2003

Numerical techniques used in droplet evaporation modeling

Robert D. Shearrer
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Shearrer, Robert D., "Numerical techniques used in droplet evaporation modeling" (2003). *Master's Theses*. 2488.

DOI: <https://doi.org/10.31979/etd.wsqe-e4fu>
https://scholarworks.sjsu.edu/etd_theses/2488

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

NUMERICAL TECHNIQUES USED IN DROPLET EVAPORATION MODELING

Thesis
Master of Science in Aerospace Engineering

by
Robert D. Shearrer

Department of Mechanical and Aerospace Engineering
San José State University
December 2003

UMI Number: 1418717

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform 1418717

Copyright 2004 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© 2003
Robert D. Shearrer
ALL RIGHTS RESERVED

Approved for the Department of Mechanical and
Aerospace Engineering

10-30-2003

Dr. Nikos Mourtos

10/31/03

Dr. Jinny Rhee

10/31/03

Dr. Periklis E. Papadopoulos

Approved for the University

Abstract
NUMERICAL TECHNIQUES USED IN DROPLET EVAPORATION
MODELING

by Robert D. Shearrer

This thesis presents the development of an explicit solution for the velocity value in the convective transport term of the governing equations of a multiple component-evaporating droplet. A FORTRAN subroutine is written for the droplet evaporation-modeling program MULTEVAP that is used to predict evaporation rate, droplet temperature and concentration profiles in a multi-component, evaporating droplet. The droplet is represented by the coupled species mass balance and thermal energy balance. Accurate prediction of the concentration profiles requires the solution of the convective transport term in the governing equations. A spatial resolution analysis shows good convergence when 100 nodes are used in the liquid phase. The gas phase velocity values calculated by MULTEVAP are compared to those calculated using a pseudo-steady state analysis. The velocity values reported by MULTEVAP differ slightly in value from those reported in the pseudo-steady state analysis. The source of the difference in reported values is investigated.

Table of Contents

List of Figures	vii
Variable Description	viii
1. Motivation	1
2. Objective	1
3. Thesis Outline	2
4. Previous Work	3
4.1. Droplet Drying.....	3
4.2. Droplet Evaporation.....	4
5. Physical Problem	5
6. Model System	6
6.1. Model Outputs	8
7. Governing Equations	9
7.1. Species Mass Balance.....	9
7.1.1. Boundary Conditions for the Species Mass Balance	11
7.2. Energy Balance.....	13
7.2.1. Boundary Conditions for the Energy Balance.....	14
7.3. Change of Coordinates	15
7.4. Dimensionless Variables	17
8. Derivation of Equations	18
8.1. Liquid Phase - Interior Nodes.....	18
8.2. Liquid Phase - Droplet Surface	24
8.3. Liquid-Gas Boundary	26
8.4. Gas Phase - Droplet Surface	26

8.5. Gas Phase - Interior Nodes	31
8.6. Matrix Structure.....	34
9. Solving the Linear Set of Equations	37
9.1. LU Decomposition.....	37
9.2. Performing the LU Decomposition	40
9.3. Pivoting.....	41
10. MULTEVAP Program Structure	44
11. Results.....	45
11.1. Future Work.....	47
References.....	48
Appendix 1 – FORTRAN Subroutines	50

List of Figures

Figure 1 - Spray Dryer Environment	5
Figure 2 - Single Droplet Model System showing Radial Coordinate	6
Figure 3 - Sample MULTEVAP Output showing species weight fractions across the droplet radius over time	8
Figure 4 - Transformed Coordinate System	15
Figure 5 - Coefficient Matrix A showing tri-diagonal structure	34
Figure 6 - Coefficient Matrix A showing vertical and horizontal symmetry	35
Figure 7 - The coefficient matrix A with the velocity vector <i>v</i> and the right-hand-side vector <i>b</i>	36
Figure 8 – Liquid phase velocity for different numbers of nodes	45
Figure 9 – Gas phase velocity as calculated by MULTEVAP and a pseudo-steady state analysis	46

Variable Description

a_0	- initial droplet radius, $[m]$
$a(t)$	- time varying droplet radius, $[m]$
$a^*(\tau)$	- dimensionless time varying droplet radius
A	- coefficient matrix in <i>Velocity</i> Subroutine
b	- right-hand-side vector in <i>Velocity</i> Subroutine
c_p	- constant pressure specific heat, $[J/kg]$
D_i	- diffusion coefficient of species i through the key species, $[m^2/s]$
D_i^*	- dimensionless diffusion coefficient of species i through the key species
H_i	- enthalpy of species i , $[J]$
k	- thermal conductivity, $[W/mK]$
L	- Lower decomposed matrix
M_i	- molecular weight of species i , $[kg/kmol]$
p	- pressure, $[Pa]$
r	- radial coordinate, $[m]$
R	- ideal gas constant, $[m^3 \text{ bar}/kmol \text{ K}]$
t	- time, $[s]$
T	- temperature, $[K]$
u_l	- liquid-gas interface velocity (droplet surface), $[m/s]$
u_l^*	- dimensionless interface velocity
U	- Upper decomposed matrix
x	- dimensionless radial coordinate inside droplet in the liquid phase

- y - dimensionless radial coordinate outside droplet in the gas phase
- α_{ij} - row i , column j element in lower element matrix of LU Decomposition
- β_{ij} - row i , column j element in upper element matrix of LU Decomposition
- ρ - bulk density, $[kg/m^3]$
- τ - dimensionless time
- θ - dimensionless temperature
- ω_i - weight fraction of species i
- U - bulk velocity term, $[m/s]$
- U^* - dimensionless bulk velocity

1. Motivation

Large molecule drug delivery to the bloodstream via the deep lung provides systemic action. (Patton and Platz, 1992). Delivery of the drug in a powder form provides shelf life advantages. The drug in powder form is packaged in individual doses. For delivery to the blood stream, the drug dose is removed from the package and aerosolized into a respirable cloud using a delivery device. The aerosolized drug is inhaled into the deep lung where it is absorbed into the patient's bloodstream.

A major challenge in producing respirable powders is imparting characteristics to the particles that will aid in their dispersibility. Two characteristics that promote dispersibility are a hydrophobic surface and a highly wrinkled morphology (Foss, 1999). A model of the evaporation process of a single droplet from the liquid to solid state is used to predict radial concentration profiles of different species existing in the droplet. From this the dominant surface species and particle morphology can be predicted. The effects of different formulation and processing parameters on particle formation can be explored quickly and easily. Droplet temperature and evaporation rate are also predicted (Foss, 1999).

The program MULTEVAP models the evaporation process of a multiple component droplet suspended in a gas environment. The radial concentration profiles and rate of evaporation are found by solving the coupled mass and energy balances inside and outside the droplet. The code uses a stable numerical method to solve the mass and energy balances. The spatial parts of the equations are converted from differential to algebraic expressions using finite differences. The time derivatives then become a series of coupled ordinary differential equations solved using an appropriate integration technique (Foss, 1999).

2. Objective

The present work is an extension of the work done by Foss (1999). Foss developed MULTEVAP, a modeling program to track the concentration profiles of species in an evaporating droplet and predict droplet temperature, morphology and evaporation rate. MULTEVAP originally used an implicit time stepping routine that was accurate but slow. The objective of this thesis is to develop an explicit solution technique for the velocity term in the convective component of the governing equations. The advantage of this technique will be to enable the use of an explicit time integration technique that will speed computation.

3. Thesis Outline

Each subsequent thesis section is described below.

Section 4, Previous Work, contains a summary of journal publications that are related to the work presented in this thesis. The types of publications found in this field are discussed.

Section 5, Physical Problem, describes the real world scenario that this model is attempting to represent. Spray drying is the technique used to produce the powders under consideration. The process is described in relation to the production of pharmaceutical powders.

Section 6, Model System, describes the simplified system being modeled. The ability of the model to capture the real-world process is discussed. The model domain is defined and the node distribution is discussed. A sample of the program output is given.

Section 7, Governing Equations, presents the equations used to define the problem. The coupled mass and energy equations are presented in spherical coordinates. The solution technique is presented along with the governing equation boundary conditions. A change of coordinates is presented and the non-dimensional variables are defined.

Section 8, Derivation of Equations, shows how the coupled equations are written for each node in the model. The liquid phase, gas phase, and interface conditions are presented. The coefficient matrix structure is shown.

Section 9, Solving the Linear Set of Equations, describes the process used to solve the linear set of equations. LU Decomposition is used to invert the coefficient matrix and solve for the velocity component of the convective terms.

Section 10, MULTEVAP Program Structure, describes the structure of the main program MULTEVAP. The location of the Velocity subroutine in the main program and how the velocity term is used are discussed.

Section 11, Results, compares the velocity values in the gas phase calculated by MULTEVAP with values found using a quasi-steady state approach.

4. Previous Work

Droplet evaporation is an area of interest for a wide range of engineering applications including combustion, food dehydration, and pharmaceuticals (Sezen, 2000); (Hecht and King, 2000); (Patton and Platz, 1992). Presented below is a review of journal publications that are related to this field. The publications are divided into two categories; publications related to the field of droplet drying in which a solid, either dissolved or suspended, is contained in the evaporating droplet as is often found in spray drying applications; and, publications related to the field of droplet evaporation, in which the droplet consist of one or more volatile components as is often found in combustion applications.

This thesis presents work applied to the field of droplet drying. Radial species concentration profiles of a drying droplet predict the final particle's surface chemistry and morphology and provide insight into its dispersibility. The numerical solution presented here provides additional insight to the field by providing a detailed description of an explicit solution for the velocity in the convective transport terms of the governing equations. This technique can be incorporated into other analytical and numerical evaporation models that include this velocity term.

4.1. Droplet Drying

Nesic and Vodnik (1991) describe the drying kinetics of a droplet containing dissolved or dispersed solids. The droplet weight and temperature were monitored during evaporation while suspended in a controlled air stream. Experimental results were in good agreement with a computer simulation solved both numerically and analytically.

Hecht and King (2000) describe experimental measurements of drying rate, loss rate of a component, drop temperature and the physical appearance of a single droplet undergoing drying. Precise measurements during the drying of coffee or sucrose solutions describe the mass loss during a process called morphological development.

Lin, Zhang, and Gentry (2000) develop a model to describe the formation of a shell. The work describes several explanations to explain the morphology developed during shell formation. Pressure build-up within the shell is listed as a factor in the final particle morphology.

Farid (2003) developed a model to predict the change in droplet mass and temperature during drying. The model accounts for droplet shrinkage and a temperature gradient inside the droplet.

4.2. Droplet Evaporation

Renksizbulut, Nafziger, and Li (1991) investigate the evaporation of droplets in high temperature air streams at intermediate Reynolds numbers. A mass transfer correlation is developed based on the Sherwood number.

Kneer, Schneider, Noll, and Wittig (1993) make modifications to the Diffusion-Limit Model to account for the variation of liquid properties due to large temperature and concentration gradients within the droplet. Effects on vaporization were studied at two gas temperatures; the effects of the change in physical properties between these two temperatures were investigated. It was found that single component models cannot accurately predict the evaporation behavior of a multi-component droplet.

Tamim and Hallett (1995) present a droplet evaporation model where the mixture composition, properties, and vapor-liquid equilibrium are described by the methods of continuous thermodynamics. The method attempts to model the behavior of droplets with hundreds or thousands of components by taking into account the properties of every component.

Elperin and Krasovitev (1995) investigate the evaporation of slurry droplets using a quasi-steady approximation. The effects of compressibility and filtration of a gas-vapor mixture through the shell are discussed. This study finds that the diffusion model, which neglects Stefan flow, incorrectly describes evaporation of a slurry droplet at the final stage of drying.

Sezen (2000) develops analytical expressions for mass concentrations of components inside and outside an evaporating multi-component fuel droplet. Unsteady droplet evaporation of size smaller than 150 microns is studied.

Yarin, Brenn, Kastner, and Tropea (2002) study the drying behavior of liquid-solid suspensions undergoing acoustically levitated drying. The researchers characterize the drying process as consisting of two stages; an initial stage of diameter reduction, followed by a second stage of constant diameter, solvent evaporation. The study takes into account the convective influence of the acoustic field on the mass transfer.

5. Physical Problem

Figure 1 - Spray Dryer Environment

Spray drying produces the powder particles used for inhalation. In spray drying, a liquid solution containing the drug dissolved in a solvent is sprayed, or atomized, into a drying chamber (Masters, 1985). The atomized solution forms a dispersion of multiple sized droplets that are mixed with a swirling drying gas. The interaction of the liquid droplets with the dry air promotes evaporation. The swirling gas-liquid drying environment is composed of regions of varying temperature, humidity, and droplet concentration. As a result, droplets see different drying environments and undergo varying drying histories depending on their size and location in the drying chamber. Spray drying is a continuous process where new droplets are being added to the drying chamber as dry particles are removed (Masters, 1985).

6. Model System

Figure 2 - Single Droplet Model System showing Radial Coordinate

The model system is a single droplet consisting of one or more non-volatile species dissolved in one or more volatile solvents. The droplet is suspended in a gas environment. The volatile components in the droplet, when exposed to the gas surroundings, begin to evaporate and diffuse away from the droplet. As the solvent(s) evaporate the droplet cools and heat is conducted from the surroundings into the droplet. The droplet reaches a quasi-steady temperature just below the wet bulb temperature of the gas (Foss, 1999; Farid, 2003). The evaporation of the solvent from the droplet causes a change in the droplet density inducing bulk flow inside the droplet. The remaining species diffuse throughout the droplet and the droplet surface moves inward, shrinking the droplet size. Evaporation continues until all volatile species leave the droplet and a dry powder particle remains. The volatile species evaporating from the droplet diffuse into the gas phase and create a humid environment that affects the droplet evaporation rate (Foss, 1999).

A single droplet model is representative of the real world spray dryer environment due to the rapid mixing that occurs inside the dryer. The rapid mixing disperses the droplets throughout the chamber preventing one droplet's evaporation from affecting the evaporation rate of its neighboring droplets. The initial, dense concentration of droplets in the spray plume rapidly becomes a sparse dispersion of low volume liquid droplets in a high volume of drying gas. The rapid drying of small droplets can occur faster than the rate of spray plume dispersion, however, the low gas stream temperature encountered at the exit of the sonic flow atomizer delays the onset of droplet evaporation maintaining the validity of a single, isolated droplet evaporation model.

Simplifications and assumptions made to the model are based on the droplet size. The droplets typically dealt with in the application of spray-dried particles for respiratory inhalation are on the order of a few microns in diameter. Droplets this small move slowly relative to the surrounding gas flow, therefore, gas flow around the droplet does

not effect droplet evaporation. The evaporation rate of the droplet is influenced principally by diffusion of the solvent vapor away from the droplet and conduction of heat into the droplet from the surrounding environment. The small size prevents internal mixing meaning mass transfer is also dominated by diffusion. Because the droplet is not effected by external gas flow and there is no internal mixing, the temperature inside the droplet is assumed to be uniform (Tamim and Hallett, 1995; Sezen, 2000). In addition, the temperature and concentration profiles inside the droplet are assumed to be spherically symmetrical reducing the problem to a one-dimensional transient solution (Foss, 1999).

The one-dimensional model coordinate system is shown in Figure 2. The radial domain, r , extends from the droplet center, $r = 0$, to a point, $r = \infty$, where the velocity is zero. The radial domain is divided into nodes at which the governing equations are solved. The model incorporates a non-uniform node spacing that allows nodes to be bunched close together at regions of interest, such as near the droplet surface where species concentrations change substantially over a short distance. Node spacing is further apart in regions of more uniform species concentration to speed computing time (Foss, 1999).

The radial concentration profiles and evaporation rates are found by solving the coupled mass and energy balances inside and outside the droplet. The momentum balance is neglected due to the small velocities encountered. An accurate prediction of the droplet evaporation rate requires tracking the moving droplet surface. This is done by transforming the droplet coordinate system to one in which the droplet boundary moves with the droplet surface. This ensures a node is always at the droplet surface. The convective terms in the mass and energy balances represent bulk flow inside the droplet and the net outflow of solvent vapors in the gas phase. They are included to get an accurate prediction of species concentration profiles in the droplet (Foss, 1999).

6.1. Model Outputs

Figure 3 - Sample MULTEVAP Output showing species weight fractions across the droplet radius over time.

MULTEVAP tracks the radial concentrations profiles of each species in the droplet as evaporation occurs. Output from the model gives a picture of the droplet composition during the drying process. Figure 3 shows the radial concentration profiles of a droplet consisting of one solvent and one non-volatile species at different time points in the evaporation process.

As the droplet evaporates the weight fraction of the non-volatile species increases. The profiles at subsequent time points show higher concentrations of the non-volatile species as well as the shrinking droplet radius. The higher weight fractions at the droplet surface shown a concentration of the non-volatile species there due to either that species surface affinity or species being swept up by the moving droplet surface. At the end of the drying process, 275 milliseconds, the solvent is completely evaporated and the weight fraction of the non-volatile species is 1.0. The droplet radius has shrunk from 10 microns to 2.5 microns.

7. Governing Equations

The governing equations are presented in this section along with the change in variables used to solve for the moving droplet surface.

7.1. Species Mass Balance

The species mass balance for species i is:

$$\rho \frac{\partial \omega_i}{\partial t} + \rho v \frac{\partial \omega_i}{\partial r} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \rho D_{i, \text{mix}} \frac{\partial \omega_i}{\partial r} \right) \quad (7.1)$$

Bulk Velocity Term

where the first term represents the transient accumulation of component i , the second term represents the convected mass flux and the third term, on the right-hand-side of equation (7.1), represents the diffusive mass flux (Foss, 1999). The convected mass flux term is necessary in the liquid phase to account for bulk flow within the droplet and in the gas phase to account for the net outflow of the solvent vapors, known as Stephan flow (Foss, 1999). Fick's Law is used to express the diffusive mass flux in terms of the mass fraction gradient, however cross-diffusion terms are considered negligible (Foss, 1999).

The number of equations is reduced by solving the weight fraction of the last species from the weight fraction of the other species:

$$\omega_n = 1 - \sum_{i=1}^{n-1} \omega_i \quad (7.2)$$

For n species, the number of species mass balances required is $n-1$ (Foss, 1999).

The focus of the thesis work is to solve for the bulk velocity in the convective transfer terms of the mass and energy balances. This is done by simultaneously solving the equation of continuity at all points inside and outside the droplet:

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \rho v) = 0 \quad (7.3)$$

Bulk Velocity Term

The total density is related to the species weight fractions and temperature by an equation of state:

$$\rho = \rho(\omega_i, T) \quad (7.4)$$

In the liquid phase an additive equation of state is used:

$$\rho_{liquid} = \sum_{i=1}^{nl1} \omega_i \rho_{i, pure} \quad (7.5)$$

where $nl1$ is the number of species in the liquid phase (Foss, 1999). In the gas phase the ideal gas law is used, expressed in terms of weight fractions:

$$\rho_{gas} = \frac{P}{RT \sum_{i=1}^{nl2} \frac{\omega_i}{M_i}} \quad (7.6)$$

where $nl2$ is the number of species existing in both the liquid and gas phases (Foss, 1999). Because the weight fractions are required to find the velocities, and the velocities are required to find the weight fractions, equations (7.1) through (7.4) are solved iteratively (Foss, 1999)

7.1.1. Boundary Conditions for the Species Mass Balance

Boundary conditions for the weight fractions and velocities are required at the droplet center, at the liquid-gas interface, and at the edge of the gas phase. At the droplet center the velocity and the mass flux for each species are zero (Foss, 1999).

$$v \Big|_{r=0} = \frac{\partial \omega_i}{\partial r} \Big|_{r=0} = 0 \quad (7.7)$$

At the liquid-gas interface, the species jump conditions are used, where Fick's Law is employed neglecting cross-diffusion terms (Foss, 1999).

$$\left[-\rho D_{i,mix} \frac{\partial \omega_i}{\partial r} + \rho \omega_i (v - u_l) \right]_{r=a}^{(liquid)} = \left[-\rho D_{i,mix} \frac{\partial \omega_i}{\partial r} + \rho \omega_i (v - u_l) \right]_{r=a}^{(gas)} \quad (7.8)$$

Another condition at the droplet surface requires that the volatile species be in chemical equilibrium. This is satisfied by equating the chemical activities of the volatile species in both phases. Since for an ideal gas the gas phase activity of a species is equal to its partial pressure, the activity of that species in solution is equal to its relative vapor pressure (Foss, 1999):

$$a_i \Big|_{r=a}^{liq} = \frac{P_i \Big|_{r=a}^{gas}}{p_i^{sat}(\omega_i, T) \Big|_{r=a}} \quad (7.9)$$

The pressures are related to the weight fractions using the ideal gas law. MULTEVAP uses the partial pressure of the volatile components at the droplet surface to calculate the weight fraction of the volatile component in the gas phase.

The overall mass balance has a jump condition that allows the equation of continuity, (7.3), to be solved for both phases. This jump condition relates the velocities in the liquid and gas phases at the droplet surface (liquid-gas interface) (Foss, 1999):

$$\rho^{(liquid)} \left(v^{(liquid)} - u_I \right) \Big|_{r=a} = \rho^{(gas)} \left(v^{(gas)} - u_I \right) \Big|_{r=a} \quad (7.10)$$

Far from the surface of the droplet the radial velocity is zero, and the concentrations of the gas phase components can vary with time:

$$\begin{aligned} v \Big|_{r=\text{inf.}} &= 0 \\ \omega_i \Big|_{r=\text{inf.}} &= \omega_{i,\text{inf.}}(t) \end{aligned} \quad (7.11)$$

7.2. Energy Balance

As the droplet evaporates, the heat of vaporization acts to cool the droplet and heat is conducted from the surroundings into the droplet (Foss, 21). At the end of evaporation the droplet temperature rises to the temperature of the surrounding gas. The uniform droplet temperature means we do not require an energy balance inside the droplet. Outside the droplet there is a temperature variation from the droplet surface to the gas phase boundary. An energy balance is solved in the gas phase. The energy balance in the gas phase in terms of temperature is:

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \underset{\substack{\uparrow \\ \text{Bulk Velocity Term}}}{v} \frac{\partial T}{\partial r} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 k \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \rho \sum_{i=1}^{nl2} H_i D_{i, \text{mix}} \frac{\partial \omega_i}{\partial r} \right) \quad (7.11)$$

The first terms represents the thermal capacitance of the droplet; the second term represents the convective heat transfer for heat convected away from the droplet by Stephan flow. The third term, or first term on the right-hand-side, represents the conductive heat transfer between the droplet and the surroundings; Fourier's law of conduction is applied. The last term represents the enthalpy that is carried by the diffusing species (Foss, 1999).

The droplet temperature is found from an energy balance about the sphere:

$$\left. \frac{a \rho C_p}{3} \frac{\partial T}{\partial t} \right|_{\text{sphere}} = \left[k \frac{\partial T}{\partial r} + \rho \sum_{i=1}^n H_i D_{i, \text{mix}} \frac{\partial \omega_i}{\partial r} - \rho \bar{H} \underset{\substack{\uparrow \\ \text{Bulk Velocity Term}}}{(v - u_l)} \right]_{r=a}^{\text{gas}} + \rho_{\text{liq}} \Delta H_{\text{vap}} \frac{\partial a}{\partial t} \quad (7.12)$$

where the first term represents the thermal capacitance of the droplet, the second term represents the flux of heat to the surroundings at the droplet surface, and the last term represents the latent heat of vaporization (Foss, 1999).

7.2.1. Boundary Conditions for the Energy Balance

The gas temperature at the droplet surface equals the droplet temperature (Foss, 1999):

$$T_{sphere} = T|_{r=a}^{(gas)} \quad (7.13)$$

Far from the sphere, the temperature is determined from the time dependent boundary condition given by the local gas conditions around the droplet in the spray dryer (Foss, 1999):

$$T|_{r=\text{inf.}} = T_{\text{inf.}}(t) \quad (7.14)$$

7.3. Change of Coordinates

A change of coordinates is made so the domain moves with the moving droplet surface. The coordinate transformation also allows a solution to be found at a point far from the droplet surface in the gas phase. Shown are the radial domain and the transformed coordinate system.

Figure 4 - Transformed Coordinate System

In the radial domain, r , the droplet radius is represented by the time-dependent variable $a(t)$. The droplet radius shrinks as the droplet evaporates and the surface moves inward. The radial domain extends from 0 at the droplet center to a point sufficiently far away in the gas phase. The droplet interior, the liquid phase, is transformed to the x coordinate; and the droplet surroundings, the gas phase, is transformed to the y coordinate by the following equations:

$$x = \frac{r}{a(t)} \quad (7.15)$$

$$y = \frac{a(t)}{r} \quad (7.16)$$

The transformed coordinates in the liquid phase vary from $x=0$ at the droplet center to $x=1$ at the droplet surface. In the gas phase y varies from 1 at the droplet surface to 0 at the gas phase boundary. The change of coordinates recasts the radial derivatives as follows:

$$\frac{\partial}{\partial r} = \frac{\partial x}{\partial r} \frac{\partial}{\partial x} \quad (7.17)$$

Using equation (7.15):

$$\frac{\partial}{\partial r} = \frac{1}{a_0 a^*} \frac{\partial}{\partial x} \quad (7.18)$$

Likewise, using equation (7.16):

$$\frac{\partial}{\partial r} = \frac{\partial y}{\partial r} \frac{\partial}{\partial y} \quad (7.19)$$

$$\frac{\partial}{\partial r} = -\frac{y^2}{a_0 a^*} \frac{\partial}{\partial y} \quad (7.20)$$

An extra term is added to the time derivatives:

$$\frac{\partial}{\partial t} = \frac{D_{1,0}^{gas}}{a_0^2} \left[\frac{\partial}{\partial \tau} - \frac{x u_I^*}{a^*} \frac{\partial}{\partial x} \right] \quad (7.21)$$

$$\frac{\partial}{\partial t} = \frac{D_{1,0}^{gas}}{a_0^2} \left[\frac{\partial}{\partial \tau} + \frac{y u_I^*}{a^*} \frac{\partial}{\partial y} \right] \quad (7.22)$$

This additional convective term in the mass and energy balances accounts for the moving coordinate system (Foss, 1999).

7.4. Dimensionless Variables

Some variables are cast in dimensionless form for ease of use:

$$\tau = \frac{D_{1,0}^{gas} t}{a_o^2} \quad (7.23)$$

τ is the dimensionless time variable with respect to the diffusion time constant of the major volatile component in the gas phase where $D_{1,0}^{gas}$ is the diffusivity of the major volatile component in the gas phase; a_0 is the initial droplet radius (Foss, 1999).

$$a^*(\tau) = \frac{a(\tau)}{a_0} \quad (7.24)$$

$$v^* = \frac{a_0 v}{D_{1,0}^{gas}} \quad (7.25)$$

v is the bulk velocity term in the convective component of the governing equations.

$$u_I^* = \frac{a_0 u_I}{D_{1,0}^{gas}} \quad (7.26)$$

$$D_i^* = \frac{D_i}{D_{1,0}^{gas}} \quad (7.27)$$

$$\theta = \frac{T - T_{ref}}{T_{ref}} \quad (7.28)$$

where $T_{ref} = T_\infty$, a point sufficiently far from the droplet surface.

8. Derivation of Equations

To solve for the bulk velocity term in the convective component of the governing equations, the equation of continuity is written for each node in the domain. These equations form a set of linear, coupled equations. When arranged in matrix form the velocity coefficients form a coefficient matrix, A .

8.1. Liquid Phase - Interior Nodes

The equation of continuity (7.3) recast with dimensionless variables in the transformed coordinates is:

$$\frac{\partial \rho_{liq}}{\partial \tau} = \frac{x u_I^*}{a^*} \frac{\partial \rho}{\partial x} - \frac{1}{x^2 a^*} \frac{\partial}{\partial x} (x^2 \rho v^*) \quad (8.1)$$

An equation of state is used to relate the total liquid density to the liquid weight fraction and temperature:

$$\rho = \rho(\omega_i, T) \quad \text{for } i = 1 \dots n-1 \quad (8.2)$$

$$\frac{\partial \rho_{liq}}{\partial \tau} = \frac{\partial \rho_{liq}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \tau} + \frac{\partial \rho_{liq}}{\partial T} \frac{\partial T}{\partial \tau} \quad (8.3)$$

Temperature is non-dimensionalized using equation (7.28):

$$\frac{\partial T}{\partial \tau} = T_{ref} \frac{\partial \theta}{\partial \tau} \quad (8.4)$$

Casting temperature, T , in the dimensionless variable θ form:

$$\frac{\partial}{\partial T} = \frac{\partial \theta}{\partial T} \frac{\partial}{\partial \theta} \quad (8.5)$$

Differentiating eq. (7.28) gives:

$$\frac{\partial}{\partial T} = \frac{1}{T_{ref}} \frac{\partial}{\partial \theta} \quad (8.6)$$

Substituting eq. (8.4) and eq. (8.6) into eq. (8.3):

$$\frac{\partial \rho_{liq}}{\partial \tau} = \frac{\partial \rho_{liq}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \tau} + \frac{\partial \rho_{liq}}{\partial \theta} \frac{\partial \theta}{\partial \tau} \quad (8.7)$$

Combining the equation of continuity (8.1) and equation (8.7):

$$\frac{xu_I^*}{a^*} \frac{\partial \rho}{\partial x} - \frac{1}{x^2 a^*} \frac{\partial}{\partial x} (x^2 \rho v^*) = \frac{\partial \rho_{liq}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \tau} + \frac{\partial \rho_{liq}}{\partial \theta} \frac{\partial \theta}{\partial \tau} \quad (8.8)$$

We need to remove the time dependent terms from equation (8.8) and cast the spatial derivatives in finite difference form. The species mass balance, equation (7.1), is used to replace the derivative of ω_i with respect to τ in equation (8.8). The species mass balance in transformed coordinates, with dimensionless variables is:

$$\begin{aligned} \frac{\partial \omega_i}{\partial \tau} = & \frac{(xu_I^* - v^*)}{a^*} \frac{\partial \omega_i}{\partial x} \\ & + \frac{1}{a^{*2}} \left[\left(\frac{D_i^{liq*}}{\rho} \frac{\partial \rho}{\partial x} + \frac{\partial D_i^{liq*}}{\partial x} \right) \frac{\partial \omega_i}{\partial x} + \frac{D_i^{liq*}}{x^2} \frac{\partial}{\partial x} \left(x^2 \frac{\partial \omega_i}{\partial x} \right) \right] \end{aligned} \quad (8.9)$$

To solve for the interface velocity, u_I^* and, v^* separate these terms and use variables to represent the non-velocity terms for ease in programming.

$$C = \frac{1}{a^{*2}} \left[\left(\frac{D_i^{liq*}}{\rho} \frac{\partial \rho}{\partial x} + \frac{\partial D_i^{liq*}}{\partial x} \right) \frac{\partial \omega_i}{\partial x} + \frac{D_i^{liq*}}{x^2} \frac{\partial}{\partial x} \left(x^2 \frac{\partial \omega_i}{\partial x} \right) \right] \quad (8.10)$$

$$A = \frac{x_j^*}{a^*} \frac{\partial \omega_i}{\partial x} \quad (8.11)$$

$$B = \frac{1}{a^*} \frac{\partial \omega_i}{\partial x} \quad (8.12)$$

Equation (8.9) becomes:

$$\frac{\partial \omega_i}{\partial \tau} = Au_I^* - Bv_j^* + C \quad (8.13)$$

The energy balance, equation (7.12), in transformed coordinates, with dimensionless variables is:

$$\begin{aligned} \frac{\partial \theta}{\partial \tau} = 3 \left[\frac{k_g}{D_{1,0}^{gas} \rho_{l,avg} c_{p,l,avg} a^{*2}} \frac{\partial \theta}{\partial y} \right]_{y=1}^{gas} \\ - \frac{\rho_{g,y=1} H_g}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} (v_g^* - u_I^*) + \frac{\rho_{l,x=N1} \Delta H_{vap} u_I^*}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} \end{aligned} \quad (8.14)$$

Separating out the velocity terms and assigning variable names gives,

$$D = \frac{3(\rho_{g,y=1} H_g + \rho_{l,x=N1} \Delta H_{vap})}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} \quad (8.15)$$

$$E = \frac{3\rho_{g,y=1} H_g}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} \quad (8.16)$$

$$F = \frac{3k_g}{D_{1,0}^{gas} \rho_{l,avg} c_{p,l,avg} a^{*2}} \frac{\partial \theta}{\partial y} \bigg|_{y=1}^{gas} \quad (8.17)$$

Equation 8.14 becomes,

$$\frac{\partial \theta}{\partial \tau} = Du_I^* - Ev_{g,y=1}^* - F \quad (8.18)$$

Substituting equations (8.13) and (8.18) into equation (8.8) gives,

$$\begin{aligned} \frac{xu_I^*}{a^*} \frac{\partial \rho}{\partial x} - \frac{1}{x^2 a^*} \frac{\partial}{\partial x} (x^2 \rho v^*) = \\ \frac{\partial \rho_{liq}}{\partial \omega_i} (Au_I^* - Bv_j^* + C) + \frac{\partial \rho_{liq}}{\partial \theta} (Du_I^* - Ev_{g,y=1}^* - F) \end{aligned} \quad (8.19)$$

On the left-hand-side of equation (8.19) let,

$$G = \frac{x}{a^*} \frac{\partial \rho}{\partial x} \quad (8.20)$$

In order to separate out the velocity terms in the second term on the left-hand-side of equation (8.19), the derivative must be broken into finite difference terms:

$$\left. \frac{\partial}{\partial x} (x^2 \rho v) \right|_j = 2x(\rho v^*) \Big|_j + x^2 \left. \frac{\partial (\rho v^*)}{\partial x} \right|_j \quad (8.21)$$

The derivative in the last term is approximated using central differencing:

$$\left. \frac{\partial (\rho v^*)}{\partial x} \right|_j = \frac{1}{\Delta x_{j-1} + \Delta x_j} \left[\frac{\Delta x_j}{\Delta x_{j-1}} (\rho_j v_j^* - \rho_{j-1} v_{j-1}^*) \right]$$

$$\left. + \frac{\Delta x_j}{\Delta x_{j-1}} (\rho_j v_j^* - \rho_{j-1} v_{j-1}^*) \right] \quad (8.22)$$

Assigning variable names equation (8.21) becomes:

$$H = \frac{x_j^2 \Delta x_j \rho_{j-1}}{\Delta x_{j-1} (\Delta x_{j-1} + \Delta x_j)} \quad (8.23)$$

$$I = 2x_j \rho_j + \frac{x_j^2 \Delta x_j \rho_j}{\Delta x_{j-1} (\Delta x_{j-1} + \Delta x_j)} - \frac{x_j^2 \Delta x_{j-1} \rho_j}{\Delta x_j (\Delta x_{j-1} + \Delta x_j)} \quad (8.24)$$

$$J = \frac{x_j^2 \Delta x_{j-1} \rho_{j+1}}{\Delta x_j (\Delta x_{j-1} + \Delta x_j)} \quad (8.25)$$

$$\left. \frac{\partial}{\partial x} (x^2 \rho v) \right|_j = -H v_{j-1}^* + I v_j^* + J v_{j+1}^* \quad (8.26)$$

Equation (8.19) becomes:

$$\begin{aligned} Gu_i^* - \frac{1}{x^2 a^*} (H v_{j-1}^* + I v_j^* + J v_{j+1}^*) = \\ \frac{\partial \rho_{liq}}{\partial \omega_i} (A u_i^* - B v_j^* + C) + \frac{\partial \rho_{liq}}{\partial \theta} (D u_i^* - E v_{g,y=1}^* - F) \end{aligned} \quad (8.27)$$

Collecting terms gives the equation to be solved for the interior nodes of the liquid droplet:

$$\begin{aligned}
& \left(\frac{H}{x_j^2 a^*} \right) v_{j-1}^* + \left(\frac{-I}{x_j^2 a^*} + \frac{\partial \rho_{liq}}{\partial \omega_i} B \right) v_j^* + \left(\frac{-J}{x_j^2 a^*} \right) v_{j+1}^* \\
& + \left(G - \frac{\partial \rho_{liq}}{\partial \omega_i} A - \frac{\partial \rho_{liq}}{\partial \theta} D \right) u_i^* + \left(\frac{\partial \rho_{liq}}{\partial \theta} E \right) v_{gas,y=1}^* \\
& = \left(\frac{\partial \rho_{liq}}{\partial \omega_i} C - \frac{\partial \rho_{liq}}{\partial \theta} F \right)
\end{aligned} \tag{8.28}$$

For the interior nodes of the droplet all derivatives are discretized using a central differencing finite scheme for a non-uniform grid. The first derivative at point x_j expressed in $O(x^2)$ accuracy is:

$$\left. \frac{\partial(\omega)}{\partial x} \right|_j = \frac{1}{\Delta x_{j-1} + \Delta x_j} \left[\frac{\Delta x_j}{\Delta x_{j-1}} (\omega_j - \omega_{j-1}) + \frac{\Delta x_{j-1}}{\Delta x_j} (\omega_{j+1} - \omega_j) \right] \tag{8.29}$$

This is the derivative for ω but the same differencing scheme works for any choice of dependent variable (temperature, density, diffusivity, etc.) (Foss, 1999). The Laplacian inside the droplet, used for equation (8.12) is:

$$\left. \frac{1}{x^2} \frac{\partial}{\partial x} \left(x^2 \frac{\partial \omega}{\partial x} \right) \right|_j = \frac{\partial^2 \omega}{\partial x^2} + \frac{2}{x} \frac{\partial \omega}{\partial x} \Big|_j \tag{8.30}$$

8.2. Liquid Phase - Droplet Surface

The droplet surface is the boundary between the liquid and gas phases. When evaluating equation (8.19) at this boundary a three-point backward differencing scheme is used. The velocity term on the left-hand-side of equation (8.19) is separated using a finite difference scheme to approximate the derivative:

$$\left. \frac{\partial}{\partial x} (x^2 \rho v) \right|_{j=NI} = 2x(\rho v) \Big|_{j=NI} + x^2 \left. \frac{\partial(\rho v)}{\partial x} \right|_{j=NI} \quad (8.31)$$

where $j = NI$ represents the last node of the liquid phase. At the droplet boundary the derivative in the last term of equation (8.31) is approximated using a three-point backward differencing scheme for a non-uniform grid:

$$\begin{aligned} \left. \frac{\partial(\rho v)}{\partial x} \right|_{j=NI} &= \frac{\Delta x_{j-1}}{\Delta x_{j-2} (\Delta x_{j-1} + \Delta x_{j-2})} \rho_{j-2} v_{j-2}^* \\ &\quad - \frac{(\Delta x_{j-1} + \Delta x_{j-2})}{\Delta x_{j-1} \Delta x_{j-2}} \rho_{j-1} v_{j-1}^* + \frac{\left[(\Delta x_{j-1} + \Delta x_{j-2})^2 - \Delta x_{j-1}^2 \right]}{\Delta x_{j-1} \Delta x_{j-2} (\Delta x_{j-1} + \Delta x_{j-2})} \rho_j v_j^* \end{aligned} \quad (8.32)$$

Assigning variables equation (8.31) becomes:

$$K = \frac{x_j^2 \Delta x_{j-1} \rho_{j-2}}{\Delta x_{j-2} (\Delta x_{j-1} + \Delta x_{j-2})} \quad (8.33)$$

$$L = \frac{x_j^2 (\Delta x_{j-1} + \Delta x_{j-2}) \rho_{j-1}}{\Delta x_{j-1} \Delta x_{j-2}} \quad (8.34)$$

$$M = 2x_j \rho_j + \frac{x_j^2 \left[(\Delta x_{j-1} + \Delta x_{j-2})^2 - \Delta x_{j-1}^2 \right] \rho_j}{\Delta x_{j-1} \Delta x_{j-2} (\Delta x_{j-1} + \Delta x_{j-2})} \quad (8.35)$$

$$\left. \frac{\partial}{\partial x} (x^2 \rho v) \right|_{j=N1} = K v_{j-2}^* - L v_{j-1}^* + M v_j^* \quad (8.36)$$

The combined equations for the liquid phase boundary, $j = N1$:

$$\begin{aligned} Gu_i^* - \frac{1}{x_j^2 a^*} (K v_{j-2}^* - L v_{j-1}^* + M v_j^*) = \\ \frac{\partial \rho_{liq}}{\partial \omega_i} (A u_i^* - B v_j^* + C) + \frac{\partial \rho_{liq}}{\partial \theta} (D u_i^* - E v_{g,y=1}^* - F) \end{aligned} \quad (8.37)$$

Collecting terms gives the equation to be solved for the boundary of the liquid droplet:

$$\begin{aligned} \left(\frac{K}{x_j^2 a^*} \right) v_{j-2}^* + \left(\frac{L}{x_j^2 a^*} \right) v_{j-1}^* + \left(-\frac{M}{x_j^2 a^*} + \frac{\partial \rho_{liq}}{\partial \omega_i} B \right) v_j^* \\ + \left(G - \frac{\partial \rho_{liq}}{\partial \omega_i} A - \frac{\partial \rho_{liq}}{\partial \theta} D \right) u_i^* + \left(\frac{\partial \rho_{liq}}{\partial \theta} E \right) v_{g,y=1}^* \\ = \left[\frac{\partial \rho_{liq}}{\partial \omega_i} C + \frac{\partial \rho_{liq}}{\partial \theta} F \right] \end{aligned} \quad (8.38)$$

The derivatives at the liquid phase boundary are discretized using the following three-point backward differencing scheme:

$$\begin{aligned} \left. \frac{\partial \omega}{\partial x} \right|_{j=N1} = \frac{\Delta x_{j-1}}{\Delta x_{j-2} (\Delta x_{j-1} + \Delta x_{j-2})} \omega_{j-2} \\ - \frac{(\Delta x_{j-1} + \Delta x_{j-2})}{\Delta x_{j-1} \Delta x_{j-2}} \omega_{j-1} + \frac{\left[(\Delta x_{j-1} + \Delta x_{j-2})^2 - \Delta x_{j-1}^2 \right]}{\Delta x_{j-1} \Delta x_{j-2} (\Delta x_{j-1} + \Delta x_{j-2})} \omega_j \end{aligned} \quad (8.39)$$

The same differencing scheme is used for all of the dependent variables.

8.3. Liquid-Gas Boundary

A jump condition is employed at the gas-liquid boundary. Equation (7.10) is used to relate the velocities in the liquid phase and in the gas phase at the droplet surface. The jump condition is,

$$\rho_{liq} \left(v_{liq} - u_I \right) \Big|_{r=a} = \rho_{gas} \left(v_{gas} - u_I \right) \Big|_{r=a} \quad (7.10)$$

Collecting velocity terms and non-dimensionalizing gives,

$$\left(v^* \rho \right) \Big|_{j=N1}^{liq} - \left(v^* \rho \right) \Big|_{j=1}^{gas} + u_I^* \left(\rho_{j=1}^{gas} - \rho_{j=N1}^{liq} \right) = 0 \quad (8.40)$$

8.4. Gas Phase - Droplet Surface

The coordinate transformation and the use of dimensionless variables in the gas phase recast the equation of continuity as:

$$\frac{\partial \rho_{gas}}{\partial \tau} = \frac{-y u_I^*}{a^*} \frac{\partial \rho}{\partial y} + \frac{y^4}{a^*} \frac{\partial}{\partial y} \left(\frac{\rho v^*}{y^2} \right) \quad (8.41)$$

An equation of state is used to relate the total gas density to the gas weight fraction and temperature:

$$\frac{\partial \rho_{gas}}{\partial \tau} = \frac{\partial \rho_{gas}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \tau} + \frac{\partial \rho_{gas}}{\partial \theta} \frac{\partial \theta}{\partial \tau} \quad (8.42)$$

See equations (8.2) through (8.7), the procedure is the same used in the liquid phase. Combining equations (8.41) and (8.42):

$$\frac{-yu_I^*}{a^*} \frac{\partial \rho}{\partial y} + \frac{y^4}{a^*} \frac{\partial}{\partial y} \left(\frac{\rho v^*}{y^2} \right) = \frac{\partial \rho_{gas}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \tau} + \frac{\partial \rho_{gas}}{\partial \theta} \frac{\partial \theta}{\partial \tau} \quad (8.43)$$

As in the liquid phase, the species mass balance and the energy balance supply the time dependency terms. The species mass balance in the y coordinate system is:

$$\begin{aligned} \frac{\partial \omega_i}{\partial \tau} = & \frac{y(yv^* - u_I^*)}{a^*} \frac{\partial \omega_i}{\partial y} \\ & + \frac{y^4}{a^{*2}} \left[\left(\frac{D_i^{gas*}}{\rho} \frac{\partial \rho}{\partial y} + \frac{\partial D_i^{gas*}}{\partial y} \right) \frac{\partial \omega_i}{\partial y} + D_i^{gas*} \frac{\partial^2 \omega_i}{\partial y^2} \right] \end{aligned} \quad (8.44)$$

Separating velocity terms and assigning variables, equation (8.44) becomes:

$$N = -\frac{y_j}{a^*} \frac{\partial \omega_i}{\partial y} \quad (8.45)$$

$$O = \frac{y_j^2}{a^*} \frac{\partial \omega_i}{\partial y} \quad (8.46)$$

$$P = \frac{y^4}{a^{*2}} \left[\left(\frac{D_i^{gas*}}{\rho} \frac{\partial \rho}{\partial y} + \frac{\partial D_i^{gas*}}{\partial y} \right) \frac{\partial \omega_i}{\partial y} + D_i^{gas*} \frac{\partial^2 \omega_i}{\partial y^2} \right] \quad (8.47)$$

Equation (8.44) becomes:

$$\frac{\partial \omega_i}{\partial \tau} = Nu_I^* + Ov_j^* + P \quad (8.48)$$

The energy balance, equation (7.12), at the droplet surface in the gas phase, is the same as used in the liquid phase. In the transformed y -coordinate system:

$$\begin{aligned} \frac{\partial \theta}{\partial \tau} = 3 \left[\frac{k_g}{D_{1,0}^{gas} \rho_{l,avg} c_{p,l,avg} a^{*2}} \frac{\partial \theta}{\partial y} \right]_{y=1}^{gas} \\ - \frac{\rho_{g,y=1} H_g}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} (\nu_g^* - u_I^*) + \frac{\rho_{l,x=N1} \Delta H_{vap} u_I^*}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} \end{aligned} \quad (8.14)$$

Since this is the same equation used in the liquid phase, the variables are kept the same:

$$D = \frac{3(\rho_{g,y=1} H_g + \rho_{l,x=N1} \Delta H_{vap})}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} \quad (8.15)$$

$$E = \frac{3\rho_{g,y=1} H_g}{T_{ref} \rho_{l,avg} c_{p,l,avg} a^*} \quad (8.16)$$

$$F = \frac{3k_g}{D_{1,0}^{gas} \rho_{l,avg} c_{p,l,avg} a^{*2}} \frac{\partial \theta}{\partial y} \bigg|_{y=1}^{gas} \quad (8.17)$$

Equation (8.14) is the same as in the gas phase:

$$\frac{\partial \theta}{\partial \tau} = Du_I^* - E\nu_{g,y=1}^* - F \quad (8.18)$$

Substituting equations (8.48) and (8.18) into equation (8.43) gives:

$$\begin{aligned} \frac{-yu_l^*}{a^*} \frac{\partial \rho}{\partial y} + \frac{y^4}{a^*} \frac{\partial}{\partial y} \left(\frac{\rho v^*}{y^2} \right) = \\ \frac{\partial \rho_{gas}}{\partial \omega_i} (Nu_l^* + Ov_j^* + P) + \frac{\partial \rho_{gas}}{\partial \theta} (Du_l^* - Ev_{g,y=1}^* - F) \end{aligned} \quad (8.49)$$

On the left-hand-side of equation (8.49) let:

$$T = \frac{y}{a^*} \frac{\partial \rho}{\partial y} \quad (8.50)$$

To separate out the velocity term in the derivative on the left-hand-side of equation (8.49), the term is separated using forward differencing:

$$\begin{aligned} \left. \frac{\partial}{\partial y} \left(\frac{\rho_j v_j^*}{y_j^2} \right) \right|_{j=1} = \frac{1}{y_j^2} \left[\frac{\Delta y_j^2 + (\Delta y_j + \Delta y_{j+1})^2}{\Delta y_j \Delta y_{j+1} (\Delta y_j + \Delta y_{j+1})} \rho_j v_j^* \right. \\ \left. + \frac{(\Delta y_j + \Delta y_{j+1})}{\Delta y_j \Delta y_{j+1}} \rho_{j+1} v_{j+1}^* - \frac{\Delta y_j}{\Delta y_{j+1} (\Delta y_j + \Delta y_{j+1})} \rho_{j+2} v_{j+2}^* \right] - \frac{2}{y_j^3} \rho_j v_j^* \end{aligned} \quad (8.51)$$

Assigning variables to equation (8.51):

$$U = \frac{\Delta y_j^2 + (\Delta y_j + \Delta y_{j+1})^2}{y_j^2 \Delta y_j \Delta y_{j+1} (\Delta y_j + \Delta y_{j+1})} \rho_j - \frac{2}{y_j^3} \rho_j \quad (8.52)$$

$$V = \frac{(\Delta y_j + \Delta y_{j+1})}{y_j^2 \Delta y_j \Delta y_{j+1}} \rho_{j+1} \quad (8.53)$$

$$W = \frac{\Delta y_j}{y_j^2 \Delta y_{j+1} (\Delta y_j + \Delta y_{j+1})} \rho_{j+2} \quad (8.54)$$

$$\left. \frac{\partial}{\partial y} \left(\frac{\rho_j v_j^*}{y_j^2} \right) \right|_{j=1} = U v_j^* + V v_{j+1}^* + W v_{j+2}^* \quad (8.55)$$

Equation (8.49) becomes:

$$\begin{aligned} Tu_l^* &= \frac{y_j^4}{a^*} (U v_j^* + V v_{j+1}^* + W v_{j+2}^*) = \\ &= \frac{\partial \rho_{gas}}{\partial \omega_i} (N u_l^* + O v_j^* + P) + \frac{\partial \rho_{gas}}{\partial \theta} (D u_l^* - E v_{g,y=1}^* - F) \end{aligned} \quad (8.56)$$

Collecting terms gives the equation to be solved for the first point in the gas phase:

$$\begin{aligned} &\left(\frac{y_j^4}{a^*} U - \frac{\partial \rho_{gas}}{\partial \omega_i} O + \frac{\partial \rho_{gas}}{\partial \theta} E \right) v_j^* + \left(\frac{y_j^4}{a^*} V \right) v_{j+1}^* + \left(\frac{y_j^4}{a^*} W \right) v_{j+2}^* \\ &+ \left(T - \frac{\partial \rho_{gas}}{\partial \omega_i} N - \frac{\partial \rho_{gas}}{\partial \theta} D \right) u_l^* = \left(\frac{\partial \rho_{gas}}{\partial \omega_i} P - \frac{\partial \rho_{gas}}{\partial \theta} F \right) \end{aligned} \quad (8.57)$$

The derivatives at the first point in the gas phase boundary are discretized using the following three-point forward differencing scheme:

$$\begin{aligned} \left. \frac{\partial \omega}{\partial y} \right|_{j=1} &= \frac{\Delta y_j^2 + (\Delta y_j + \Delta y_{j+1})^2}{\Delta y_j \Delta y_{j+1} (\Delta y_j + \Delta y_{j+1})} \omega_j \\ &+ \frac{(\Delta y_j + \Delta y_{j+1})}{\Delta y_j \Delta y_{j+1}} \omega_{j+1} - \frac{\Delta y_j}{\Delta y_{j+1} (\Delta y_j + \Delta y_{j+1})} \omega_{j+2} \end{aligned} \quad (8.58)$$

The same differencing scheme is used for all of the dependent variables.

8.5. Gas Phase - Interior Nodes

In the gas phase interior points, the continuity equation is the same as used in the gas phase liquid-gas interface:

$$\frac{\partial \rho_{gas}}{\partial \tau} = \frac{-yu_l^*}{a^*} \frac{\partial \rho}{\partial y} + \frac{y^4}{a^*} \frac{\partial}{\partial y} \left(\frac{\rho v^*}{y^2} \right) \quad (8.41)$$

The derivative on the right-hand-side containing the velocity terms is approximated with the central differencing scheme used in the liquid phase interior points:

$$\begin{aligned} \frac{\partial}{\partial y} \left(\frac{\rho v}{y^2} \right) = \frac{1}{y^2} \left\{ \frac{1}{\Delta y_{j-1} + \Delta y_j} \left[\frac{\Delta y_j}{\Delta y_{j-1}} \right. \right. \\ \left. \left. \left(\rho_j v_j^* - \rho_{j-1} v_{j-1}^* \right) + \frac{\Delta y_{j-1}}{\Delta y_j} \left(\rho_{j+1} v_{j+1}^* - \rho_j v_j^* \right) \right] \right\} - \frac{2}{y_j^3} \rho_j v_j^* \end{aligned} \quad (8.59)$$

Assigning variables to equation (8.42):

$$X = \frac{-\Delta y_j}{y_j^2 \Delta y_{j-1} (\Delta y_{j-1} + \Delta y_j)} \rho_{j-1} \quad (8.60)$$

$$Y = \frac{\Delta y_j}{y_j^2 \Delta y_{j-1} (\Delta y_{j-1} + \Delta y_j)} \rho_j - \frac{\Delta y_{j-1}}{y_j^2 \Delta y_j (\Delta y_{j-1} + \Delta y_j)} - \frac{2\rho_j}{y_j^3} \quad (8.61)$$

$$Z = \frac{-\Delta y_{j-1}}{y_j^2 \Delta y_j (\Delta y_{j-1} + \Delta y_j)} \rho_{j+1} \quad (8.62)$$

$$\frac{\partial}{\partial y} \left(\frac{\rho v^*}{y^2} \right) = X v_{j-1}^* + Y v_j^* + Z v_{j+1}^* \quad (8.63)$$

Equation (8.41) becomes:

$$\frac{\partial \rho_{gas}}{\partial \tau} = T u_I^* + \frac{y_j^4}{a^*} (X v_{j-1}^* + Y v_j^* + Z v_{j+1}^*) \quad (8.64)$$

The equation of state is used to relate the total gas density to the gas weight fraction and temperature:

$$\frac{\partial \rho_{gas}}{\partial \tau} = \frac{\partial \rho_{gas}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \tau} + \frac{\partial \rho_{gas}}{\partial \theta} \frac{\partial \theta}{\partial \tau} \quad (8.42)$$

For the gas phase interior points, the derivative of θ with respect to τ comes from the energy balance, equation (7.11), in transformed coordinates with dimensionless variables:

$$\begin{aligned} \frac{\partial \theta}{\partial \tau} = & \frac{-y(u_I^* - y v^*)}{a^*} \frac{\partial \theta}{\partial y} + \frac{y^4}{\rho C_{p,l,avg} D_{1,0}^{gas} a^{*2}} \frac{\partial}{\partial y} \left(k \frac{\partial \theta}{\partial y} \right) \\ & + \frac{y^4}{\rho C_{p,l,avg} T_{ref} a^{*2}} \frac{\partial}{\partial y} \left(\rho \sum_{i=1}^{nl2} H_i D_i^* \frac{\partial \omega_i}{\partial y} \right) \end{aligned} \quad (8.65)$$

Assigning variables to equation (8.65):

$$AA = \frac{-y_j}{a^*} \frac{\partial \theta}{\partial y} \quad (8.66)$$

$$BB = \frac{y_j^2}{a^*} \frac{\partial \theta}{\partial y} \quad (8.67)$$

$$CC = \frac{y^4}{\rho C_{p,l,avg} T_{ref} a^{*2}} \frac{\partial}{\partial y} \left(\rho \sum_{i=1}^{nl2} H_i D_i^* \frac{\partial \omega_i}{\partial y} \right)$$

$$+ \frac{y^4}{\rho C_{p,l,avg} D_{1,0}^{gas} a^{*2}} \frac{\partial}{\partial y} \left(k \frac{\partial \theta}{\partial y} \right) \quad (8.68)$$

The transformed energy balance, equation (8.65) becomes:

$$\frac{\partial \theta}{\partial \tau} = AAu_I^* + BBv_j^* + CC \quad (8.69)$$

The species mass balance comes from the first point in the gas phase equation (8.48):

$$\frac{\partial \omega_i}{\partial \tau} = Nu_I^* + Ov_j^* + P \quad (8.48)$$

Substituting equations (8.64) and (8.69) into equation (8.42):

$$\begin{aligned} \frac{\partial \rho_{gas}}{\partial \tau} &= \frac{\partial \rho_{gas}}{\partial \omega_i} (Nu_I^* + Ov_j^* + P) \\ &\quad + \frac{\partial \rho_{gas}}{\partial \theta} (AAu_I^* + BBv_j^* + CC) \end{aligned} \quad (8.70)$$

Combining equations (8.64) and (8.70) and collecting the velocity terms gives the equation to solve for the gas phase interior points:

$$\begin{aligned} \left(\frac{y_j^4}{a^*} X \right) v_{j-1}^* + \left(\frac{y_j^4}{a^*} Y - \frac{\partial \rho}{\partial \omega_i} O + \frac{\partial \rho}{\partial \theta} BB \right) v_j^* + \left(\frac{y_j^4}{a^*} Z \right) v_{j+1}^* \\ + \left(T - \frac{\partial \rho}{\partial \omega_i} N - \frac{\partial \rho}{\partial \theta} AA \right) u_I^* = \frac{\partial \rho}{\partial \omega_i} P + \frac{\partial \rho}{\partial \theta} CC \end{aligned} \quad (8.71)$$

8.6. Matrix Structure

The coefficient matrix $A(m,n)$ is shown below. This is a 13×13 matrix for a model with 7 nodes in the liquid phase and 7 nodes in the gas phase. The first node in the liquid phase and the last node in the gas phase are known from the boundary conditions. There is an additional node at the liquid-gas interface so that the matrix dimensions are given by:

$$m = n = n1 + n2 - 1 \quad (8.72)$$

j represents the model node.

n =	1	2	3	4	5	6	7	8	9	10	11	12	13
m, j =	2	3	4	5	6	7	interface	1	2	3	4	5	6
1 2	C_{11}	C_{12}	0	0	0	0	u_l	u_g	0	0	0	0	0
2 3	C_{21}	C_{22}	C_{23}	0	0	0	u_l	u_g	0	0	0	0	0
3 4	0	C_{32}	C_{33}	C_{34}	0	0	u_l	u_g	0	0	0	0	0
4 5	0	0	C_{43}	C_{44}	C_{45}	0	u_l	u_g	0	0	0	0	0
5 6	0	0	0	C_{54}	C_{55}	C_{56}	u_l	u_g	0	0	0	0	0
6 7	0	0	0	C_{64}	C_{65}	C_{66}	u_l	u_g	0	0	0	0	0
7 int.	0	0	0	0	0	C_{76}	C_{77}	C_{78}	0	0	0	0	0
8 1	0	0	0	0	0	0	u_l	C_{88}	C_{89}	C_{810}	0	0	0
9 2	0	0	0	0	0	0	u_l	C_{98}	C_{99}	C_{910}	0	0	0
10 3	0	0	0	0	0	0	u_l	0	C_{109}	C_{1010}	C_{1011}	0	0
11 4	0	0	0	0	0	0	u_l	0	0	C_{1110}	C_{1111}	C_{1112}	0
12 5	0	0	0	0	0	0	u_l	0	0	0	C_{1211}	C_{1212}	C_{1213}
13 6	0	0	0	0	0	0	u_l	0	0	0	0	C_{1312}	C_{1313}

Figure 5 - Coefficient Matrix A showing tri-diagonal structure

Figures 5 and 6 show the symmetry of the matrix.

n =	1	2	3	4	5	6	7	8	9	10	11	12	13
m, j =	2	3	4	5	6	7	interface	1	2	3	4	5	6
1 2	C_{11}	C_{12}	0	0	0	0	u_l	u_g	0	0	0	0	0
4 3	C_{21}	C_{22}	C_{23}	0	0	0	u_l	u_g	0	0	0	0	0
5 4	0	C_{32}	C_{33}	C_{34}	0	0	u_l	u_g	0	0	0	0	0
4 5	0	0	C_{43}	C_{44}	C_{45}	0	u_l	u_g	0	0	0	0	0
5 6	0	0	0	C_{54}	C_{55}	C_{56}	u_l	u_g	0	0	0	0	0
6 7	0	0	0	C_{64}	C_{65}	C_{66}	u_l	u_g	0	0	0	0	0
7 int.	0	0	0	0	0	C_{76}	C_{77}	C_{78}	0	0	0	0	0
8 1	0	0	0	0	0	0	u_l	C_{88}	C_{89}	C_{810}	0	0	0
9 2	0	0	0	0	0	0	u_l	C_{98}	C_{99}	C_{910}	0	0	0
10 3	0	0	0	0	0	0	u_l	0	C_{109}	C_{1010}	C_{1011}	0	0
11 4	0	0	0	0	0	0	u_l	0	0	C_{1110}	C_{1111}	C_{1112}	0
12 5	0	0	0	0	0	0	u_l	0	0	0	C_{1211}	C_{1212}	C_{1213}
13 6	0	0	0	0	0	0	u_l	0	0	0	0	C_{1312}	C_{1313}

Figure 6 - Coefficient Matrix A showing vertical and horizontal symmetry

The liquid phase terms are located in the upper left region of the matrix. Column $n=6$, $j=7$ is the edge of the liquid phase. Column $n=7$ is the coefficient for the liquid-gas interface velocity. Column $n=8$, $j=1$ in the gas phase is the gas phase velocity coefficient at the first node in the gas phase. This term comes from the energy balance about the sphere, equation (7.12). Row $m=7$ is the jump-condition coefficients from equation (8.40). The lower right quadrant of the coefficient matrix holds the gas phase terms. Figure 7 shows the coefficient matrix A with the velocity vector we are solving for and the right-hand-side vector **b**

$$\begin{bmatrix}
C_{11} & C_{12} & 0 & 0 & 0 & 0 & u & v_g & 0 & 0 & 0 & 0 & 0 \\
C_{21} & C_{22} & C_{23} & 0 & 0 & 0 & u & v_g & 0 & 0 & 0 & 0 & 0 \\
0 & C_{32} & C_{33} & C_{34} & 0 & 0 & u & v_g & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & C_{43} & C_{44} & C_{45} & 0 & u & v_g & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & C_{54} & C_{55} & C_{56} & u & v_g & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & C_{64} & C_{65} & C_{66} & u & v_g & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & C_{76} & C_{77} & C_{78} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & u & C_{88} & C_{89} & C_{810} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & u & C_{98} & C_{99} & C_{910} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & u & 0 & C_{109} & C_{1010} & C_{1011} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & u & 0 & 0 & C_{1110} & C_{1111} & C_{1112} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & u & 0 & 0 & 0 & C_{1211} & C_{1212} & C_{1213} \\
0 & 0 & 0 & 0 & 0 & 0 & u & 0 & 0 & 0 & 0 & C_{1312} & C_{1313}
\end{bmatrix}
\begin{bmatrix}
v_2 \\
v_3 \\
v_4 \\
v_5 \\
v_6 \\
v_7 \\
u \\
v_1 \\
v_2 \\
v_3 \\
v_4 \\
v_5 \\
v_6
\end{bmatrix}
=
\begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
b_4 \\
b_5 \\
b_6 \\
b_7 \\
b_8 \\
b_9 \\
b_{10} \\
b_{11} \\
b_{12} \\
b_{13}
\end{bmatrix}$$

Figure 7 - The coefficient matrix **A** with the velocity vector **v** and the right-hand-side vector **b**.

The FORTRAN subroutine *Velocity* was written to calculate the coefficient terms of matrix **A** and the right-hand-side vector, **b**. The call to the subroutine is:

```

SUBROUTINE VELOCITY(W,T,ASTAR,DENS,CP,COND,
DIFF,ENTH,DENSLAVG,CPLAVG,UI,V)

```

All of the variables are passed to the subroutine from the previous time-step, and a new velocity vector, **V**, is calculated. Subroutine VELOCITY is located in the appendix.

9. Solving the Linear Set of Equations

The equations derived in the previous section form a set of linear, algebraic equations. The equations are arranged in matrix form:

$$A \cdot v = b \quad (9.1)$$

where A is the coefficient matrix, and b is the right-hand-side vector matrix that were developed in the previous section. The vector matrix v is the bulk velocity vector matrix we are solving for. In general terms A and b are as shown:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix} \quad (9.2)$$

There are n unknowns, $v_1, v_2, v_3, \dots, v_n$ and m equations, the elements of the coefficient matrix A and the vector matrix b are known. We solve for the unknown bulk velocity vector by solving for v in equation (9.1):

$$v = A^{-1} \cdot b \quad (9.3)$$

We have a large number of linear equation solvers to choose from. The current version of MULTEVAP uses *LU Decomposition* to solve equation (9.3).

9.1. LU Decomposition

LU Decomposition, Lower-Upper Decomposition, solves the set of equations by breaking them up into two successive linear sets (Press, Teukolsky, Vetterling, & Flannery, 1992). The coefficient matrix A is decomposed into two triangular matrices, the lower matrix L , containing matrix elements lying on the diagonal and below, and the upper matrix U , containing elements on the diagonal and above. The decomposition is shown in equations (9.4) and (9.5) for a sample 4×4 matrix.

$$L \cdot U = A \quad (9.4)$$

$$\begin{bmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix} \cdot \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ 0 & \beta_{22} & \beta_{23} & \beta_{24} \\ 0 & 0 & \beta_{33} & \beta_{34} \\ 0 & 0 & 0 & \beta_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (9.5)$$

Decomposing **A** into two triangular matrices allows us to use forward and backward substitution to make solving the set of equations “trivial”(Press, et al., 1992).

Equation (9.1) is solved as follows:

$$A \cdot v = b \quad (9.1)$$

$$A \cdot v = (L \cdot U) \cdot v = L \cdot (U \cdot v) = b \quad (9.6)$$

First solve for the vector **y** using forward substitution:

$$U \cdot v = y \quad (9.7)$$

Substituting equation (9.7) into equation (9.6) gives:

$$L \cdot y = b \quad (9.8)$$

In matrix form equation (9.8) is:

$$\begin{bmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (9.9)$$

Matrix multiplication gives,

$$\begin{aligned}
 \alpha_{11}y_1 + 0 + 0 + 0 &= b_1 \\
 \alpha_{21}y_1 + \alpha_{22}y_2 + 0 + 0 &= b_2 \\
 \alpha_{31}y_1 + \alpha_{32}y_2 + \alpha_{33}y_3 + 0 &= b_3 \\
 \alpha_{41}y_1 + \alpha_{42}y_2 + \alpha_{43}y_3 + \alpha_{44}y_4 &= b_4
 \end{aligned} \tag{9.10}$$

Solving for y gives,

$$\begin{aligned}
 y_1 &= \frac{b_1}{\alpha_{11}} \\
 y_2 &= \frac{1}{\alpha_{22}} \left[b_2 - (\alpha_{21}y_1) \right] \\
 y_3 &= \frac{1}{\alpha_{33}} \left[b_3 - (\alpha_{31}y_1 + \alpha_{32}y_2) \right] \\
 y_4 &= \frac{1}{\alpha_{44}} \left[b_4 - (\alpha_{41}y_1 + \alpha_{42}y_2 + \alpha_{43}y_3) \right]
 \end{aligned} \tag{9.11}$$

In a general format for any size matrix y_i is calculated using the following equation:

$$y_i = \frac{1}{\alpha_{ii}} \left[b_i - \sum_{j=1}^{i-1} \alpha_{ij}y_j \right] \quad \text{for } i = 1, 2, 3, \dots, N \tag{9.12}$$

Equation (9.7) is solved similarly using back substitution, in a general format the unknown vector matrix is solved for using,

$$v_i = \frac{1}{\beta_{ii}} \left[y_i - \sum_{j=i+1}^n \beta_{ij}v_j \right] \quad \text{for } i = N, N-1, \dots, 1 \tag{9.13}$$

9.2. Performing the LU Decomposition

To solve for the α 's and β 's of equation (9.5) we can start by writing out the equations for each term:

$$\begin{bmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix} \cdot \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ 0 & \beta_{22} & \beta_{23} & \beta_{24} \\ 0 & 0 & \beta_{33} & \beta_{34} \\ 0 & 0 & 0 & \beta_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (9.5)$$

Matrix multiplication gives N^2 equations for the $N^2 + N$ unknowns. The extra N unknowns arise from the diagonal being represented twice (Press et al., 1992). The required additional equations are specified arbitrarily and the remaining unknowns are solved for. We specify:

$$\alpha_{ii} \equiv 1 \quad \text{for } i = 1, 2, 3 \dots N \quad (9.14)$$

It turns out that the $N^2 + N$ set of equations can be solved for easily by arranging the equations in a certain order. This method is referred to as *Crout's Algorithm* (Press, et al., 1992). The order is as follows:

- 1) Set $\alpha_{ii} = 1$ for $i = 1, 2, 3 \dots N$.
- 2) For each $j = 1, 2, 3 \dots N$ do these two procedures:
 - a) For $i = 1, 2, 3 \dots j$ solve for β_{ij} using:

$$\beta_{ij} = a_{ij} - \sum_{k=1}^{i-1} \alpha_{ik} \beta_{kj} \quad (9.15)$$

- b) For $i = j+1, j+2 \dots N$ solve for α_{ij} using:

$$\alpha_{ij} = \frac{1}{\beta_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \alpha_{ik} \beta_{kj} \right) \quad (9.16)$$

Crout's Algorithm arranges the equations in such an order that the α 's and β 's that appear on the right hand side are already determined by the time they are needed. In addition, each a_{ij} is used only once allowing us to store the newly calculated α or β in its location, thereby destroying the original A matrix as we go. This is referred to as "in place decomposition" (Press et al., 1992). A is returned as:

$$\begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ \alpha_{21} & \beta_{22} & \beta_{23} & \beta_{24} \\ \alpha_{31} & \alpha_{32} & \beta_{33} & \beta_{34} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \beta_{44} \end{bmatrix} \quad (9.17)$$

The $\alpha_{ii} = 1$ for $i = 1 \dots N$ terms are not stored (Press, et al., 1992).

9.3. Pivoting

In order to keep the Crout's method stable against round-off error, pivoting is performed. Pivoting places a favorable element - usually the element with the largest magnitude, along the matrix diagonal. Selecting the largest element in the matrix as the pivot element, termed total pivoting, requires too much time and computer memory to be efficient; partial pivoting, where the largest element in a column is moved to the diagonal through row interchanges, is sufficient to maintain the stability of Crout's method and is more efficient to implement (Press, et al., 1992).

Crout's method with partial pivoting, takes advantage of the fact that equations (9.15) and (9.16) are identical for $i = j$ except for the division in the latter equation. The selection of a divisor, or pivot element, can be done after all the elements in the column have been calculated. Then the largest magnitude element is chosen as the pivot element and all the divisions are done at once (Press, et al., 1992). The routine used in MULTEVAP finds the largest element in each row and scales the row by that element while it is looking for the largest pivot element, this is referred to as *implicit pivoting*. Scaling each row by the largest element in that row removes exceptionally large or small numbers and improves the routine's stability against round-off error (MacDonald, 1998).

The partial pivoting decomposes matrix A into a row-wise permutation of A . The FORTRAN subroutine used to perform the decomposition is located in the Appendix, titled *SUBROUTINE ludcmp*. The call to the subroutine is:

SUBROUTINE ludcmp(a,n,np,indx,d)

The variables used in the call to the subroutine are described below:

a - the coefficient matrix $A(1:n, 1:n)$
n - the dimension of matrix A
np - the memory size allocated to matrix A
indx(1:n) - an output vector that records the row permutation effected by the partial pivoting
d - a flag output as ± 1 depending on whether the number of row interchanges was even or odd.

The subroutine *ludcmp* can be summarized as follows:

- 1) Find the largest magnitude element in each row. This will be the scaling factor.
- 2) Perform Crout's method. For each column j , up to row $i = j-1$:
 - a) Calculate β_{ij} for $i = 1 \dots j-1$ using equation (9.7). These are the elements above the diagonal.
 - b) Calculate β_{ij} for $i = j, j+1 \dots N$ using equation (9.7). These are the elements on and below the diagonal.
 - c) Multiply the terms by the scaling factor found in step 1.
 - d) Compare the β_{ij} terms on the diagonal and below. Is the scaled β_{ij} term greater than the previous term?
 - i) If so, set $imax = i$ to record the row number with the largest β_{ij} term.
 - ii) Set $aamax = \beta_{ij}$, the scaled β_{ij} term.
 - e) Is the current column equal to the row containing the largest magnitude β_{ij} , i.e. does the largest magnitude β_{ij} lie on the diagonal? If not then interchange the rows.
 - i) For all columns move the row with the largest element so that the large element lies on the diagonal.
 - (1) $dum = \text{largest magnitude element in column}$
 - (2) $\text{largest element} = \text{current element}$
 - (3) $\text{current element} = dum$ (the largest element)

-where *dum* is a dummy variable.
 - f) If there are elements below the diagonal that require division by the pivot element, i.e., the current column is not the last column. This step completes the calculation of equation (9.8).
 - i) $dum = 1/a(j,j)$ where $a(j,j)$ is the new pivot element
 - ii) $a(i,j) = a(i,j) dum$
 - iii) Return to step 2 and repeat this procedure for the next column.

The result of the above steps is the output matrix shown in equation (9.17) where the original A matrix has been replaced by the decomposed A matrix containing the α and β terms of the lower and upper matrices.

The next step is to perform the forward and backward substitutions. This is achieved with a call to the FORTRAN subroutine *lubksb*. This subroutine unscrambles the permutation from *ludcmp* and solves equations (9.12) and (9.13). The call to the subroutine is:

SUBROUTINE lubksb(a,n,np,indx,b)

The variables are described below:

a - the decomposed matrix $A(1:n, 1:n)$ from *ludcmp*
n - the dimension of matrix A
np - the memory size allocated to matrix A
indx(1:n) - the output vector from *ludcmp*
b - the right-hand-side vector from eq.(5.1)

The subroutine *lubksb* can be summarized as follows:

1. Loop through the rows, $i = 1, n$
 - a. Unscramble the permutation performed in *ludcmp* by moving the right-hand-side element to the proper row.
 - b. Perform forward substitution
2. Loop through the rows, $i = N, 1, -1$
 - a. Perform the backward substitution

Subroutine *lubksb* replaces the right-hand-side vector, b with the solution vector from v equation (5.1).

Execution of the decomposition in *ludcmp* requires approximately $N^3/3$ executions of the inner loops, each containing a multiply and a sum. This is the operation count for solving one or a few right-hand-sides and is a factor of three better than similar linear equation solvers (Press, et al., 1992).

ludcmp and *lubksb* are called from within the *Velocity* subroutine. The vector v from *lubksb* is the solution to the *Velocity* subroutine.

10. MULTEVAP Program Structure

The MULTEVAP Program reads the initial droplet and drying environment conditions from an input file. This information is then used to calculate the weight fractions and temperatures at each successive time step as the droplet evaporates.

The *Velocity* subroutine is called from within the *Derivs* subroutine; that uses the new velocity values to calculate the time derivative of the dependent variables. The time stepping routine used in MULTEVAP is an explicit Euler; this subroutine uses the time derivatives to calculate the new weight fraction and temperature values.

At each new time step, the velocity vector is found using the weight fractions and temperatures from the previous time step. The velocity vector is used to calculate the time derivative values of the dependent variables; and, time derivative values are used to calculate the new weight fractions and temperatures.

The MULTEVAP Program structure is listed:

PROGRAM MULTEVAP

- 1) Read input file
 - a) Number of species in the droplet, in the gas phase and in both phases
 - b) Initial conditions; droplet size, weight fractions, and temperatures.
 - c) Time duration
 - d) Number of nodes in the liquid phase and gas phase.
- 2) Initialize dependent variable array with initial value weight fractions and temperatures.
- 3) Initialize velocity array to zero.
- 4) Calculate bulk density at each node using weight fractions and temperatures.
- 5) Load density array with bulk density at each node in the liquid and gas.
- 6) CALL EULER
 - a) CALL DERIVS
 - i) CALL VELOCITY
 - (1) Get new velocity array using weight fractions and temperatures from the previous time step.
 - ii) Use new velocity values to calculate the time derivatives at each node.
 - b) Calculate time step.
 - c) Calculate new dependent variable values at new time step.
 - d) Advance to the next time point.
- 7) Update liquid and gas phase bulk properties based on new weight fractions and Temperatures.
- 8) Compare current time step with final time step, return to step 6 if the current time step is less than the final time step.

11. Results

A spatial resolution analysis was performed on the output from the Velocity subroutine. The number of nodes in the liquid and gas phase was increased until the velocity values ceased to show a change. Figure 8 shows a plot of the velocity at each node in the liquid phase for a node value of 11, 22, 44, 88, and 100 nodes.

Figure 8 – Liquid phase velocity for different numbers of nodes.

As the number of nodes increased, the change in the velocity value at the droplet surface, radial coordinate 1.00 in Figure 8, differed in value by a factor of 59%, 55%, 53%, and 12% respectively with the increasing number of nodes. More than 100 nodes could not be run on the current version of MULTEVAP due to the limitation of the dimensioned variables. However, the output from the Velocity subroutine is expected to converge at a little more than 100 nodes.

A pseudo-steady state analysis predicts that the Stefan flow velocity decreases as $1/r^2$.

$$v(r) = -u_i \left(\frac{\rho_{liq} - \rho_{gas}}{\rho_{gas}} \right) \left(\frac{a}{r} \right)^2 \quad (11.1)$$

A comparison of the gas phase velocity calculated with the pseudo-steady analysis and with MULTEVAP shows a discrepancy in the predicted velocity value. Figure 9 shows the gas phase velocity for a pure water droplet with an initial diameter of 10 microns. The gas phase conditions far from the droplet are 25 °C and 0 %RH.

Figure 9 – Gas phase velocity as calculated by MULTEVAP and a pseudo-steady state analysis.

The plot of the gas phase velocity calculated by MULTEVAP (MVAP), while similar in value to those calculated using the pseudo-steady state solution (SS), is not as accurate as expected. The discrepancy in values could be attributed to difficulties encountered while implementing the explicit solution for velocity. Implementing the boundary conditions at the droplet surface in the *Velocity* subroutine requires estimating the droplet surface velocity, u_i . There is a discrepancy in this estimated value and the value that is later calculated by inverting the A matrix. MULTEVAP is not able to progress through the evaporation of the droplet. Evaporation begins to occur at the start of the cycle, however, the species weight fractions cease to change after a short period of time.

11.1. Future Work

Additional work needs to be performed to determine the source of the error in the MULTEVAP program. A comparison of the velocity values output from the Velocity subroutine to experimental data would provide information about the accuracy of the subroutine. The discrepancy between the estimated surface velocity and the calculated surface velocity in the Velocity subroutine needs to be investigated. The method of predicting this value may need to be changed.

References

- Elperin, T.; Krasovitev, B. Evaporation of liquid droplets containing small solid particles. *Int. J. Heat Mass Transfer*. 1995, Vol. 38, No. 12, pp. 2259-2267
- Farid, M. A new approach to modeling of single droplet drying. *Chemical Engineering Science*. 2003, Vol. 58, pp. 2985-2993.
- Foss, W. Mathematical Model of a Single Multi-Component Evaporating Droplet. 1999. *Nektar Therapeutics (formerly Inhale Therapeutic Systems)*.
- Hecht, J.P.; King, C.J. Spray Drying: Influence of Developing Drop Morphology on Drying Rates and Retention of Volatile Substances. 1. Single-Drop Experiments. *Ind. Eng. Chem. Res.* 2000, Vol. 39, pp. 1756-1765.
- Kneer, R.; Schneider, M.; Noll, B.; Wittig, S. Effects of Variable Liquid Properties on Multicomponent Droplet Vaporization. *J. Engineering for Gas Turbines Power*. 1993, Vol. 115, pp. 467-472
- Lin, J. C.; Zhang, H.; Gentry, J. Spray Drying Droplet Morphology: Formation of Shell. *Journal of Aerosol Science*. 2000, Vol. 31, Suppl. 1, pp. S797-S798.
- Masters, K. *Spray Drying Handbook*, 4th ed.; Halsted Press: New York, 1985.
- Patton, J.; Platz, R. Pulmonary Delivery of Peptides and Proteins for systemic action. *Advanced Drug Delivery Reviews*. 1992, Vol. 8, pp. 179-228.
- Press, Teukolsky, Vetterling, & Flannery (1992), *Numerical Recipes in Fortran 77 - The Art of Scientific Computing, Second Edition, Volume 1 of Fortran Numerical Recipes*; Cambridge University Press, 1992.
- MacDonald. *LU Decomposition* (1998);
<http://www.physics.udel.edu/faculty/macdonald/>; accessed Feb. 2003.
- Nesic, S.; Vodnik, J. Kinetics of Droplet Evaporation. *Chemical Engineering Science*. 1991, Vol. 46, No. 2, pp. 527-537.
- Sezen, Y. Unsteady Mass Transfer Analysis of Evaporation of a Single Multicomponent Droplet with Diffusion and Constant Rate Processes. *Int. Comm. Heat Mass Transfer*. 2000, Vol. 27, No. 4, pp. 559-568.
- Renksizbulut, M.; Nafziger, R.; Li, X. A Mass Transfer Correlation for Droplet Evaporation in High-Temperature Flows. *Chemical Engineering Science*. 1991, Vol. 46, No. 9, pp. 2351-2358.

Tamim, J.; Hallett, W.L.H. A Continuous Thermodynamics Model for Multicomponent Droplet Vaporization. *Chemical Engineering Science*. 1995, Vol. 50, No. 18, pp. 2933-2942.

Yarin, A.L.; Brenn, G.; Kastner, O.; Tropea, C. Drying of acoustically levitated droplets of liquid-solid suspensions: Evaporation and crust formation. *Physics of Fluids*. 2002, Vol. 14, No. 7, pp. 2289-2298.

Appendix 1 – FORTRAN Subroutines

1. Subroutine <i>Velocity</i>	A1
2. Subroutine <i>ludcmp</i>	A11
3. Subroutine <i>lubksb</i>	A13

```

SUBROUTINE VELOCITY(W,T,ASTAR,DENS,CP,COND,DIFF,ENTH,
+  DENSLAVG,CPLAVG,UI,V)
c 34567 10 234567 20 234567 30 234567 40 234567 50 234567 60 234567 70 2
c
c  VELOCITY calculates the radial velocity at all nodes
c  both inside and outside the droplet by using
c  the equation of continuity. The form of the equation is different
c  in each phase.
c
c  Inputs:
c  W(L,J,N) = weight fraction at collocation point J, of species L and in phase N.
c  T(J) = array of dimensionless temperatures
c  ASTAR = the dimensionless radius of the sphere.
c  DENS(J,N) = mass density at point J in phase N
c  CP(J,N) = specific heat at point J in phase N
c  COND(J) = thermal conductivity a point J
c  DIFF(L,J,N) = diffusivity of species L at point J in phase N
c  ENTH(L,J,N) = specific enthalpy of species L
c  DENSLAVG = average density of the droplet
c  CPLAVG = average heat capacity of the droplet
c  UI = velocity of the surface of the sphere.
c
c  Outputs:
c    V(I,N) = velocity at collocation point I in phase N.
c
c  Working variables:
c    I = collocation point
c    J = collocation point
c    N = phase
c    WLIQ(L) = array of weight fractions for all L components in the liquid.
c    WGAS(L) = array of weight fractions for all L components in the gas.
c    THETA = dimensionless temperature
c    DENS = mass density.
c    DDENSDT(M) = the time (TAU) derivative of the density at some
c      integration point M.
c    X,Y = the dimensionless radial coordinate.
c  Subroutines and Functions:
c    DENSITY = a function which returns the density for a given
c      composition, WARRAY or WLASTARY, in phase N.
c
c 34567 10 234567 20 234567 30 234567 40 234567 50 234567 60 234567 70 2
IMPLICIT REAL*8(A-H,O-Z)
CHARACTER*8 NAME(10)
DIMENSION LBOTH(10),W0(10)

```

```

DIMENSION AA(201,201),V(101,2),RHS(201),DIFF(10,101,2)
DIMENSION CP(101,2),ENTH(10,101,2),COND(101),DENS(101,2)
DIMENSION W(10,101,2),WLIQ(10),WGAS(10),T(101),VEL(201,2)
DIMENSION X(101),Y(101),DX(101),DY(101),DDENSDT(201,2)
DIMENSION DDENSDW(10,201,2)
COMMON /PARAM/
NAME,NCASE,NL1,NL2,NL12,NL,KEY1,KEY2,LBOTH,W0,N1,
+ N2,A0,DIFF0,T0,TSPHERE0,TREF,TAU0
COMMON /FD/ X,Y,DX,DY
EXTERNAL DENSLIQ,DENSGAS,DDENSGASDT,DDENSGASW,DDENSLIQDT,
+ DDENSLIQDW
c
c   Print out AA matrix
c
OPEN (UNIT=60,FILE='AA.DAT',STATUS='NEW',FORM='FORMATTED')
OPEN (UNIT=90,FILE='UI.DAT',STATUS='NEW',FORM='FORMATTED')
c
c   Calculate the temperature derivative of the density at all points.
c
DO 33 J=1,N1                                     ! In the liquid
  DO 31 L=1,NL1
    WLIQ(L) = W(L,J,1)
  CONTINUE 31
  DDENSDT(J,1) = DDENSLIQDT(WLIQ,T(1)) ! T(1) is the sphere temp.
  DO 32 L=1,NL1
    IF (L.EQ. KEY1) GOTO 32
    DDENSDW(L,J,1) = DDENSLIQDW(L,WLIQ,T(1))
  CONTINUE 32
CONTINUE 33
DO 30 J=1,N2                                     ! In the gas
  DO 34 L=NL1+1,NL
    WGAS(L) = W(L,J,2)
  CONTINUE 34
  DDENSDT(J,2) = DDENSGASDT(WGAS,T(J))
  DO 35 L=NL1+1,NL
    IF (L-NL1.EQ. KEY2) GOTO 35
    DDENSDW(L,J,2) = DDENSGASDW(L,WGAS,T(J))
  CONTINUE 35
CONTINUE 30

V(1,1) = 0.D0                                     ! V=0 at center
c
c   initialize AA
c

```

```

DO 2 N=1,N1+N2-1
  DO 1 M=1,N1+N2-1
    AA(N,M)=0.D0
  CONTINUE 1
CONTINUE 2
c
c  Estimate the velocity of the surface, UI, using all volatile species
c
UI = 0.D0
DO 3 L=1,NL12                                     !Do only for the volatile components
  H1 = DX(N1-2)
  H2 = DX(N1-1)
  W1 = W(L,N1-2,1)
  W2 = W(L,N1-1,1)
  W3 = W(L,N1,1)
c  The first derivative at X=1 in the liquid phase using backward difference.
DWDX = H2/H1/(H1+H2)*W1-(H1+H2)/H1/H2*W2
+  +((H1+H2)**2-H2**2)/H1/H2/(H1+H2)*W3
L2 = L+NL1                                         ! The corresponding species in the gas phase.
H1 = DY(1)
H2 = DY(2)
W1 = W(L2,1,2)
W2 = W(L2,2,2)
W3 = W(L2,3,2)
c  The first derivative at Y=1 in the gas phase using forward difference.
DWDY = -1.D0*((H1+H2)**2-H1**2)/H1/H2/(H1+H2)*W1
+  +(H1+H2)/H1/H2*W2-H1/H2/(H1+H2)*W3
UIL = (DENS(1,2)*(DIFF(L2,1,2)/ASTAR*DWDY+W(L2,1,2)*V(1,2))
+  +DENS(N1,1)*(DIFF(L,N1,1)/ASTAR*DWDX-W(L,N1,1)*V(N1,1)))
+  /(DENS(1,2)*W(L2,1,2)-DENS(N1,1)*W(L,N1,1))
UI = UI+UIL
CONTINUE 3
c
c  calculate WGAS for use in ENTHGAST function
c
DO 4 L=1,NL1                                         ! loop thru all liquid species
  WLIQ(L)=W(L,N1,1)
CONTINUE 4
DO 5 L=NL1+1,NL                                     ! loop thru all gas species
  WGAS(L)=W(L,1,2)
CONTINUE 5

CD=3.D0*(DENS(1,2)*ENTHGAST(WGAS,T(1))+DENS(N1,1)
+  *ENTHVAP(WLIQ,T(1)))/DENSLAVG/CPLAVG/TREF/ASTAR

```



```

CE=3.D0*DENS(1,2)*ENTHGAST(WGAS,T(1))/DENSLAVG/CPLAVG/TREF/ASTAR
R
c
c      dT/dx taken at x=1 forward diff
c
H1=DY(1)
H2=DY(2)
T1=T(1)                                ! temp at j=1 gas phase
T2=T(2)                                ! temp at j=2 gas phase
T3=T(3)                                ! temp at j=3 gas phase
DTDY=-1.D0*((H1+H2)**2-H1**2)/H1/H2/(H1+H2)*T1
+ +(H1+H2)/H1/H2*T2-H1/H2/(H1+H2)*T3
CF=3.D0*COND(1)*DTDY/DIFF0/DENSLAVG/CPLAVG/ASTAR**2
c
c      construct inner matrix, liquid phase
c
DO 10 J=2,N1-1
  H1=DX(J-1)
  H2=DX(J)
  DENS1=DENS(J-1,1)
  DENS2=DENS(J,1)
  DENS3=DENS(J+1,1)
  DDENSDX=(H2/H1*(DENS2-DENS1)+H1/H2*(DENS3-DENS2))/(H1+H2)
  CG=X(J)*DDENSDX/ASTAR
  SUMA=0.D0
  SUMB=0.D0
  SUMC=0.D0
  DO 6 L=1,NL1
    IF (L .EQ. KEY1) GOTO 6
    W1=W(L,J-1,1)
    W2=W(L,J,1)
    W3=W(L,J+1,1)
    DWDX=(H2/H1*(W2-W1)+H1/H2*(W3-W2))/(H1+H2)
    SUMA=SUMA+X(J)/ASTAR*DWDX*DDENSDW(L,J,1)
    SUMB=SUMB+DWDX/ASTAR*DDENSDW(L,J,1)
    DIFF1=DIFF(L,J-1,1)
    DIFF2=DIFF(L,J,1)
    DIFF3=DIFF(L,J+1,1)
    DDIFFDX=(H2/H1*(DIFF2-DIFF1)+H1/H2*(DIFF3-DIFF2))/(H1+H2)
    D2WDX2=2.D0*(W3-2.D0*W2+W1-(H2-H1)*DWDX)/(H2**2+H1**2)
    DLAPLACE=D2WDX2+2.D0*DWDX/X(J)
    CC=((DIFF(L,J,1)/DENS(J,1)*DDENSDX+DDIFFDX)*DWDX+DIFF(L,J,1)
    + *DLAPLACE)/ASTAR**2
    SUMC=SUMC+CC*DDENSDW(L,J,1)

```

```

CONTINUE 6
SUMD = CD*DDENSDDT(J,1)
CH=X(J)**2*DX(J)*DENS(J-1,1)/DX(J-1)/(DX(J-1)+DX(J))
CI=2.D0*X(J)*DENS(J,1)+X(J)**2*DX(J)*DENS(J,1)/DX(J-1)
+ /(DX(J-1)+DX(J))-X(J)**2*DX(J-1)*DENS(J,1)/DX(J)
+ /(DX(J-1)+DX(J))
CJ=X(J)**2*DX(J-1)*DENS(J+1,1)/DX(J)/(DX(J-1)+DX(J))
SUME=CE*DDENSDDT(J,1) !changed from -1.D0*CE*DDENSDDT(J,1)
SUMF=CF*DDENSDDT(J,1)

IF (J.NE. 2) AA(J-1,J-2)=CH/ASTAR/X(J)**2           ! v(j-1)
AA(J-1,J-1)=SUMB-CI/X(J)**2/ASTAR                   ! v(j)
AA(J-1,J)=-1.D0*CJ/ASTAR/X(J)**2                   ! v(j+1)
AA(J-1,N1)=-1.D0*SUMD                                ! ui
AA(J-1,N1+1)=SUME                                    ! v(g)(y=1)
RHS(J-1)=SUMC-SUMF                                   ! RHS

CONTINUE 10
c
c      j = n1, backward difference
c
J=N1                                                    ! droplet surface
H1=DX(J-2)
H2=DX(J-1)
DENS1=DENS(J-2,1)
DENS2=DENS(J-1,1)
DENS3=DENS(J,1)
DDENSDDX=H2/H1/(H1+H2)*DENS1-(H1+H2)/H1/H2*DENS2+((H1+H2)**2-H2**2)
+ /H1/H2/(H1+H2)*DENS3
CG=X(J)*DDENSDDX/ASTAR
SUMA=0.D0
SUMB=0.D0
SUMC=0.D0
DO 20 L=1,NL1
  IF (L.EQ.KEY1) GOTO 20                                ! skip key phase
  IF (L.BOTH(L).EQ.1) THEN                              ! if species L exist in both phases
    L2=L+NL1                                           ! the corresponding species in the gas phase
    H1=DY(1)
    H2=DY(2)
    W1=W(L2,1,2)
    W2=W(L2,2,2)
    W3=W(L2,3,2)
c      the first derivative at Y=1 in the gas phase, forward difference
    DRTVGAS=-1.D0*((H1+H2**2)-H1**2)/H1/H2/(H1+H2)*W1
    + +(H1+H2)/H1/H2*W2-H1/H2/(H1+H2)*W3

```

```

c    the first derivative at X=1 in the liquid phase from the BC of species L
      DWDX=(ASTAR*W(L,N1,1)*(V(N1,1)-UI)
      +   -DENS(1,2)/DENS(N1,1)*(DIFF(L2,1,2)*DRTVGAS
      +   +ASTAR*W(L2,1,2)*(V(1,2)-UI))/DIFF(L,N1,1)
      ELSE                                     ! for species confined to the liquid
        DRTVGAS=0.D0
        DWDX=ASTAR*W(L,N1,1)*(V(N1,1)-UI)/DIFF(L,N1,1)
      END IF
      SUMA=SUMA+X(J)/ASTAR*DWDX*DDENS(DW(L,J,1)
      SUMB=SUMB+DWDX/ASTAR*DDENS(DW(L,J,1)
      WFALSE=W(L,N1-1,1)+2.D0*DX(N1-1)*DWDX      ! dwdx taken from b.c.
      XMID1=1.D0-DX(N1-1)/2.D0
      XMID2=1.D0+DX(N1-1)/2.D0
c    The Laplacian in spherical coordinates at X=1 using the false point
      DLAPLACE=(XMID2**2*WFALSE-(XMID2**2+XMID1**2)*W(L,N1,1)
      +   +XMID1**2*W(L,N1-1,1))/DX(N1-1)**2
      H1=DX(J-2)                                ! drop surface
      H2=DX(J-1)
      DIFF1=DIFF(L,N1-2,1)
      DIFF2=DIFF(L,N1-1,1)
      DIFF3=DIFF(L,N1,1)
      DDIFFDX=H2/H1/(H1+H2)*DIFF1-(H1+H2)/H1/H2*DIFF2+      ! dDIFF/dx
      backward diff
      +   ((H1+H2)**2-H2**2)/H1/H2/(H1+H2)*DIFF3
      CC=((DIFF(L,J,1)/DENS(J,1)*DDENS(DW(L,J,1)
      +   *DLAPLACE)/ASTAR**2
      SUMC=SUMC+CC*DDENS(DW(L,J,1)
CONTINUE 20
SUMD = CD*DDENS(DT(J,1)
CK=DX(J-1)*X(J)**2*DENS(J-2,1)/DX(J-2)/(DX(J-1)+DX(J-2))
CL=(DX(J-1)+DX(J-2))*X(J)**2*DENS(J-1,1)/DX(J-1)/DX(J-2)
CM=2.D0*X(J)*DENS(J,1)+((DX(J-1)+DX(J-2))**2-DX(J-1)**2)
+   *X(J)**2*DENS(J,1)/DX(J-1)/DX(J-2)/(DX(J-1)+DX(J-2))
SUME = CE*DDENS(DT(J,1)
SUMF = CF*DDENS(DT(J,1)

AA(N1-1,N1-3)=-1.D0*CK/ASTAR/X(J)**2      ! v(j-2) liq
AA(N1-1,N1-2)=CL/X(J)**2/ASTAR             ! v(j-1) liq
AA(N1-1,N1-1)=SUMB-CM/ASTAR/X(J)**2       ! v(j) liq
AA(N1-1,N1)=-1.D0*SUMD ! CG-SUMA-SUMD     ! ui
AA(N1-1,N1+1)=SUME                         ! v(1) gas
RHS(N1-1)=SUMC-SUMF                       ! RHS
c
c    jump condition at liquid-gas interface

```

```

c
AA(N1,N1-1)=DENS(N1,1)
AA(N1,N1)=DENS(1,2)-DENS(N1,1)
AA(N1,N1+1)=-1.D0*DENS(1,2)
RHS(N1)=0.D0
c
c  gas phase, j=1, forward difference
c
J=1
H1=DY(1)
H2=DY(2)
DENS1=DENS(1,2)
DENS2=DENS(2,2)
DENS3=DENS(3,2)
DDENSDY=-1.D0*((H1+H2)**2-H1**2)/H1/H2/(H1+H2)*DENS1
+ +(H1+H2)/H1/H2*DENS2-H1/H2/(H1+H2)*DENS3
CT=-1.D0*Y(1)*DDENSDY/ASTAR
SUMN=0.D0
SUMO=0.D0
SUMP=0.D0
DO 36 L=NL1+1,NL                                ! loop species in gas phase
  IF (L-NL1 .EQ. KEY2) GOTO 36
  IF (LBOTH(L).EQ.1) THEN                          ! if species L exist in both phases
    L1=L-NL1                                       ! the corresponding species in the gas phase
    H1=DX(N1-2)
    H2=DX(N1-1)
    W1=W(L1,N1-2,1)
    W2=W(L1,N1-1,1)
    W3=W(L1,N1,1)
c    the first derivative at X=1 in the liquid phase, backwards difference
    DRTVLIQ= H2/H1/(H1+H2)*W1-(H1+H2)/H1/H2*W2
    + +(H1+H2)**2-H2**2)/H1/H2/(H1+H2)*W3
c    the first derivative at X=1 in the liquid phase from the BC of species L
    DWDY=(DENS(N1,1)/DENS(1,2)*(ASTAR*W(L1,N1,1)*(V(N1,1)-UI)
    + -DIFF(L1,N1,1)*DRTVLIQ)-ASTAR*W(L,1,2)*(V(1,2)-UI))
    + /DIFF(L,1,2)/Y(1)**2
  ELSE
c    for the surface point in the gas phase, X=1 only if necessary
    DWDY=-1.D0*ASTAR*W(L,1,2)*(V(1,2)-UI)/DIFF(L,1,2) ! dw/dy from bc
  ENDIF
  SUMN=SUMN-Y(J)*DWDY/ASTAR*DDENSDW(L,1,2)
  SUMO=SUMO+Y(J)**2*DWDY/ASTAR*DDENSDW(L,1,2)
  WFALSE=W(L,2,2)-2.D0*DY(1)*DWDY
  D2WDY2=(WFALSE-2.D0*W(L,1,2)+W(L,2,2))/DY(1)**2

```

```

DIFF1=DIFF(L,1,2)
DIFF2=DIFF(L,2,2)
DIFF3=DIFF(L,3,2)
DDIFFDY=-1.D0*((H1+H2)**2-H1**2)/H1/H2/(H1+H2)
+ *DIFF1+(H1+H2)/H1/H2*DIFF2-H1/H2/(H1+H2)*DIFF3
COEFP=((DIFF(L,1,2)/DENS(1,2)*DDENSDY+DDIFFDY)      ! COEFP used
CP in use
+ *DWDY+DIFF(L,1,2)*D2WDY2)*Y(1)**4/ASTAR**2
SUMP=SUMP+COEFP*DDENSDW(L,1,2)
CONTINUE 36
T1=T(1)      ! temp of liq at interface
T2=T(2)      ! temp at j=1 gas phase
T3=T(3)      ! temp at j=1 gas phase
DTDY=-1.D0*((H1+H2)**2-H1**2)/H1/H2/(H1+H2)*T1
+ +(H1+H2)/H1/H2*T2-H1/H2/(H1+H2)*T3
SUMD=CD*DDENSDT(1,2)
CU=-1.D0*((DY(1)+DY(2))**2-DY(1)**2)/Y(1)**2/DY(1)/DY(2)
+ /(DY(1)+DY(2))*DENS(1,2)-2.D0*DENS(1,2)/Y(1)**3
CV=(DY(1)+DY(2))/Y(1)**2/DY(1)/DY(2)*DENS(2,2)
CW=-1.D0*DY(1)/Y(1)**2/DY(2)/(DY(1)+DY(2))*DENS(3,2)
SUME=CE*DDENSDT(1,2)
SUMF=CF*DDENSDT(1,2)

AA(N1+1,N1)=-1.D0*SUMD !CT-SUMN-SUMD      ! ui
AA(N1+1,N1+1)=Y(1)**4*CU/ASTAR-SUMO+SUME      ! v(1)
AA(N1+1,N1+2)=Y(1)**4*CV/ASTAR      ! v(2)
AA(N1+1,N1+3)=Y(1)**4*CW/ASTAR      ! v(3)
RHS(N1+1)=SUMP-SUMF      ! rhs,end j=1

c
c  gas phase inner points, j=2-n2-1, central differences
c
DO 56 J=2,N2-1
  H1=DY(J-1)
  H2=DY(J)
  DENS1=DENS(J-1,2)
  DENS2=DENS(J,2)
  DENS3=DENS(J+1,2)
  DDENSDY=(H2/H1*(DENS2-DENS1)+H1/H2*(DENS3-DENS2))/(H1+H2)
  CT=-1.D0*Y(J)*DDENSDY/ASTAR
  COND1=COND(J-1)
  COND2=COND(J)
  COND3=COND(J+1)
  DKDY=(H2/H1*(COND2-COND1)+H1/H2*(COND3-COND2))/(H1+H2)
  T1=T(J-1)

```

```

T2=T(J)
T3=T(J+1)
DTDY=(H2/H1*(T2-T1)+H1/H2*(T3-T2))/(H1+H2)
D2TDY2=2.D0/(H2**2+H1**2)*(T3-2.D0*T2+T1-(H2-H1)*DTDY)
SUMN=0.D0
SUMO=0.D0
SUMP=0.D0
DSUMDY=0.D0
DO 58 L=NL1+1,NL
  IF (L-NL1 .EQ. KEY2) GOTO 58
  W1=W(L,J-1,2)
  W2=W(L,J,2)
  W3=W(L,J+1,2)
  DWDY=(H2/H1*(W2-W1)+H1/H2*(W3-W2))/(H1+H2)
  D2WDY2=2.D0/(H2**2+H1**2)*(W3-2*W2+W1-(H2-H1)*DWDY)
  DIFF1=DIFF(L,J-1,2)
  DIFF2=DIFF(L,J,2)
  DIFF3=DIFF(L,J+1,2)
  DDIFFDY=(H2/H1*(DIFF2-DIFF1)+H1/H2*(DIFF3-DIFF2))/(H1+H2)
  SUMN=SUMN-Y(J)*DWDY/ASTAR*DDENSDW(L,J,2)
  SUMO=SUMO+Y(J)**2*DWDY/ASTAR*DDENSDW(L,J,2)
  COEFP=((DIFF(L,J,2)/DENS(J,2)*DDENSDY+DDIFFDY)*DWDY
  + +DIFF(L,J,2)*D2WDY2)*Y(J)**4/ASTAR**2
  SUMP=SUMP+COEFP*DDENSDW(L,J,2)
  DENS1=DENS(J-1,2)
  DENS2=DENS(J,2)
  DENS3=DENS(J+1,2)
  ENTH1=ENTH(L,J-1,2)
  ENTH2=ENTH(L,J,2)
  ENTH3=ENTH(L,J+1,2)
  ENTHN1=ENTH(NL1+KEY2,J-1,2)
  ENTHN2=ENTH(NL1+KEY2,J,2)
  ENTHN3=ENTH(NL1+KEY2,J+1,2)
  DSUMDY=DSUMDY+((ENTH2-ENTHN2)*DENS2*DIFF2*D2WDY2
  + +DWDY/(H1+H2)*(H2/H1*((ENTH2-ENTHN2)*DENS2*DIFF2
  + -(ENTH1-ENTHN1)*DENS1*DIFF1)
  + +H1/H2*((ENTH3-ENTHN3)*DENS3*DIFF3
  + -(ENTH2-ENTHN2)*DENS2*DIFF2)))
CONTINUE 58
CAA=-1.D0*Y(J)/ASTAR*DTDY*DDENSDT(J,2)
CBB=Y(J)**2*DTDY/ASTAR*DDENSDT(J,2)
CCC=Y(J)**4/DENS(J,2)/CP(J,2)/ASTAR**2*(1.D0/DIFF0*(DKDY*DTDY
+ +COND(J)*D2TDY2)+1.D0/TREF*DSUMDY)*DDENSDT(J,2)
CX=-1.D0*H2*DENS(J-1,2)/Y(J)**2/(H1+H2)/H1

```

```

CY=H2*DENS(J,2)/H1/Y(J)**2/(H1+H2)-H1*DENS(J,2)/H2
+ /Y(J)**2/(H1+H2)-2.D0*DENS(J,2)/Y(J)**3
CZ=H1*DENS(J+1,2)/(H1+H2)/H2/Y(J)**2

AA(N1+J,N1)=-1.D0*CAA !CT-SUMN-CAA
AA(N1+J,N1+J-1)=Y(J)**4*CX/ASTAR
AA(N1+J,N1+J)=Y(J)**4*CY/ASTAR-SUMO-CBB
IF (J.NE. N2-1) AA(N1+J,N1+J+1)=Y(J)**4*CZ/ASTAR
RHS(N1+J)=SUMP+CCC
CONTINUE 56
c
c      Solve for velocity using LU decomposition
c
MATRIX = N1+N2-1
NP = 201
CALL LUDCMP(AA,MATRIX,NP,INDX,D)
CALL LUBKSB(AA,MATRIX,NP,INDX,RHS)

V(1,1) = 0.D0
DO 90 J=2,N1
    V(J,1) = RHS(J-1)
90 CONTINUE
UI = RHS(N1)
DO 91 J=1,N2-1
    V(J,2) = RHS(N1+J)
91 CONTINUE
V(N2,2) = 0.D0

RETURN
END

```

```

SUBROUTINE ludcmp(a,n,np,indx,d)
INTEGER n,np,indx(n),NMAX
REAL*8 d,a(np,np),TINY
PARAMETER (NMAX=500,TINY=1.0d-20)
INTEGER i,imax,j,k
REAL*8 aamax,dum,sum,vv(NMAX)

d = 1.d0
do 12 i=1,n
  aamax = 0.d0
  do 11 j = 1,n
    if (abs(a(i,j)).gt.aamax) aamax = abs(a(i,j))  ! largest element in column
  continue 11
  if (aamax.eq.0.d0) pause 'singular matrix in ludcmp'
  vv(i) = 1.d0/aamax  ! scaling factor
continue 12
do 19 j = 1,n  ! Crout's Method, loop through columns
  do 14 i = 1,j-1  ! loop through rows above diagonal
    sum = a(i,j)
    do 13 k = 1,i-1
      sum = sum-a(i,k)*a(k,j)  ! Calculate  $\exists$  term for eq.(5.15)
    continue 13
    a(i,j) = sum
  continue 14
  aamax = 0.d0
  do 16 i = j,n  ! Loop through rows on and below the diag.
    sum = a(i,j)
    do 15 k = 1,j-1
      sum = sum-a(i,k)*a(k,j)  ! Calculate  $\forall$  term for eq. (5.16)
    continue 15
    a(i,j) = sum
    dum = vv(i)*abs(sum)
    if (dum.ge.aamax) then
      imax = i  ! Record row with largest magnitude element
      aamax = dum
    endif
  continue 16
  if (j.ne.imax) then  ! If the largest magnitude element is off the diag.
    do 17 k = 1,n
      dum = a(imax,k)  ! Move the largest magnitude element to diag.
      a(imax,k) = a(j,k)
      a(j,k) = dum
    continue 17
  d = -d

```



```

        vv(imax) = vv(j)
    endif
    indx(j) = imax
    if (a(j,j).eq.0.d0) a(j,j) = TINY
    if (j.ne.n) then                ! Complete division by the pivot element (5.16)
        dum = 1.d0/a(j,j)
        do 18 i = j+1,n
            a(i,j) = a(i,j)*dum
        continue 18
    endif
    continue 19

return
END

```

```

SUBROUTINE lubksb(a,n,np,indx,b)
INTEGER n,np,indx(n)
REAL*8 a(np,np),b(n)
INTEGER i,ii,j,ll
REAL*8 sum

ii=0
do 12 i=1,n
  ll=indx(i)
  sum=b(ll)
  b(ll)=b(i)
  if (ii.ne.0)then
    do 11 j=ii,i-1
      sum=sum-a(i,j)*b(j)
    continue 11
  else if (sum.ne.0.) then
    ii=i
  endif
  b(i)=sum
continue 12
do 14 i=n,1,-1
  sum=b(i)
  do 13 j=i+1,n
    sum=sum-a(i,j)*b(j)
  continue 13
  b(i)=sum/a(i,i)
continue 14

return
END

```