis=0,    remark='Laminar',
ifenrg=2,   remark='Solve energy eq., first order',
ifrho=1,    remark='Temp. dependent density',
gz=-9.8e-03,   remark='Gravity',
ipdis=1,    remark='Hydrostatic pressure distribution in z-direction',
ihtc=2,     remark='Evaluate heat transfer with solid conduction',
iwsh=1,     remark='Wall shear active',
delt=1.e-4,  remark='',
twfin=4000.,   remark='',
rmrhoe=0.,   remark='Density diffusion term coefficient',
rmrho=0.,    remark='Energy diffusion term coefficient',
remark=' RMRHO, RMRHOE = 1 REQUIRED FOR VAPOR TRANSPORT MODEL ',
iusrd=1,
$end

$LIMITS
$END

$PROPS
units='si',
rhof=1.1687,  remark='Bulk nominal density at 310 K for energy equation',
mu1=2.e-05,  remark='Bulk nominal dynamic viscosity density for momentum equation',
cv1=717.,    remark='Const. Vol. Specific heat of dry air',
thc1=0.026,  remark='Bulk nominal thermal conductivity for energy equation',
thexf1=0.003226,  remark='Approximate thermal expansion coefficient',
```
tstar = 300., remark='Reference temperature for thermal expansion coefficient',
remark='THEXF1, TSTAR not needed for vapor transport model',
remark='but are used in some parts of code to set up simulation',
$end
$scalar
remark='Water vapor scalars not used for this problem but need to be defined to use the ',
remark=' radiation and water transport modules together',
remark='Scalar 1 is the diffusing/advecting water',
remark='Scalar 2 (non-diffusing) is for storing the surface phase change flux values',
remark='Scalar 3 (non-diffusing) is for storing the relative humidity values',
remark='Scalar 4 (non-diffusing) is for storing the net surface liquid accumulation',
remark='Scalar 5 (non-diffusing) is for storing the vapor water concentration',
remark='Scalar 6 (non-diffusing) is for storing the liquid water (mist) concentration',
remark='Scalar 7 (non-diffusing) is for storing the calculation iterations for the',
remark=' wall mass flux',
remark='Scalar 8 (non-diffusing) is for storing the calculation iterations for the',
remark=' mist phase change in the fluid interior',
remark='Scalars 7 and 8 are helpful in tuning the value of vaprlx if necessary',
nsc=8,
isclr(1)=3, cmsc(1)=0.26e-04, scltit(1)="Tot.Water", rmsc=0.,
isclr(2)=0, cmsc(2)=0., scltit(2)="Liq.Flux",
isclr(3)=0, cmsc(3)=0., scltit(3)="Rel.Hum",
isclr(4)=0, cmsc(4)=0., scltit(4)="Net.Liq",
isclr(5)=0, cmsc(5)=0., scltit(5)="Vap.Wat",
isclr(6)=0, cmsc(6)=0., scltit(6)="Liq.Wat",
isclr(7)=0, cmsc(7)=0., scltit(7)="Itr.Wall",
isclr(8)=0, cmsc(8)=0., scltit(8)="Itr.Mesh",
$end
$bcdata
wl=1, remark='Symmetry at left to make all the heat go into the inner face',
wr=2, remark='Constant temperature wall at right',
tbc(2)=300.,
wf=1, wbk=1, remark='Symmetry at front and back',
wb=2, remark='Insulated wall',
wt=2, remark='Insulated wall',
$end
$mesh
px(1) =-0.025, py(1) =0.0, pz(1) = 0.0,
px(2) = 0.0, py(2) =1.0, pz(2) = 0.1,
px(3) = 0.1,
px(4) = 0.125,
nxcell(1)=5,
nxcell(2)=20,
nxcell(3)=5,
nxcelt= 30, nycelt=1, nzcelt=30,
$end
$obs
avrck=-3.1,
nobs = 2,
tobs(1)=0., tobs(2)=1000.,
remark='Obstacle 1.  Left channel wall.  Total heat output 50W',
C2  INPUT FILE FOR CONVECTION WITH RADIATION

Heat transfer in 2-D Sq. Box w/ natural convection, radiation, NO moisture
Ra ~ 6000 for natural convection only
Box is 0.1 m x 0.1 m long,
Adiabatic at top and bottom,
20 W generated in left wall,
300 K at outer face of right wall.
Both walls have $k=1 \text{ W/(m*K)}$, $r_{cobs}=1000 \text{ joule/(m}^3\text{K)}$
Moisture at left and right walls if modeled
Radiation at left and right walls if modeled

$xput$
  remark='units are SI',
  itb=0, remark='Sharp interface not required',
  ifvis=0, remark='Laminar',
  ifenrg=2, remark='Solve energy eq., first order',
  ifrho=1, remark='Temp. dependent density',
  gz=-9.8e-03, remark='Gravity',
  ipdis=1, remark='Hydrostatic pressure distribution in z-direction',
  ihtc=2, remark='Evaluate heat transfer with solid conduction',
  iwsh=1, remark='Wall shear active',
  delt=1.e-4, remark='',
  twfin=4000., remark='',
  rmrhoe=0., remark='Density diffusion term coefficient',
  rmrho=0., remark='Energy diffusion term coefficient',
  remark='RMRHO, RMRHOE = 1 REQUIRED FOR VAPOR TRANSPORT MODEL',
  iusrd=1,
$end$

$limits$
$end$

$props$
  units='si',
  rhof=1.1687, remark='Bulk nominal density at 310 K for energy equation',
  mu1=2.e-05, remark='Bulk nominal dynamic viscosity density for momentum equation',
  cv1=717., remark='Const. Vol. Specific heat of dry air',
  thc1=0.026, remark='Bulk nominal thermal conductivity for energy equation',
  thexf1=0.003226, remark='Approximate thermal expansion coefficient',
  tstar = 300., remark='Reference temperature for thermal expansion coefficient',
  remark='THERX1, TSTAR not needed for vapor transport model',
  remark='but are used in some parts of code to set up simulation',
$end$

$scalar$
  remark='Water vapor scalars not used for this problem but need to be defined to use the',
  remark=' radiation and water transport modules together',
  remark='Scalar 1 is the diffusing/adveacting water',
  remark='Scalar 2 (non-diffusing) is for storing the surface phase change flux values',
  remark='Scalar 3 (non-diffusing) is for storing the relative humidity values',
  remark='Scalar 4 (non-diffusing) is for storing the net surface liquid accumulation',
  remark='Scalar 5 (non-diffusing) is for storing the vapor water concentration',
  remark='Scalar 6 (non-diffusing) is for storing the liquid water (mist) concentration',
  remark='Scalar 7 (non-diffusing) is for storing the calculation iterations for the',
  remark=' wall mass flux',
  remark='Scalar 8 (non-diffusing) is for storing the calculation iterations for the',
  remark=' mist phase change in the fluid interior',
  remark='Scalars 7 and 8 are helpful in tuning the value of vaprx if necessary',
  nsc=8,
  isclr(1)=3, cmsc(1)=0.26e-04, scltit(1)='Tot.Water', rmsc=0.,
  isclr(2)=0, cmsc(2)=0., scltit(2)='Liq.Flux',
  isclr(3)=0, cmsc(3)=0., scltit(3)='Rel.Hum',

C–4
isclr(4)=0, cmsc(4)=0., scltit(4)='Net.Liq',
isclr(5)=0, cmsc(5)=0., scltit(5)='Vap.Wat',
isclr(6)=0, cmsc(6)=0., scltit(6)='Liq.Wat',
isclr(7)=0, cmsc(7)=0., scltit(7)='Itr.Wall',
isclr(8)=0, cmsc(8)=0., scltit(8)='Itr.Mesh',
$end

$bcdata
  wl=1,           remark='Symmetry at left to make all the heat go into the inner face',
  wr=2,           remark='Constant temperature wall at right',
  tbc(2)=300.,
  wf=1, wbk=1,    remark='Symmetry at front and back',
  wb=2,           remark='Insulated wall',
  wt=2,           remark='Insulated wall',
$end

$mesh
  px(1) =-0.025, py(1) =0.0,   pz(1) = 0.0,
  px(2) = 0.0,     py(2) =1.0,   pz(2) = 0.1,
  px(3) = 0.1,
  px(4) = 0.125,   nxcell(1)=5,
  nxcell(2)=20,
  nxcell(3)=5,
  nxcelt= 30,     nycelt=1,   nzcelt=30,
$end

$obs
  avrck=-3.1,
  nobss = 2,
  tobs(1)=0., tobs(2)=1000.,
  remark='Obstacle 1. Left channel wall. Total heat output 50W',
  xl(1)=-0.025, xh(1)=0.,
  kobs(1)= 1., rcobs(1)=10000.,
  twobs(1,1)=300.,
  pobs(1,1)=20., pobs(2,1)=20.,
  remark='Obstacle 2. Right channel wall. Conduction to constant temp',
  xl(2)=0.1,   xh(2)=0.125,
  kobs(2)= 1.,  rcobs(2)=10000.,
  twobs(1,2)=300.,
$end

$fl
  remark='No vapor in fluid',
  sclri(1)=0.0,
  sclri(5)=0.0,
  sclri(6)=0.0,
  presi=101325.,
$end

$bf
$end

$temp
  ntmp=1,
tempi=300.,
$end

$sotn
$end

$grafic
$end

$parts
$end

$usrdat

nrsrf_stg=2,

eps_stg(1) = .9,
rad_stg(1,1) = 1.
rad_stg(2,1) = 0.004,  rad_stg(3,1) = 0.004,
rad_stg(4,1) = -1.,  rad_stg(5,1) = 2.,
rad_stg(6,1) = 0.0,  rad_stg(7,1) = 0.2,

eps_stg(2) = .9,
rad_stg(1,2) = 2.,
rad_stg(2,2) = 0.096,  rad_stg(3,2) = 0.104,
rad_stg(4,2) = -1.,  rad_stg(5,2) = 2.,
rad_stg(6,2) = 0.0,  rad_stg(7,2) = 0.2,

iusrcf_stg=0,

istwtf_stg='tw',  remark='Energy for phase change comes from wall',

isvap_stg = 1,   remark='Define scalar index for water concentration DO NOT CHANGE',
isliq_stg = 2,   remark='Define scalar index for liquid flux DO NOT CHANGE',
isrh_stg = 3,   remark='Define scalar index for rel. hum. DO NOT CHANGE',
isliq_stg = 2,   remark='Define scalar index for liquid flux DO NOT CHANGE',
isrh_stg = 3,   remark='Define scalar index for rel. hum. DO NOT CHANGE',
isvap_stg = 5,   remark='Define scalar index for vapor water DO NOT CHANGE',
isywv_stg = 6,   remark='Define scalar index for vapor water DO NOT CHANGE',
rvap_stg=4186.,  remark='Gas constant for water vapor',
rvap_stg=289.,  remark='Gas constant for air',
vaprx_stg=0.8,  remark='Relaxation factor for phase change iterations',
rhlim_stg='y',  remark='Limit rel humidity to 100 percent',

$end

C3  INPUT FILE FOR CONVECTION WITH MOISTURE TRANSPORT

Heat transfer in 2-D Sq. Box w/ natural convection, NO radiation, moisture
Ra ~ 6000 for natural convection only
Box is 0.1 m x 0.1 m long,
Adiabatic at top and bottom,
20 W generated in left wall, 
300 K at outer face of right wall. 
Both walls have $k=1$ w/(m*K), $rcobs=1000$*joule/(m^3*K) 
Moisture at left and right walls if modeled 
Radiation at left and right walls if modeled 

$xput$
   remark='units are SI',
   itb=0,    remark='Sharp interface not required',
   ifvis=0,  remark='Laminar',
   ifenrg=2, remark='Solve energy eq., first order',
   ifrho=1,  remark='Temp. dependent density',
   gz=-9.8e-03, remark='Gravity',
   ipdis=1,  remark='Hydrostatic pressure distribution in z-direction',
   ihtc=2,   remark='Evaluate heat transfer with solid conduction',
   iwsh=1,   remark='Wall shear active',
   delt=1.e-4, remark='',
   twfin=4000., remark='',
   rmrhoe=1., remark='Density diffusion term coefficient',
   rmrho=1., remark='Energy diffusion term coefficient',
   remark=' RMRHO, RMRHOE REQUIRED FOR VAPOR TRANSPORT MODEL',
   iusrd=1,
$end$

$limits$
$end$

$props$
   units='si',
   rhof=1.1687, remark='Bulk nominal density at 310 K for energy equation',
   mu1=2.e-05, remark='Bulk nominal dynamic viscosity density for momentum equation',
   cv1=717., remark='Const. Vol. Specific heat of dry air',
   thc1=0.026, remark='Bulk nominal thermal conductivity for energy equation',
   thexf1=0.003226, remark='Approximate thermal expansion coefficient',
   tstar = 300., remark='Reference temperature for thermal expansion coefficient',
   remark='THEXF1, TSTAR not needed for vapor transport model',
   remark='but are used in some parts of code to set up simulation',
$end$

$scalar$
   remark='Water vapor scalars not used for this problem but need to be defined to use the ',
   remark=' radiation and water transport modules together',
   remark='Scalar 1 is the diffusing/adveecting water',
   remark='Scalar 2 (non-diffusing) is for storing the surface phase change flux values',
   remark='Scalar 3 (non-diffusing) is for storing the relative humidity values',
   remark='Scalar 4 (non-diffusing) is for storing the net surface liquid accumulation',
   remark='Scalar 5 (non-diffusing) is for storing the vapor water concentration',
   remark='Scalar 6 (non-diffusing) is for storing the liquid water (mist) concentration',
   remark='Scalar 7 (non-diffusing) is for storing the calculation iterations for the',
   remark=' wall mass flux',
   remark='Scalar 8 (non-diffusing) is for storing the calculation iterations for the',
   remark=' mist phase change in the fluid interior',
   remark=' Scalars 7 and 8 are helpful in tuning the value of vaprlx if necessary',
   nsc=8,
   isclr(1)=3, cmsc(1)=0.26e-04, sctlit(1)='Tot.Water', rmssc=0.,
isclr(2)=0, cmssc(2)=0., scltit(2)='Liq.Flux',
isclr(3)=0, cmssc(3)=0., scltit(3)='Rel.Hum',
isclr(4)=0, cmssc(4)=0., scltit(4)='Net.Liq',
isclr(5)=0, cmssc(5)=0., scltit(5)='Vap.Wat',
isclr(6)=0, cmssc(6)=0., scltit(6)='Liq.Wat',
isclr(7)=0, cmssc(7)=0., scltit(7)='Ltr.Wall',
isclr(8)=0, cmssc(8)=0., scltit(8)='Ltr.Mesh',
end

$bcdata
   w=1, remark='Symmetry at left to make all the heat go into the inner face',
   w=2, remark='Constant temperature wall at right',
   tbc(2)=300.,
   w=1, wb=1, remark='Symmetry at front and back',
   wb=2, remark='Insulated wall',
   wt=2, remark='Insulated wall',
end

$mesh
   px(1)=-0.025, py(1)=0.0, pz(1)=0.0,
   px(2)=0.0, py(2)=1., pz(2)=0.1,
   px(3)=0.1, px(4)=0.125,
   nxcell(1)=5,
   nxcell(2)=20,
   nxcell(3)=5,
   nxcell=30, nycell=1, nzcell=30,
end

$obs
   avrck=-3.1,
   nobs=2,
   obs(1)=0., obs(2)=1000.,
   remark='Obstacle 1.  Left channel wall.  Total heat output 50W',
   x(1)=-0.025, xh(1)=0.,
   kobs(1)=1., rcobs=10000.,
   twobs(1,1)=300.,
   pobs(1,1)=20., pobs(2,1)=20.,
   remark='Obstacle 2.  Right channel wall. Conduction to constant temp',
   x(2)=0.1, x(2)=0.125,
   kobs(2)=1., rcobs(2)=10000.,
   twobs(1,2)=300.,
end

$fl
   remark='No vapor in fluid',
   scrlr(1)=0.0,
   scrlr(5)=0.0,
   scrlr(6)=0.0,
   presi=101325.,
end

$bf
end
$temp
   ntmp=1,
   tempi=300.,
$end

$motn
$end

$grafic
$end

$parts
$end

$usrdat
   istwtf_stg='tw', remark='Energy for phase change comes from wall',
   imoist_stg(1)=-1,
   imoist_stg(2)=-2,
   isvap_stg = 1, remark='Define scalar index for water concentration DO NOT CHANGE',
   isliq_stg = 2, remark='Define scalar index for liquid flux DO NOT CHANGE',
   isrh_stg = 3, remark='Define scalar index for rel. hum. DO NOT CHANGE',
   istlq_stg = 4, remark='Define scalar index for tot.liq. accum. DO NOT CHANGE',
   isywv_stg = 5, remark='Define scalar index for vapor water DO NOT CHANGE',
   isywl_stg = 6, remark='Define scalar index for liquid water (mist) DO NOT CHANGE',
   hvvap_stg=2304900., remark='Heat of vaporization',
   cvvap_stg=1370., remark='Water vapor specific heat (const. vol.)',
   cvliq_stg=4186., remark='Water liquid specific heat (const. vol.)',
   rvap_stg=416., remark='Gas constant for water vapor',
   rgas_stg=289., remark='Gas constant for air',
   vaprlx_stg=0.8, remark='Relaxation factor for phase change iterations',
   rhlim_stg='y', remark='Limit rel humidity to 100 percent',
$end