To the Graduate Council:

I am submitting herewith a thesis written by Jessica Lee Gass entitled "Comparison of Numerical Models in Condensation Theory." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in mathematics.

K C Reddy

Dr. K. C. Reddy, Major Professor

We have read this thesis
and recommend its acceptance:

Frank S. Collins

Charles K. Mesler

Accepted for the Council:

L. W. Minkel

Associate Vice Chancellor
and Dean of the Graduate School

COMPARISON OF NUMERICAL MODELS IN CONDENSATION THEORY

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Jessica Lee Gass

August 1999

DEDICATION

This thesis is dedicated to my husband:

H. Dan Thomas

and also my parents:

Gerry L. Gass

and

Diane M. Gass

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Frank Collins, for giving me the opportunity to pursue graduate education and for giving me my thesis topic. I would also like to express appreciation for the other members of my thesis committee: Dr. Charles Merkle and Dr. K. C. Reddy. I am most grateful to the staff at the University of Tennessee Space Institute library especially Mary Lo who did literature searches for me and amazes me at how much she gets done in such a short time. Most of all I would like to thank my friends who kept me from going insane through the thesis process: Kelly Gass, Jamie Umscheid, Allen Hild (thanks for letting me use your scanner!), Sheri Hlady, Allen McCarley, and Peter Montgomery. Thanks to my family for all the encouragement throughout my entire life. Of course, I thank my husband who has not only loved me but has been a valuable teacher. I want to also thank God for giving me the will to persevere and the avenue of prayer, which has been most relieving.

ABSTRACT

The primary purpose of this thesis is to solve equations that model the nucleation of water molecules. The equations are derived for the number of clusters (nuclei) with a given number of molecules and the rates of development of the clusters. A finite difference algorithm is formulated based on the Crank-Nicolson scheme to calculate the nonequilibrium formation of clusters. The results gathered from this procedure are compared to results from another source. Also, the numerical results are compared to the classical equation used for the formation rate of clusters. The comparison shows a slight difference between the two sets of results.

After expanding to include a full condensation model, an ultimate goal would be to incorporate the developed code into an existing Computational Fluid Dynamics (CFD) flow solver which then can be used to simulate condensation created for flow visualization during wind tunnel tests.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

A_1	Single vapor molecule
A_g	Cluster with g molecules
A_i	Cluster with i molecules
A_j	Cluster with j molecules
f_g	Number of clusters with g molecules
f_g/n_1	Probability of having clusters with g molecules
g	Number of molecules in a droplet
g^*	Number of molecules in a droplet of critical size
I^0	Rate of formation of clusters of critical size
I_g	Formation rate of clusters
k	Boltzmann constant
K_g	Ratio of reaction rates in equilibrium
$K_g^{(f)}$	Forward reaction rate
$K_g^{(r)}$	Reverse reaction rate
m	Mass of a molecule
n_g	Equilibrium Boltzmann statistical distribution
n_1	Distribution of molecules
p_c	Pressure at the saturation point (for p-T diagram)
p_v	Vapor pressure at the supersaturated conditions
p_∞	Saturation vapor pressure at the mixture temperature
r	Droplet radius

r^*	Critical droplet radius
T	Temperature
T_∞	Saturation temperature
V_l	Volume of a single molecule
α_c	Condensation coefficient
χ	Temperature gradient
ΔG	Gibbs free energy
ΔG^*	Gibbs free energy at critical size
μ_l	Chemical potential of the liquid
μ_v	Chemical potential of the vapor
ρ_l	Density of the liquid
σ	Specific surface free energy (surface tension)

1. INTRODUCTION

1.1 Statement of Problem

Numerical analysis is a useful tool for many different aspects of science. Computational results can be gathered to aid experimental methods, particularly for wind tunnel testing that relies strongly on computational procedures. Before performing experiments in wind tunnels, one would like to know the most advantageous testing conditions. Most numerical methods for wind tunnels primarily produce data that directly relate to the experiments, such as flow patterns, but for this thesis the concern is with a method of flow visualization used to observe those flow patterns.

There are many methods of flow visualization. The vapor screen technique which uses condensation is of particular interest. A numerical model is needed to obtain the correct parameters for testing. Ryzhov, Pirumov, and Gorbunov in reference (1) have developed a model for the beginning stages of condensation. This work is important since the model may help predict when condensation occurs in wind tunnel testing. The equations used in this reference are verified and some variables changed for better accuracy. This thesis attempts to confirm the results from the reference.

1.2 Flow Visualization

Flow visualization methods are important to aerodynamic research, since there are many phenomena that occur within a flow field that cannot be seen by the human eye. The purpose of these techniques is "to enable direct observation of the corresponding fluid motion" around the test model (Asanuma 2). Examining the flow patterns can

allow a better grasp of the flow physics and can be used to create mathematical models.

Experiments using flow visualization have led to many discoveries in low and high-speed flows. In 1883, Osborne Reynolds found a transition from laminar to turbulent flow by incorporating dye in glass tubes to observe the flow of water. This detection was the start of many studies to understand this transformation. Also in the 1880's, another crucial breakthrough came from acoustic tests done by Ernst Mach that led to investigations with high-speed flows. As a result of this research "the first successful photographs of a supersonic projectile flowfield" were taken (Mueller 3). These examples show the importance of flow visualization in identifying many supersonic flow phenomena.

Presently, flow visualization is widely used, and numerous techniques have been developed. Some methods such as schlieren, shadowgraph, and holography use optics to visualize flow patterns. These methods require complex systems of mirrors, lenses, light sources, and cameras to record experiments. Another technique known as the smoke-wire method employs a wire covered with oil, which produces smoke filaments when heated. Oil can also be used for surface flow pattern techniques. A thin layer of a liquid is applied to the surface of the model that reveals a pattern during testing (Settles 4).

1.3 Vapor Screen Method

The scattering of an intense sheet of laser light from aerosol particles is the requisite for the vapor screen method. The sheet is placed in the flow field perpendicular to the direction of the flow. Then, a camera is used to record the now visible flow patterns. For this method to work properly there must be a concentrated area of fog in

the flow field of interest (Collins 5).

In the analysis described in this study, the aerosol particles are water droplets which make the method complex, since conditions in the wind tunnel must be favorable to produce condensation. This procedure is also difficult since there must be some effort placed on introducing condensation to the correct area of the flow. Knowing these factors, the prediction of condensation is significant to the success of the method (Settles 4).

Despite the problems with vapor screens there are advantages. Some of the requirements for the occurrence of condensation (such as humidity) are readily available to the test section, which means less trouble preparing the test facility. Also the formation of droplets happens naturally and is therefore beneficial to the procedure.

1.4 Numerical Modeling

A model is an equation or set of equations that represents a process (Fowler 6). Solving these equations will hopefully give an accurate representation of what would happen in an actual experiment. Since the model equations usually do not have analytical solutions, the solution to the equations heavily relies on a computer that can solve equations with their associated boundary conditions.

The first step in modeling is to decide what is the problem. For the condensation model, when does condensation occur? This leads to the definition of variables that relate to the problem such as, in the case of condensation, temperature, pressure, number density of water droplets, etc. Next, a relation has to be found for the parameters either by experiment or theoretical reasoning. Once this is done, the model is expressed

mathematically. Fortunately, for this problem, the theory has already been established, and the equations that govern the process are known. The purpose of this thesis is to confirm that the theory is correct by solving the equations and comparing with results gathered from reference (1).

1.5 Summary of Reference (1)

Ryzhov, Pirumov, and Gorbunov (1) performed a study in the 1980's of nonequilibrium condensation as a result of growing attention to diverse problems concerning condensation. The reference gives an assortment of articles done prior to the study to show the significance of the issue. The motivation to the research was to review general condensation theory in high-speed flows and present numerical techniques that can be used to solve the governing equations.

The text gave an explanation of the theory for homogeneous and heterogeneous condensation by exhibiting the governing equations and also discussed the numerical methods used to solve these equations. In addition, a survey was done of experimental techniques used to study condensation. A chapter was dedicated to the thermodynamic properties and to determine the parameters for homogeneous condensation. This thesis uses the governing equations discussed above to develop a computer code to model homogeneous condensation. Comparison is made with the solution of these equations as presented by Ryzhov, Pirumov, and Gorbunov (1).

2. CONDENSATION THEORY

2.1 Explanation of Water Vapor Condensation

Condensation is either homogeneous or heterogeneous. Heterogeneous condensation happens when the flow is seeded with foreign particles. For wind tunnel testing, condensation has been observed to be homogeneous since there is only one condensing component and not enough foreign particles naturally occur to account for the large number of condensation nuclei that are measured. Wind tunnel air is considered to be moist air which is assumed to be a mixture of perfect gases, including water vapor which is the only condensable element considered in this analysis of homogeneous condensation.

As the air expands through a nozzle it becomes saturated and eventually supersaturated. The behavior of condensation for the flow through a nozzle can be explained by the p-T diagram in Figure 2.2.1. First the gas expands along the adiabatic curve 2 passing the place where the saturation curve 1 and curve 2 cross, which is the saturation point (Ryzhov et al. 1). After this point the water vapor advances to a state of supersaturation at p_c at which time condensation spontaneously occurs, and then it advances to the saturation curve. The droplets are produced from the water vapor by spontaneous nucleation and rapid growth of initially small nuclei. As the expansion continues the water vapor remains on the saturation curve until it is condensed onto the droplets and the air remains saturated. For this thesis, the initial departure from the supersaturation point, p_c, is analyzed.

The process of condensation of the supersaturated air takes place in a short

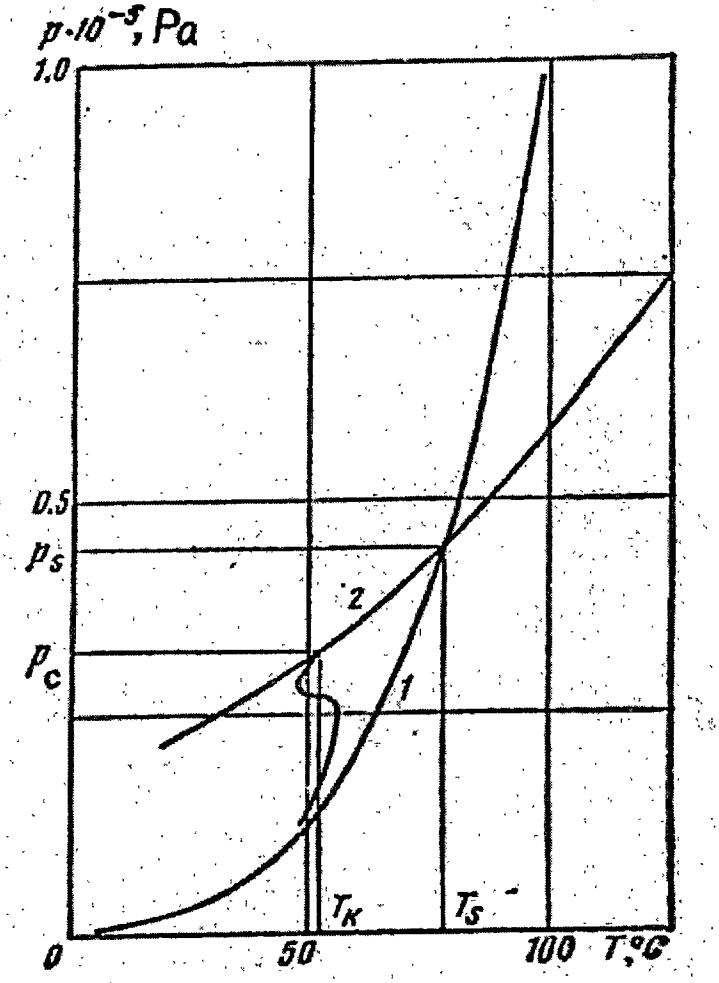

Figure 2.2.1: p-T Diagram: (1) Saturation Curve and (2) Expansion Process from Ref. (1)

amount of time, which can be divided, into three periods. The initial period is a very slight transient period which lasts only tens of microseconds "when the embryo population is steadying off up to moderately large (McDonald 7)" numbers of molecules. The second period lasts a little longer, about tens or hundreds of milliseconds, in which supercritical embryos are being produced at a quasi-steady rate. At this point the embryos are stable and will continue to grow until the process consumes the excess vapor. Nucleation is considered to be complete at this level. The third stage deals with diffusional growth where the vapor pressure drops from supersaturated to saturated. Only the first stage (nucleation), to be described in the next section, will be examined in this thesis.

2.2 Homogeneous Condensation

The most important part of condensation theory is in the study of the development of condensation nuclei. As stated in the previous section, there are two types of condensation. Homogeneous condensation is considered in this study, and the analysis follows that given in reference (1). Initially, the phases, liquid and vapor, are in equilibrium; however, during rapid expansion in a nozzle, a nonequilibrium state of supersaturation develops. Growth of the water droplet nuclei proceeds through departures from equilibrium between the phases.

In equilibrium, the liquid and gaseous phases have the same temperature. To describe the nucleation of spherical droplets, the change in the Gibbs free energy required to generate a droplet of radius, r, is used:

$$\Delta G = g[\mu_l(p_\infty, T) - \mu_v(p_v, T)] + 4\pi r^2 \sigma \qquad (2.2.1)$$

where g is the number of molecules in a droplet with radius r, μ_l and μ_v are the chemical potentials of the liquid and the vapor, p_v and p_∞ are the vapor pressure at the supersaturated conditions in the air and saturation vapor pressure at the mixture temperature, respectively, and σ is the specific surface free energy (surface tension), which is assumed to be independent of the number of molecules in the droplet. The radius of the spherical droplet can be written as:

$$r = \left(\frac{3gV_l}{4\pi} \right)^{1/3} \tag{2.2.2}$$

where V_l is the volume of a single molecule. It is obvious from (2.2.2) that:

$$g = \frac{4\pi r^3}{3V_l} \tag{2.2.3}$$

To determine the equilibrium conditions, the Gibbs free energy, ΔG, is minimized:

$$\frac{d\Delta G}{dg} = \mu_l(p_\infty, T) - \mu_v(p_v, T) + 4\pi\sigma \frac{dr^2}{dg}\left(1 + \frac{1}{2}\frac{d\ln\sigma}{dr^2} \right) = 0 \tag{2.2.4}$$

Assuming the surface tension, σ, is not dependent on the radius, r, the critical number of molecules in a cluster can be acquired by solving (2.2.4) for $g = g^*$:

$$g^* = \frac{32\pi\sigma^3 V_l^2}{3(\mu_v(p_v, T) - \mu_l(p_\infty, T))^3} \tag{2.2.5}$$

As stated above, the vapor is assumed to be a mixture of ideal gases, hence $\mu_v(p_v,T) - \mu_l(p_\infty,T) = kT\ln(p_v/p_\infty)$, where k is the Boltzmann constant, and so (2.2.5) changes to:

$$g^* = \frac{32\pi\sigma^3 V_l^2}{3\left(kT\ln\left(\frac{p_v}{p_\infty} \right) \right)^3} \tag{2.2.6}$$

and rearranging gives:

$$kT\ln\left(\frac{p_v}{p_\infty}\right)=\left(\frac{32\pi\sigma^3 V_l^2}{g^*}\right)^{1/3} \tag{2.2.7}$$

The critical radius, r^*, is obtained by substituting the critical number of molecules, g^*, into (2.2.2):

$$r^*=\frac{2\sigma V_l}{kT\ln\left(\frac{p_v}{p_\infty}\right)} \tag{2.2.8}$$

and now by placing the critical values from (2.2.6) and (2.2.8) into the Gibbs free energy equation, (2.2.1) becomes:

$$\Delta G^*=\frac{16\pi}{3}\left(\frac{V_l}{kT\ln\left(\frac{p_v}{p_\infty}\right)}\right)^2\sigma^3 \tag{2.2.9}$$

The next step is to express the Gibbs equilibrium equation in terms of the number of molecules in a cluster, g, and the number of molecules in a cluster when $\Delta G = \Delta G^*$, g^*, by first substituting (2.2.7) and (2.2.2) into (2.2.1). After then factoring out the ΔG^* term from (2.2.9) the new Gibbs equilibrium equation is:

$$\Delta G=\sigma\left(\frac{4\pi V_l^2}{3}\right)^{1/3} g^{*2/3}\left[3\left(\frac{g}{g^*}\right)^{2/3}-2\frac{g}{g^*}\right] \tag{2.2.10}$$

Nucleation occurs when the clusters of molecules in a vapor combine with another molecule, resulting in larger clusters. This process has been shown to occur by the addition of a single vapor molecule, A_1, at each step. The chemical reactions required

to form a cluster having g water molecules is:

$$A_1+A_1 \Leftrightarrow A_2$$
$$A_2+A_1 \Leftrightarrow A_3$$

.

.

.

$$A_{g-1}+A_1 \Leftrightarrow A_g$$

where $g>g^*$ and A_g is a cluster with g molecules (g-mers). In the case of condensation, clusters grow from A_g to A_{g+1} by the addition of a single molecule; conversely, evaporation results from the loss of a molecule (i.e. A_{g+1} to A_g) where the cluster becomes smaller. Also, the formation of larger clusters can result from the collision of two smaller clusters containing i and j molecules:

$$A_i+A_j \rightarrow A_{i+j}$$

or can separate into two smaller clusters:

$$A_{i+j} \rightarrow A_i+A_j$$

These two reactions have been shown to be so rare that they can be ignored. The formation of dimers, a cluster with two molecules, requires a three-body collision to satisfy conservation of momentum and energy:

$$A_1+A_1+A_1 \Leftrightarrow A_2+A_1;\ A_1+A_1+B \Leftrightarrow A_2+B$$

where B is a third substance such as the carrier gas.

Condensation occurs when the growth process dominates the decay process and the number of large clusters increases. Consider one of the reaction rates above when condensation occurs. When molecules are added, the process is described by a forward

reaction rate, $K_g^{(f)}$. Conversely, if molecules are lost (evaporation) then the process is a reverse reaction rate, $K_g^{(r)}$. The rate of formation, I_g, of clusters containing g molecules can be found if the reaction rate constants are known. The forward reaction constant from reference (1), which is simply proportional to the rate at which water molecules collide with the clusters' surface, is given by:

$$K_g^{(f)} = \alpha_c 4\pi r^2 \frac{p_v}{(2\pi mkT)^{1/2}} \tag{2.2.11}$$

where α_c is the condensation coefficient. The condensation coefficient gives the probability that colliding water molecule and cluster will form a larger cluster. Since the reverse reaction constant $K_g^{(r)}$ is difficult to determine, the ratio of the reaction rates in equilibrium is used (the law of mass action):

$$K_g = \frac{n_g}{n_{g-1} n_1} = \frac{K_g^{(f)}}{K_g^{(r)}} \tag{2.2.12}$$

where

$$n_g = n_1 \exp\left(-\frac{\Delta G}{kT}\right) \tag{2.2.13}$$

is the equilibrium Boltzmann statistical distribution, $n_1 = p_v/kT$, and K_g is the equilibrium constant. From the evaporation and condensation reactions, the number of clusters, f_g, of class g per unit volume can be obtained by the following equation:

$$\frac{\partial f_g}{\partial t} = I_g - I_{g+1}, \quad g = (2,...,G) \tag{2.2.14}$$

where from reference (1), I_g represents the rate of formation of clusters given by:

$$I_g = K_g^{(f)} n_1 f_{g-1} - K_g^{(r)} f_g = K_g^{(f)} n_1 n_{g-1} \left(\frac{f_{g-1}}{n_{g-1}} - \frac{f_g}{n_g} \right) \qquad (2.2.15)$$

For equilibrium conditions the number of clusters is given by the equilibrium distribution function, n_g, and can be found from (2.2.13), but for the nonequilibrium conditions the system of differential equations (2.2.14) must be solved to find the number of clusters, f_g. For very large g, with $\Delta g = \pm 1$, the rate of formation of the clusters from (2.2.15) can be rewritten as:

$$I(g,t) = -K_g^{(f)} n_1 n_g \frac{\partial}{\partial g}\left(\frac{f_g}{n_g} \right) = -D \frac{\partial f}{\partial g} - \frac{Df}{kT} \frac{\partial \Delta G}{\partial g} \qquad (2.2.16)$$

where $D = K_g^{(f)} n_1$. The equation (2.2.16) is derived by reference (1) by letting $\Delta g \rightarrow dg$. The substitution of $\partial I / \partial g$ for $I_g - I_{g+1}$ (using (2.2.16)) into equation (2.2.14) gives:

$$\frac{\partial f}{\partial t} = -\frac{\partial I}{\partial g} = \frac{\partial}{\partial g}\left(D \frac{\partial f}{\partial g} \right) + \frac{1}{kT} \frac{\partial}{\partial g}\left(Df \frac{\partial \Delta G}{\partial g} \right) \qquad (2.2.17)$$

By differentiating and grouping like terms, (2.2.17) becomes

$$\frac{dT}{dt} \frac{\partial f}{\partial T} = A(g,T) \frac{\partial^2 f}{\partial g^2} + B(g,T) \frac{\partial f}{\partial g} + C(g,T) f \qquad (2.2.18)$$

where the temperature gradient dT/dt is assumed to be constant and will be denoted by χ and

$$A(g,T) = K_g^{(f)} \qquad (2.2.19)$$

$$B(g,T) = \frac{\partial K_g^{(f)}}{\partial g} + \frac{K_g^{(f)}}{kT} \frac{\partial \Delta G}{\partial g} \qquad (2.2.20)$$

$$C(g,T) = \frac{1}{kT}\left(K_g^{(f)} \frac{\partial^2 \Delta G}{\partial g^2} + \frac{\partial K_g^{(f)}}{\partial g} \frac{\partial \Delta G}{\partial g} \right) \qquad (2.2.21)$$

The boundary conditions are fixed at the Boltzmann equilibrium distribution, n_g, at the saturation temperature, T_∞, on the left (small g) and at zero on the right ($g \rightarrow \infty$). The initial condition is the Boltzmann equilibrium distribution, n_g, at the saturation temperature, T_∞. Equation (2.2.18) must be solved for the probability of clusters, f_g/n_1, with g molecules for the boundary and initial conditions stated above. After this is done, the calculated f_g/n_1 is then used in (2.2.16) to get the rates of formation of clusters.

In homogeneous condensation theory, the rate of formation of clusters of critical size:

$$I^0 = \frac{\alpha_c}{\rho_l}\left(\frac{p_v}{kT}\right)^2 \left(\frac{2m\sigma}{\pi}\right)^{1/2} \exp\left(-\frac{4\pi\sigma r^{*2}}{3kT}\right) \qquad (2.2.22)$$

where ρ_l is the density of the liquid and m is the mass of a molecule, has been the principal equation. This equation, which can be shown to be the stationary solution of the above equation (1), has been classically used to illustrate the rate of passage of clusters over the maximum Gibbs free energy value, in other words, the rate of the development of clusters at the critical value, g^*. Equation (2.2.22) should not be confused with the nonstationary formation rate equation (2.2.16) given above. This thesis shows that equation (2.2.16) gives a more accurate model for the rate of formation of nuclei than is given by the stationary value in equation (2.2.22). The values of the formation rate, I_g, obtained by numerically solving (2.2.16) are considered to be the corrected nucleation rates. Both formation rates, I and I^0, are compared in this analysis to

test the accuracy of equation (2.2.22).

2.3 Numerical Scheme

The equation (2.2.18) is solved first for the probability of clusters, f_g/n_1, with g molecules, per volume. The numerical analysis for this problem uses first order backward differencing on the temperature derivative and the Crank-Nicolson form on the spatial derivatives in (2.2.18) by taking the average between the j and j+1 temperature levels. This gives the following expression:

$$\chi\frac{f_i^{j+1}-f_i^j}{\Delta T}=\left(\frac{A(g,T)\left(f_{i-1}^{j+1}-2f_i^{j+1}+f_{i+1}^{j+1}\right)}{2\Delta g^2}+\frac{B(g,T)}{4\Delta g}\left(f_{i+1}^{j+1}-f_{i-1}^{j+1}\right)+\frac{C(g,T)f_i^{j+1}}{2}+\frac{d_i^j}{2}\right) \tag{2.3.1}$$

where

$$d_i^j=\frac{A(g,T)}{\Delta g^2}\left(f_{i-1}^j-2f_i^j+f_{i+1}^j\right)+\frac{B(g,T)}{2\Delta g}\left(f_{i+1}^j-f_{i-1}^j\right)+C(g,T)f_i^j \tag{2.3.2}$$

and subscript i stands for the value of g and superscript j corresponds to the temperature. Factoring out the like terms in f in (2.3.1) gives

$$\left[-\frac{\Delta T}{2\chi}\left(\frac{A(g,T)}{\Delta g^2}+\frac{B(g,T)}{2\Delta g}\right)+1\right]f_{i-1}^{j+1}+\left[\frac{\Delta T}{2\chi}\left(\frac{2A(g,T)}{\Delta g^2}-C(g,T)\right)\right]f_i^{j+1}+\left[-\frac{\Delta T}{2\chi}\left(\frac{A(g,T)}{\Delta g^2}-\frac{B(g,T)}{2\Delta g}\right)\right]f_{i+1}^{j+1}=d_i^j+f_i^j \tag{2.3.3}$$

which represents a linear, tridiagonal system of equations. At the initial condition the nonequilibrium distribution function, f_i^0, coincides with the Gibbs equilibrium distribution function, n_g. The boundary conditions are at $g_0 = 2$ where $f_0^j = n_2$, and at $g \rightarrow$

∞, $f_i^j \rightarrow 0$. Since the initial condition is not physical and overestimates the distribution of molecules, the right hand boundary is needed to force the nonequilibrium distribution to zero. The purpose of this step is to obtain the initial condition for the next stage of condensation which is not a concern in this investigation. Notice that the theory assumes that n_1 is constant, that is, that there is replacement of the monomer. This is another indication that the theory only applies at the initial stage of condensation.

The next step is to solve (2.2.16) for the nonstationary formation rate of clusters, I, by substituting the values obtained for the probability of clusters, f_g/n_1, with g molecules, into the digitized form of equation (2.2.16):

$$I_i^j = -D\left[\frac{f_{i+1}^j - f_{i-1}^j}{2\Delta g} + \frac{f_i^j}{kT}\frac{\partial \Delta G}{\partial g}\right] \tag{2.3.4}$$

where

$$\frac{\partial \Delta G}{\partial g} = 2\sigma\left(\frac{4\pi V_1^2}{3}\right)^{1/3} g^{*2/3}\left[\frac{1}{g^{1/3}g^{*2/3}} - \frac{1}{g^*}\right] \tag{2.3.5}$$

is obtained by differentiating equation (2.2.10).

Before going further, some useful information should be discussed and important variables defined. First, the equations for the number of clusters, f_g, and the Gibbs equilibrium distribution function, n_g, are both divided by the total distribution of molecules, n_1, to determine the probability density of the development of clusters with g molecules. It is important to note the coefficients in equation (2.3.3). The values at the initial saturation temperature, T_∞, for $A(g, T)\Delta T/\Delta g^2$, $B(g, T)\Delta T/2\Delta g$, and $C(g, T)$ are shown in Figure 2.3.1. The ΔT and Δg are assigned the values of 1e-5 K and 0.39,

respectively. These values were determined, by trial and error, from $A(g, T)\Delta T/\Delta g^2$ which needs to be less than or equal to one for stability. The figure shows that this term stays below one for all g molecules evaluated. The other two terms are mostly negative and cause the solutions at each temperature drop to stay below the initial condition. The Gibbs equation, ΔG, used to calculate the Gibbs equilibrium function, n_g, is obtained from (2.2.10) and requires values for the number of molecules of the critical size, g^*, and the volume of a liquid molecule which is $V_l = k/(\rho_l R_w)$ where R_w is the gas constant for water and is given by 461.52 J/(kg K) and k is the Boltzmann constant with value of 1.381e-23 J/K. The critical number, g^*, is calculated from (2.2.6). The following are empirical relations used in this analysis for the density of the liquid, ρ_l, the surface tension, σ, and the saturation pressure, p_∞ (5):

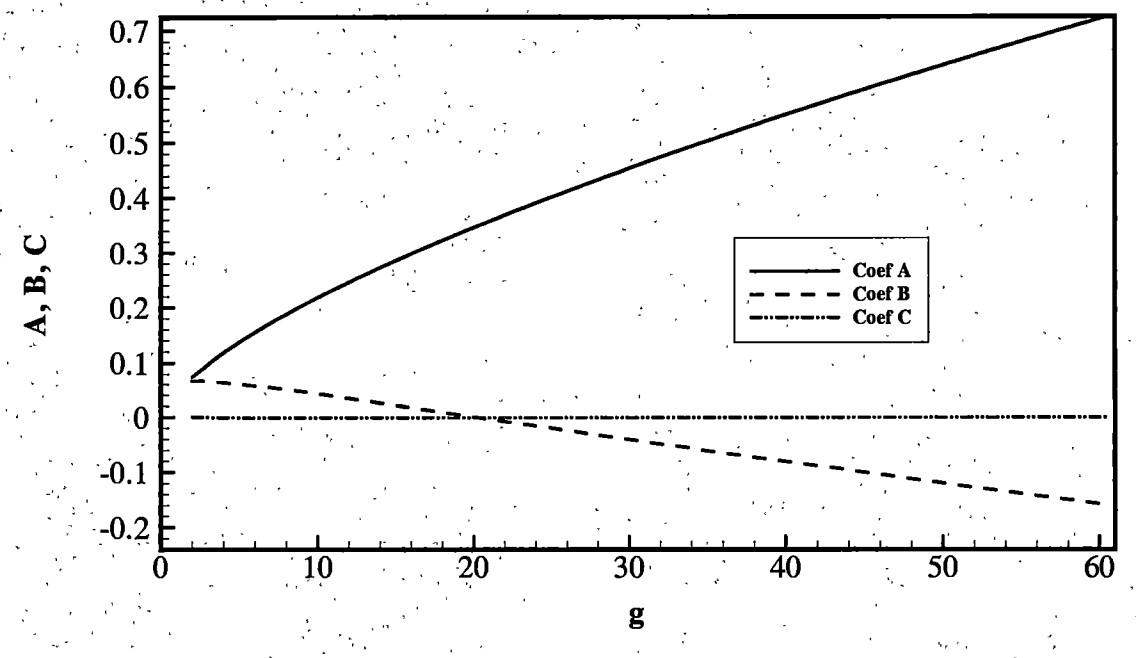

Figure 2.3.1: Plot of Coefficients $A(g, T)\Delta T/\Delta g^2$, $B(g, T)\Delta T/2\Delta g$, and $C(g, T)$ from Equations (2.2.19) - (2.2.21) at $T_\infty = 323$ K, $\chi = 1e6$ K/s, and $p_v/p_\infty = 6$

$$\rho_l = 1000.23236 - 0.01681514\times(T-273.16) - 0.004487861\times(T-273.16)^2 \text{ kg/m}^3$$

$$\sigma = 0.0757075 - 0.0001470996\times(T-273.16) - 1.08505e-7\times(T-273.16)^2 - 1.31655e-9\times(T-273.16)^3 \text{ N/m}$$

$$p_\infty = 1.01325e5\times 10^{5.4266514-2005.1/T+1.3869*10^4*x/T*\left(10^{D*x*x}-1\right)+E*\left(10^{F*z^{1.25}}\right)} \text{ Pa}$$

where

$$x = T^2 - 2.937e5$$
$$z = 647.27 - T$$
$$D = 1.1965e-11$$
$$E = -4.4e-3$$
$$F = -5.7148e-3$$

The first two relations were obtained from Collins (5) and the latter from reference (8).

The relations above are different from those used in reference (1) which are:

$$\rho_l = (1.11 - 0.004\times(T-273.16))\times 10^3 \text{ kg/m}^3$$
$$\sigma = (128 - 0.19T)\times 10^{-3} \text{ N/m}$$
$$p_\infty = \exp\left(28.26 - \frac{5338}{T}\right)/10 \text{ Pa}$$

It is believed that the reference (1) relations are not as accurate as those used for this study. For this thesis, the condensation coefficient, α_c, is constant and is taken from reference (1):

$$\alpha_c = \begin{cases} 1, & (T < 273\text{ K}) \\ \exp\left(-3.22 + \frac{875}{T}\right) & (T > 273\text{ K}) \end{cases}$$

This expression has no explanation. Other authors have assumed values that differ from the one above by many orders of magnitude.

3. RESULTS AND CONCLUSIONS

3.1 Results and Conclusions from Reference (1)

The most important part of the results is the solution for the probability distribution of clusters, f_g/n_1, and the formation rate of clusters, I_g. The results for these equations obtained from reference (1) are illustrated by Figure 3.1.1. The upper right corner of the figure shows the initial condition, n_g, and the solution for the probability of clusters, f_g/n_1, after cooling to a final temperature. Since there are no numbers, this figure is believed to be a sketch of what the solution should look like and not an actual solution. The larger portion of the figure gives the relation of I_g/I^0 with respect to ΔT. The difference, ΔT, is equal to $T - T_\infty$ where T is the saturation temperature at the vapor pressure, p_v, and T_∞ is the starting saturation temperature, corresponding to p_∞, which is taken to be 323 K. Then the supersaturation ration, $S = p_v(T)/p_\infty(T_\infty)$. The ratio of the calculated formation rate of clusters, I_g, and the stationary formation rate of clusters of critical size, I^0, is shown for two temperature gradients, 10^5 K/s and 10^6 K/s. Reference (1) calculates the stationary and nonstationary solutions for the formation rate, I_g. The stationary case assumes that $\partial f/\partial t = 0$.

These results indicate that at a smaller ΔT the solutions for the stationary formation rate is greater than the nonstationary one, I/I_H, although as ΔT increases and becomes large the difference diminishes. The stationary rates for the two temperature gradients, curve 1 and curve 2, are also compared. For small ΔT, there is a large deviation between curve 1 and curve 2, but as the ΔT increases the difference between

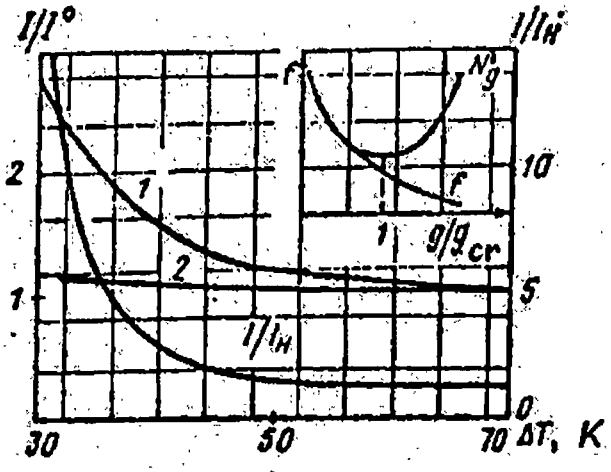

Figure 3.1.1: Plot of I_g/I^0 and I/I_H with Respect to $\Delta T = T - T_\infty$ for $\chi = 10^6$ K/s (1) and 10^5 K/s (2) where $T_\infty = 323$ K for Water Vapor from Reference (1)

the two disappears. From these observations, reference (1) makes the conclusion that, except at small ΔT, equation (2.2.22) can be used instead of solving equation (2.2.16) numerically.

3.2 Results and Conclusions for this Thesis

The initial condition, n_g, is dependent upon the Gibbs energy equation, ΔG, from (2.2.10) which is represented by Figure 3.2.1. The figure shows maximums at the critical values, g^*, for different supersaturation ratios, $S = p_v/p_\infty$. If the number of molecules, g, exceeds the critical number, g^*, the droplet grows; conversely, evaporation occurs if g drops below the critical number. Gathered from the figure, the peaks for the ΔG and g^* decrease with increasing S. It can be inferred that the larger the value of S the more likely that the droplets will form and induce condensation.

The initial condition is used in solving the tridiagonal system (2.3.3) for the

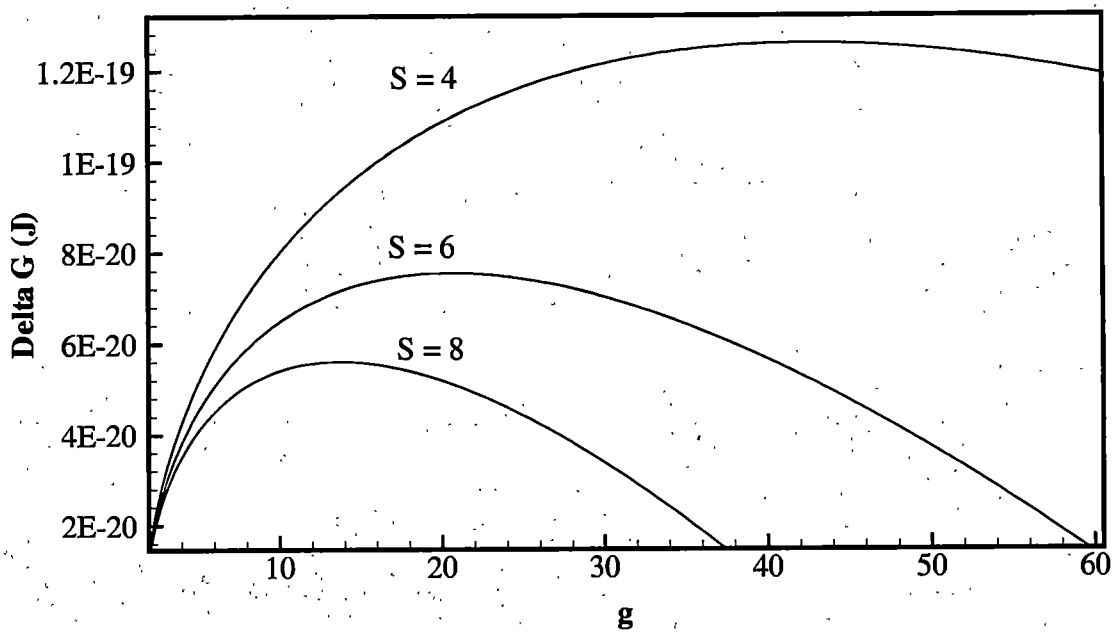

Figure 3.2.1: Gibbs Free Energy, ΔG, vs. the Number of Molecules, g, at the Saturation Temperature, T = 323 K, for Supersaturation Ratios, S = 4, 6, 8

nonequilibrium probability distribution, f_g/n_1, shown in Figures 3.2.2 - 3.2.4. At the left-hand boundary condition at $g_0 = 2$, the nonequilibrium distribution coincides with the equilibrium distribution, n_g. The right-hand boundary is chosen to be at g = 60.5 where the probability distribution, f_g/n_1, is forced to zero. For the chosen number of temperatures and grid points ($\Delta g = 0.39$ and ΔT = 1e-5), the value of 60.5 represents memory limit of the computer and software environment used to compile the Crank-Nicolson code. The cooling rate, χ, is assigned the same value as reference (1), which is 10^6 K/s. The nonequilibrium probability distribution, f_g/n_1, is calculated by cooling from the initial saturation temperature of 323 K to 322.983 K. It should be noted that reference (1) calls f the nonequilibrium distribution, but it is basically the probability of having a particle with g molecules. As seen in the figures, with a small amount of cooling the

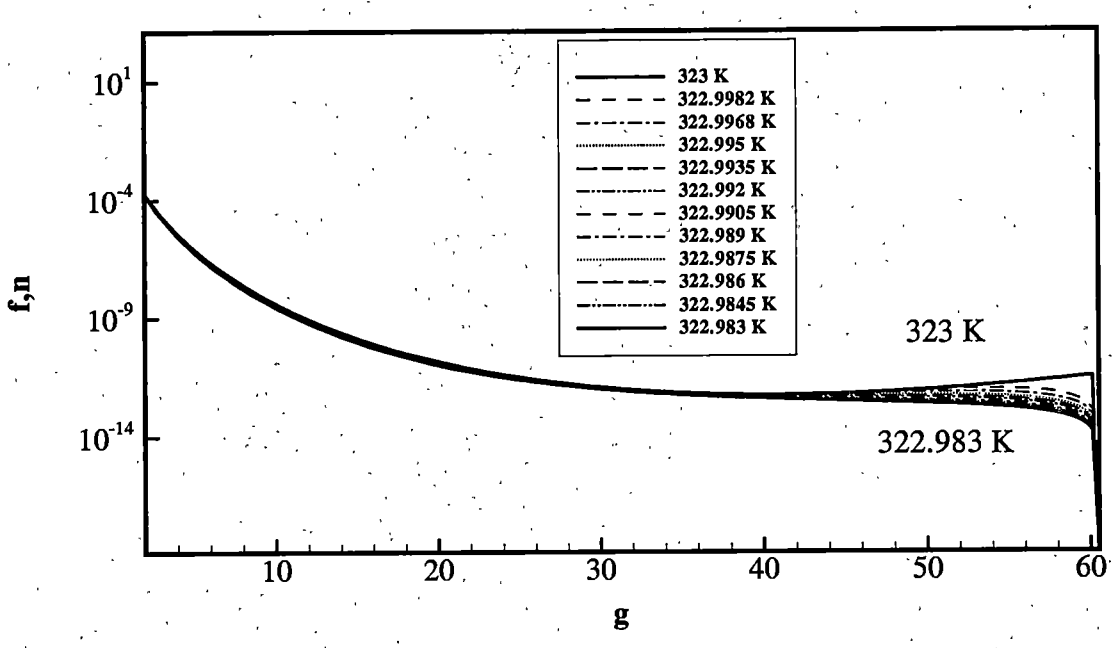

Figure 3.2.2: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 4 with Temperature Gradient of 10^6 K/s and Saturation Temperature of 323 K

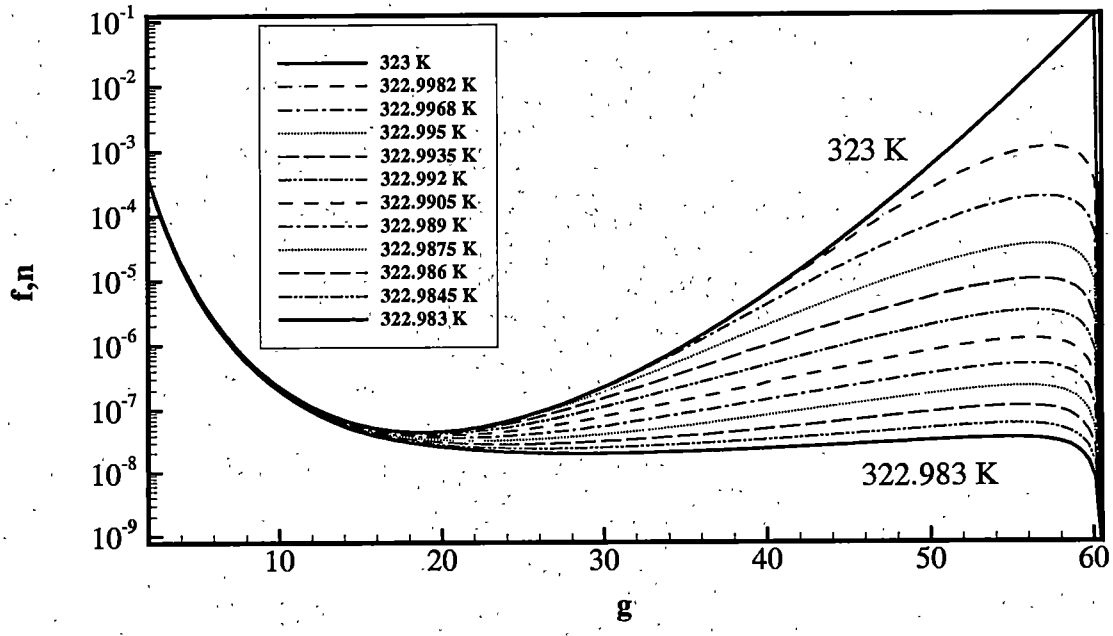

Figure 3.2.3: Probability of Clusters, f_g/n_1, and Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 6 with Temperature Gradient of 10^6 K/s and Saturation Temperature of 323 K

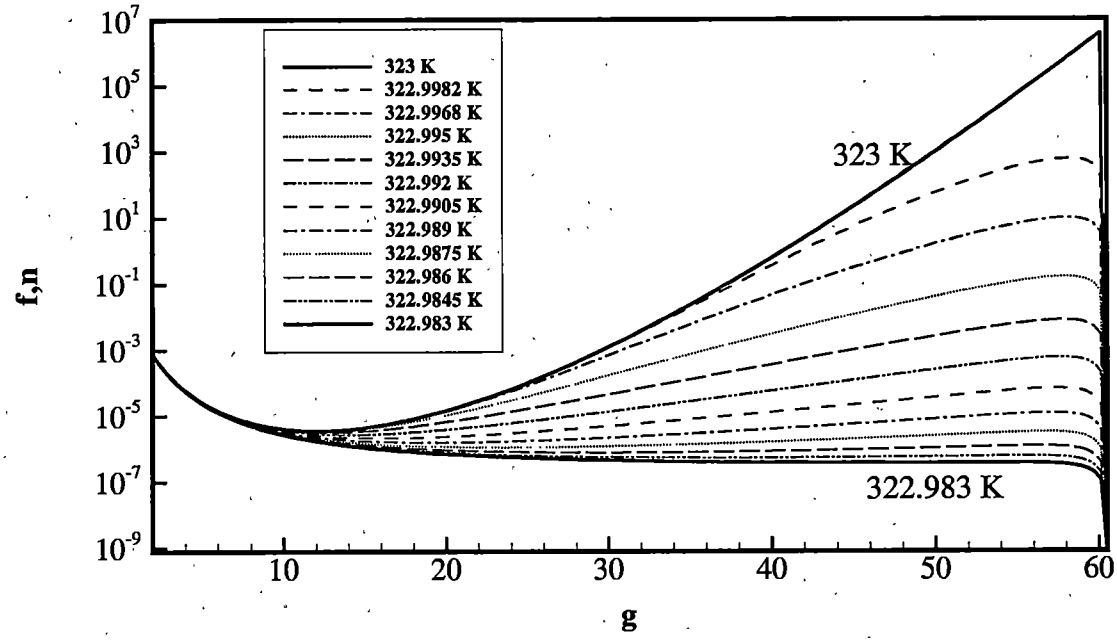

Figure 3.2.4: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 8 with Temperature Gradient of 10^6 K/s and Saturation Temperature of 323 K

probability drops by a considerable amount towards zero. With the cooling rate at 10^6 K/s, the time taken to cool 0.017 K is 17 ns which shows the speed at which the process takes place. Also, the probability of having clusters with g molecules increases as the supersaturation ratio, S, increases, since the critical value is so large at the lower values of S.

Figures 3.2.5 - 3.2.7 illustrate the effect on the probability by changing the temperature gradient to 10^5 K/s. The probability drops a considerable amount more than the 10^6 K/s temperature gradient from the equilibrium distribution. As can be seen in Figures 3.2.8 – 3.2.10, the probabilities decrease even more by decreasing the saturation temperature to 273 K. However, the same observation can be made for both changes that

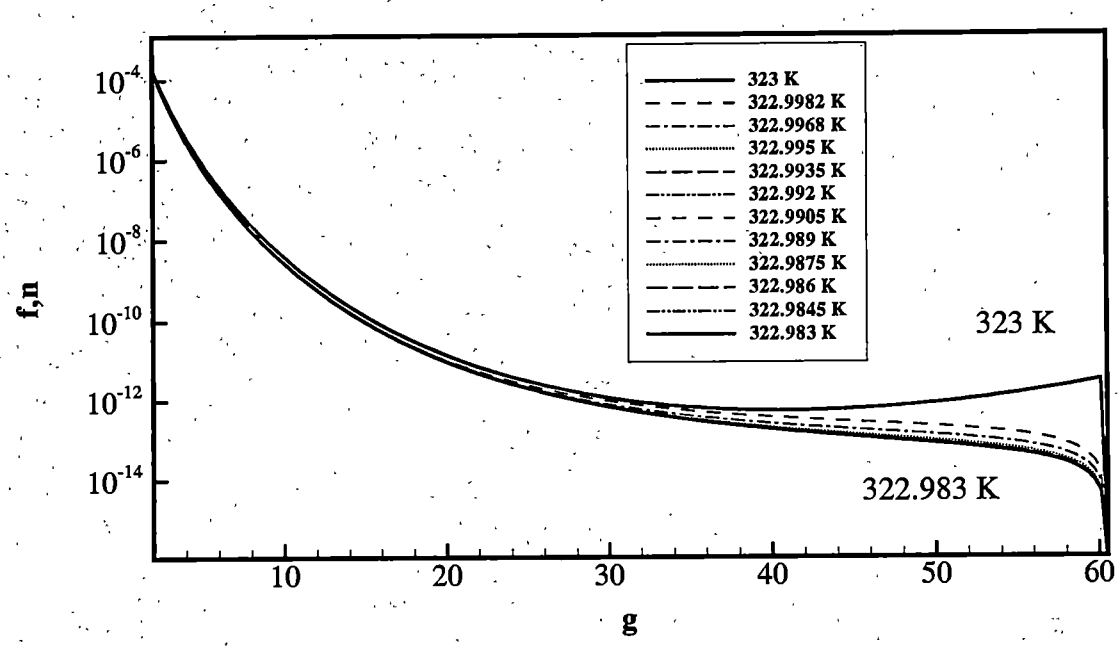

Figure 3.2.5: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 4 with Temperature Gradient of 10^5 K/s and Saturation Temperature of 323 K

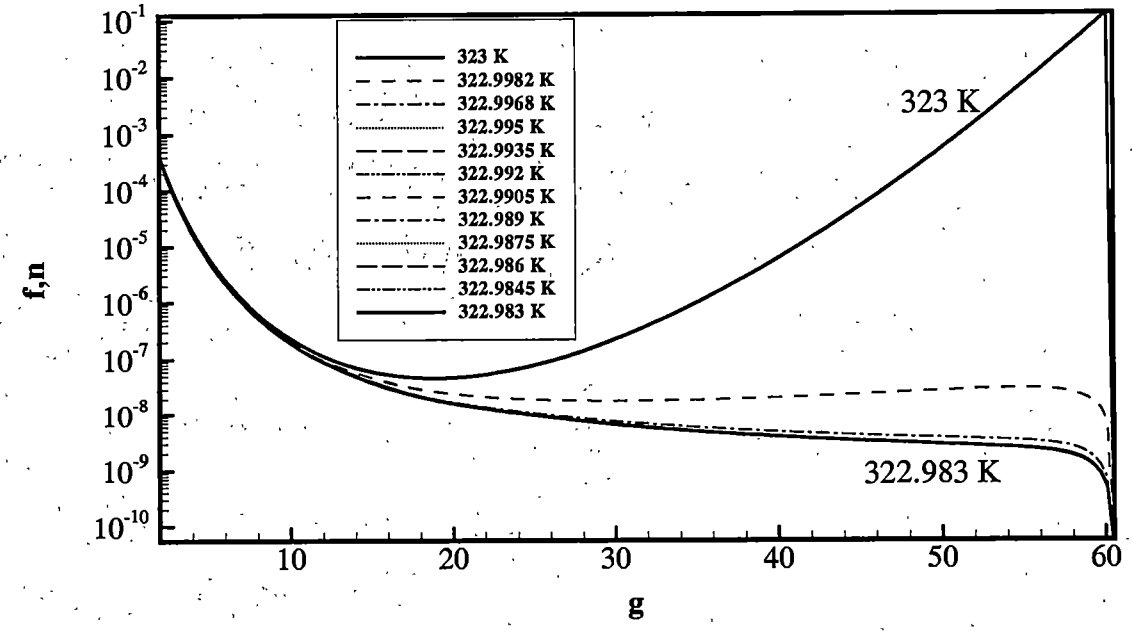

Figure 3.2.6: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 6 with Temperature Gradient of 10^5 K/s and Saturation Temperature of 323 K

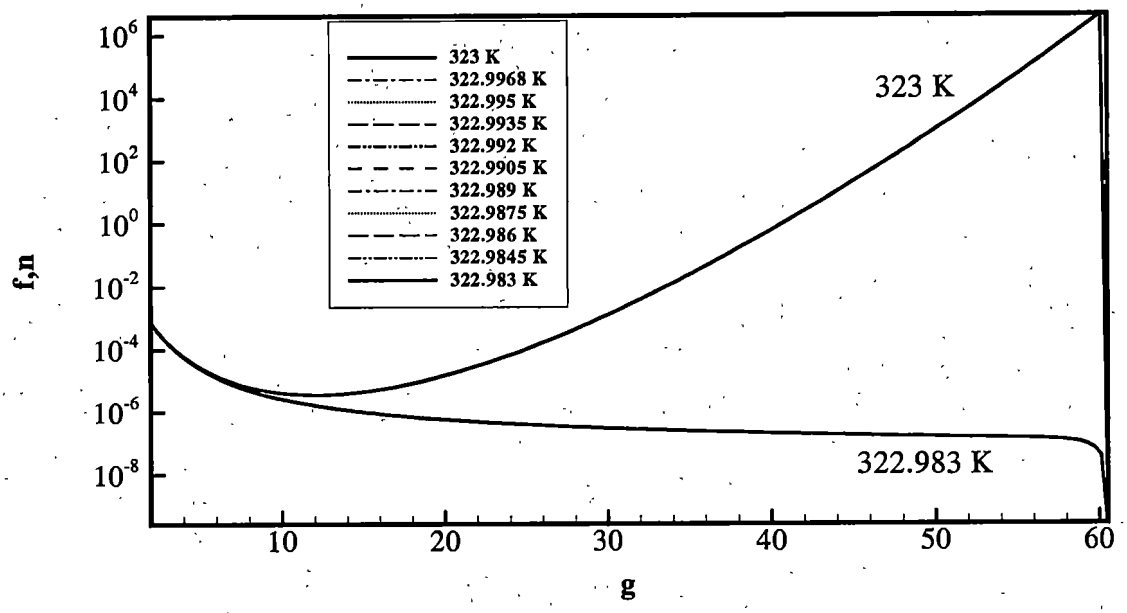

Figure 3.2.7: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 8 with Temperature Gradient of 10^5 K/s and Saturation Temperature of 323 K

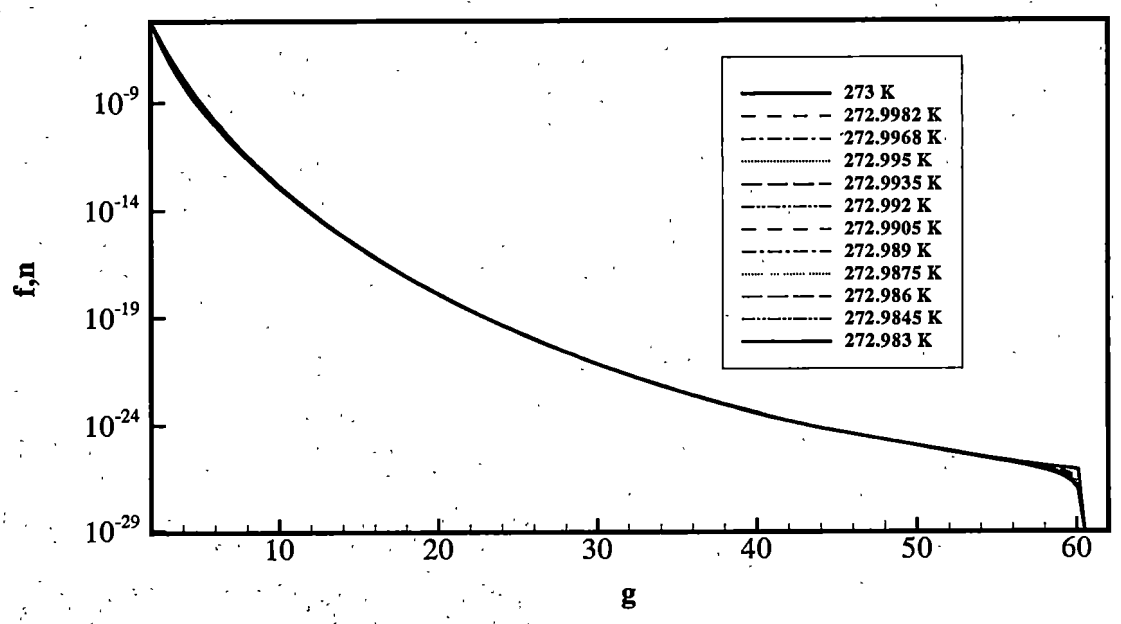

Figure 3.2.8: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 4 with Temperature Gradient of 10^6 K/s and Saturation Temperature of 273 K

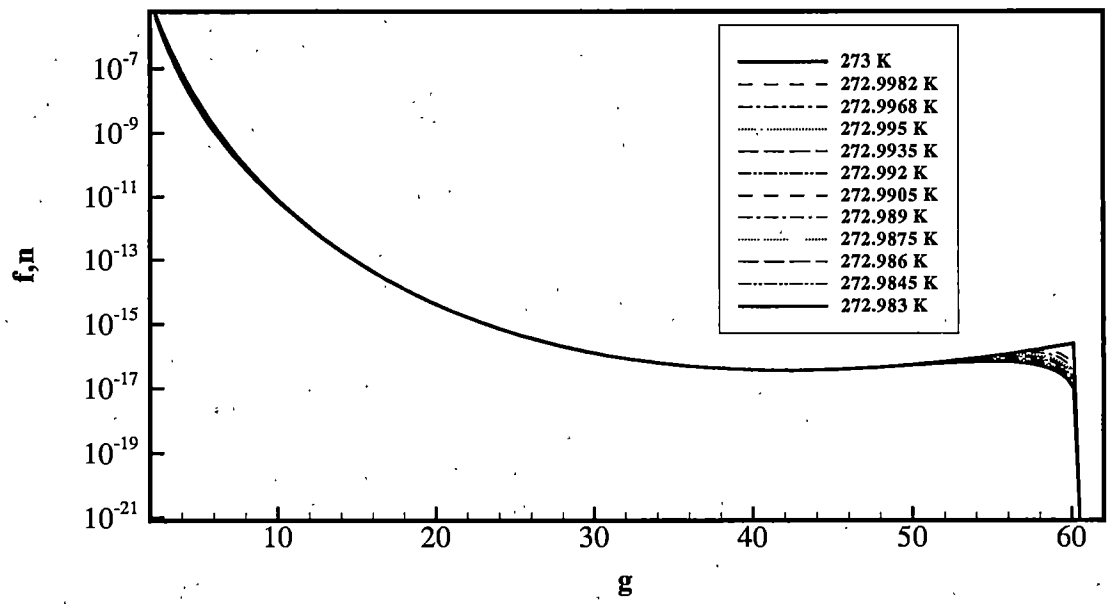

Figure 3.2.9: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 6 with Temperature Gradient of 10^6 K/s and Saturation Temperature of 273 K

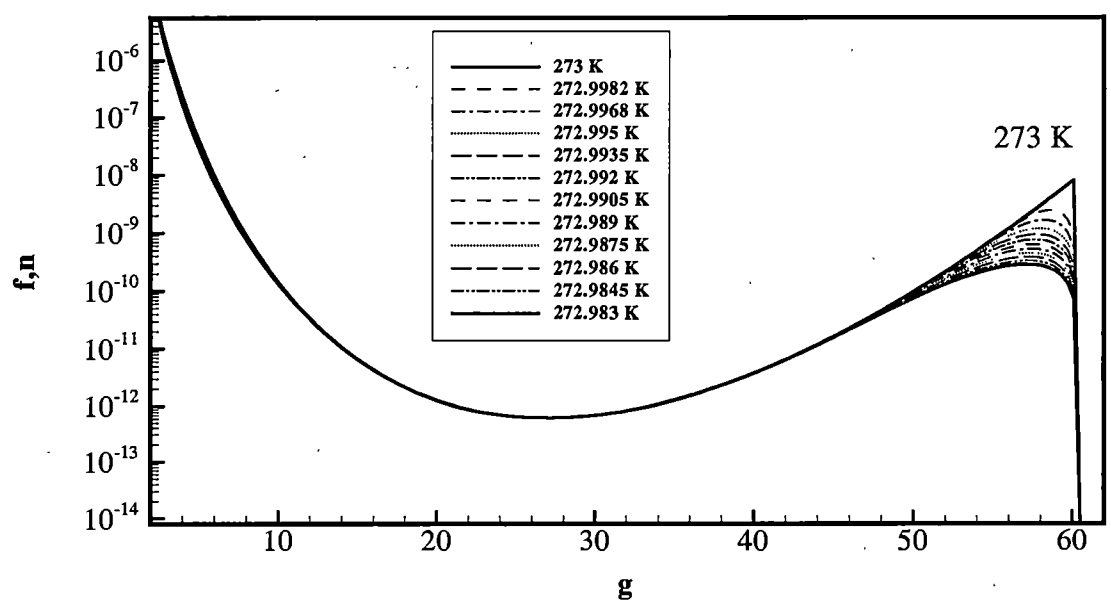

Figure 3.2.10: Probability of Clusters, f_g/n_1, and the Equilibrium Distribution, n_g, vs. the Number of Molecules, g, for S = 8 with Temperature Gradient of 10^6 K/s and Saturation Temperature of 273 K

the greater the supersaturation ratio, S, the greater the probability of having clusters with g molecules. It can also be inferred that at greater saturation temperatures the nucleation stage occurs more rapidly.

The values at the critical number, g^*, are 40.7515 for the solutions in Figures 3.2.2 and 3.2.5, 18.8743 in Figures 3.2.3 and 3.2.6, 12.0745 in Figures 3.2.4 and 3.2.7, and in Figures 3.2.8 – 3.2.10, the values are much larger. For the number of molecules, g, less than g^* the probability, f_g/n_1, closely follows the equilibrium distribution. However, as the number of molecules, g, increases, the nonequilibrium distribution, f_g/n_1, diverges from the initial condition. This is caused by the evaporation of clusters possessing g greater than g^*. However, the theory gives the rate of passage of clusters over the ΔG hill. This information can be used as the initial condition for the next stage of condensation.

The rate of formation of clusters, I_g, containing g molecules is calculated from (2.3.4) by using the solution for the probability of clusters, f_g/n_1. Figure 3.2.11 shows a plot of the ratio of formation rates, I_g/I^0, for ΔT at the corresponding critical number and for two temperature gradients: curve 1 at 10^5 K/s and curve 2 at 10^6 K/s. Both of the curves calculate the nonstationary solution. The first observation from the figure is that as ΔT increases the ratio of I_g/I^0 decreases. The results show that at small ΔT, the values of curve 2 are larger than those of curve 1, though as ΔT increases the difference between the two diminishes. Also, both curves have a negative slope. The magnitude of the initial slope of curve 2 is greater than that of curve 1. As ΔT increases the magnitudes of the slopes of both curves decrease. The plot suggests that the higher the

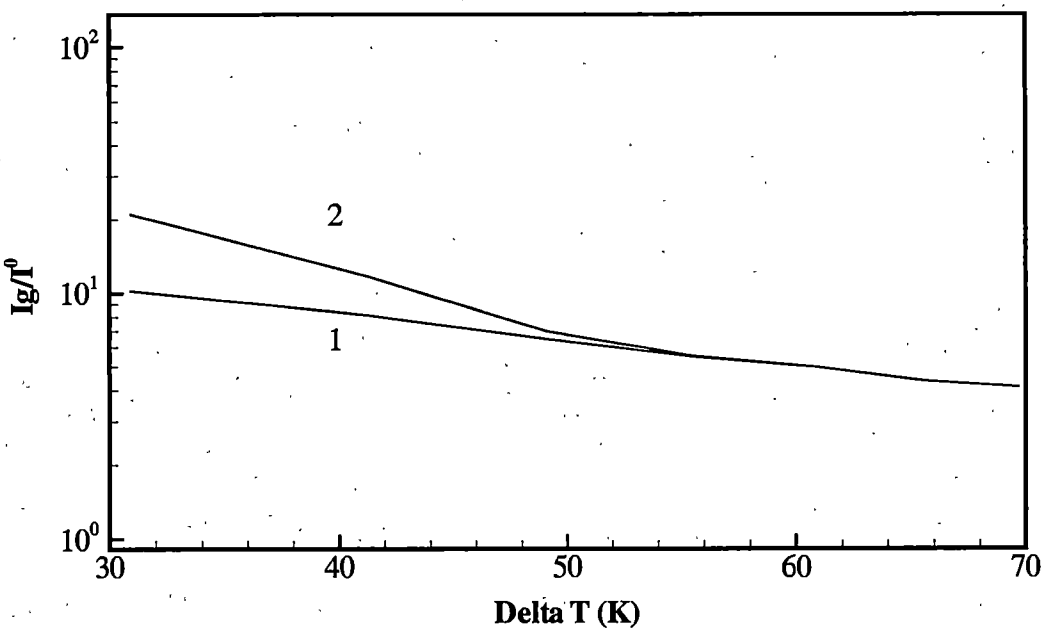

Figure 3.2.11: Ratios of Formation Rates, I_g/I^0, with Temperature Gradients of (1) 10^6 K/s and (2) 10^5 K/s at S = 8, Saturation Temperature of 323 K

cooling rate the larger the difference between the two formation rates, I_g and I^0, up to a certain ΔT where the two curves merge.

There is a direct correlation between ΔT and the vapor pressure, p_∞, such that as ΔT increases the vapor pressure increases above the equilibrium vapor pressure for the saturation temperature, T_∞. Keeping this in mind, Figure 3.2.11 indicates that as the vapor pressure decreases the formation rates increase which occurs as a result of the increase in Gibbs energy, ΔG.

3.3 Comparison of Results

The purpose of this thesis is to compare the results from the numerical scheme above with that of reference (1). The derivations and equations used are the same as

those in reference (1) except for the expressions for density, surface tension, and vapor pressure. The vapor pressure from reference (1) has about a 2% difference when compared to steam table values (Van Wylen 9) while for the equation used in this study the difference is 0.2%. The difference is calculated in the following way: $(p_{table} - p_{\infty})/p_{table}$, where p_{table} is the saturation pressure listed in the steam tables at the desired temperature. The temperature unit conversion is important in deciding the percent difference. The steam tables record the temperature in Celsius while all temperatures used in this thesis are in Kelvin. The conversion to Kelvin is T + 273.16. If the 0.16 is not added to the temperature, the percent differences are the same for both studies. Figure 3.3.1 shows the contrast between the two pressure calculations. For the densities, ρ, and the surface tensions, σ, there is not a significant difference between the reference (1) values and those used in this thesis.

Both studies computed the probability of clusters, f_g/n_1, and the formation rates, I_g, of these clusters. The probability distributions from this study seem to be analogous with those in reference (1) since both curves demonstrate the drop towards zero at the right boundary. For this investigation, only the nonstationary rates are calculated. The rates for this thesis are higher than those for the reference. This variation might be due to differences in values of Δg and ΔT used in the numerical solution. It can be concluded from both studies that the classical formation rate equation (2.2.22) is not useful at lower ΔT and that the numerical studies are more accurate.

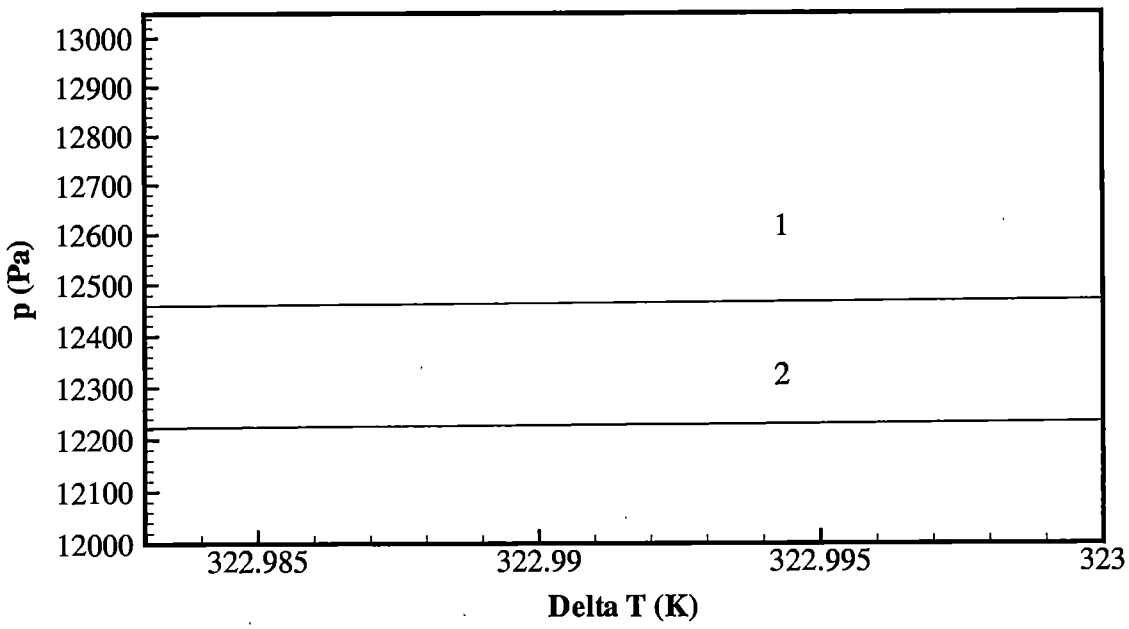

Figure 3.3.1: Plot of Pressures from Reference (1) and for the Thesis Example

LIST OF REFERENCES

LIST OF REFERENCES

(1) Ryzhov, Y. A., U. G. Pirumov, and V. N. Gorbunov, Nonequilibrium Condensation in High-Speed Gas Flows, Gordon and Breach, New York, 1974.

(2) Asanuma, Tsuyoshi, "Flow Visualization Techniques in Japan," International Symposium on Flow Visualization, Tokyo, Japan, October 1977.

(3) Mueller, T. J., "The Role of Flow Visualization in the Study of Afterbody and Base Flows," *Experiments in Fluids*, Volume 3, 1985, 61-70.

(4) Settles, Gary S., "Flow Visualization Techniques for Practical Aerodynamic Testing," Third International Symposium on Flow Visualization, September 1983, University of Michigan, Ann Arbor, Michigan, 306-315.

(5) Collins, F. G., "Development of Laser Vapor Screen Flow Visualization System for 16T," AEDC TMR-96-E8, September 1996.

(6) Fowler, A. C., Mathematical Models in the Applied Sciences, Cambridge University Press, Cambridge, United Kingdom, 1997.

(7) McDonald, J. E., "Homogeneous Nucleation of Vapor Condensation. II. Kinetic Aspects," Am. J. Phys., Vol. 31, No. 1, pp. 31-41, 1962.

(8) "Properties of Ordinary Water-Substance in all its Phases: Water Vapor, Water, and all the Ices," Compiled by N. Ernest Dorsey, NBS, Washington DC, American Chemical Society Monograph Series (Facsimile of 1940 Edition), Hafner Publishing Company, New York, 1968.

(9) Van Wylen, Gordon J. and Richard Sonntag, Fundamentals of Classical Thermodynamics, John Wiley and Sons, Inc., New York, 1986.

APPENDIX

```
/*************************************************************************/
/*************************************************************************/
/*  Program          :  Condensation                                     */
/*                                                                       */
/*  Programmer       :  Jessica Thomas                                   */
/*                                                                       */
/*  Date             :  July 5, 1999                                     */
/*                                                                       */
/*  Description      :  This ANSI C program calculates the number of clusters of  */
/*                      g per unit volume and the rates of formation of condensation */
/*                      nuclei                                           */
/*                                                                       */
/*************************************************************************/
/*************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define iMax 501          /* max allowable # of rows and columns */
#define jMax 1701

double tol = 1e-5;
double inf = 1e+15;

double dg, dT, alphc, m, k, Rw,S;
double D, E, F, Xi;
double f[iMax][jMax], g[iMax], Inew[iMax][jMax];
double ginit, gmax, Tinit;
int NI, NJ;
char outFileStr[100];


/*************************************************************************/
/************************** PROCEDURES ***********************************/
/*************************************************************************/
/*************************************************************************/

void initparam (void)

/* initial parameters */

{
FILE *fp;
```

```
  if ((fp = fopen("input.dat", "r")) == NULL) {
     printf("cannot open file");
     exit(1);
  }

//  NI=iMax-1;
//  NJ=jMax-1;

  fscanf(fp, "%i", &NI);
  fscanf(fp, "%i", &NJ);
  fscanf(fp, "%lf", &ginit);
  fscanf(fp, "%lf", &gmax);
  fscanf(fp, "%le", &dg);
  fscanf(fp, "%lf", &dT);
  fscanf(fp, "%lf", &Xi);
  fscanf(fp, "%lf", &Tinit);
  fscanf(fp, "%lf", &alphc);
  fscanf(fp, "%le", &m);
  fscanf(fp, "%le", &k);
  fscanf(fp, "%lf", &Rw);
  fscanf(fp, "%lf", &S);
  fscanf(fp, "%le", &D);
  fscanf(fp, "%le", &E);
  fscanf(fp, "%le", &F);
  fscanf(fp, "%s", &outFileStr);

  fclose(fp);
}

/*************************************************************************/

double sig(double atc)

{
double functval;

  functval = 0.0757075-0.0001470996*atc-(1.08505e-7)*pow(atc,2)-
        (1.31655e-9)*pow(atc,3);

  return functval;
}

/*************************************************************************/

double rhol(double atc)
```

```
{

double functval;

   functval = 1000.23236-0.01681514*atc-0.004487861*pow(atc,2);

   return functval;

}

/*************************************************************************/

double pv(double aT)

{
double x, z, ps;
double functval;

   x = aT*aT-2.937e5;
   z = 647.27-aT;
   ps = 1.01325*pow(10.0,5.0)*pow(10.0,(5.4266514-2005.1/aT+(1.3869e-4)*x/aT*
      (pow(10.0,D*x*x)-1.0)+E*pow(10.0,F*pow(z,1.25))));

   functval=ps*S;

   return functval;
}

/*************************************************************************/

double Vl(double atc)

{
double functval;

   functval = k/(rhol(atc)*Rw);

   return functval;
}


/*************************************************************************/

double gstar(double aT)
```

```
{
double atc, functval;

   atc = aT - 273.16;
   functval = 32.0*M_PI*pow(sig(atc),3.0)/(3.0*k*pow(Rw*rhol(atc),2.0)*
         pow(aT*log(S),3.0));

   return functval;
}

/*************************************************************************/

double Kg(double aT, double ag)

{
double atc, aVl, functval;

   atc = aT - 273.16;
   aVl = Vl(atc);
   functval = 4.0*M_PI*alphc*pow(3.0*ag*aVl/4.0/M_PI,2.0/3.0)*pv(aT)/
                              pow(2.0*M_PI*m*k*aT,1.0/2.0);

   return functval;
}

/*************************************************************************/

double partialKg(double aT, double ag)

{
double atc, aVl, functval;

   atc = aT - 273.16;
   aVl = Vl(atc);
   functval = 8.0*M_PI/3.0*alphc*pow(3.0*aVl/4.0/M_PI,2.0/3.0)*
                              pow(ag,-1.0/3.0)*pv(aT)/pow(2.0*M_PI*m*k*aT,1.0/2.0);

   return functval;
}

/*************************************************************************/

double delGg(double aT, double ag)
```

```
{
double atc, aVl, functval;

   atc = aT - 273.16;
   aVl = Vl(atc);
   functval = sig(atc)*pow(4.0/3.0*M_PI*aVl*aVl,1.0/3.0)*
         (3.0*pow(ag,2.0/3.0)-2.0*ag/pow(gstar(aT),1.0/3.0));

   return functval;
}

/*************************************************************************/

double partialdelGg(double aT, double ag)

{
double atc, aVl, functval;

   atc = aT - 273.16;
   aVl = Vl(atc);
   functval = sig(atc)*pow(4.0/3.0*M_PI*aVl*aVl,1.0/3.0)*
         (2.0*pow(ag,-1.0/3.0)-2.0*pow(gstar(aT),-1.0/3.0));

   return functval;
}

/*************************************************************************/

double doubpartdelGg(double aT, double ag)

{
double atc, aVl, functval;

   atc = aT - 273.16;
   aVl = Vl(atc);
   functval = sig(atc)*pow(4.0/3.0*M_PI*aVl*aVl,1.0/3.0)*
         (-2.0/3.0*pow(ag,-4.0/3.0));

   return functval;
}


/*************************************************************************/

double A(double aT, double ag)
```

```
{
double functval;

   functval=Kg(aT, ag);

   return functval;
}

/*************************************************************************/

double B(double aT, double ag)

{
double functval;

   functval = partialKg(aT,ag)+partialdelGg(aT,ag)*Kg(aT,ag)/k/aT;

   return functval;
}

/*************************************************************************/

double C(double aT, double ag)

{
double functval;

   functval = (Kg(aT,ag)*doubpartdelGg(aT,ag)+partialKg(aT,ag)*
                                    partialdelGg(aT,ag))/k/aT;

   return functval;
}

/*************************************************************************/

double ng(double aT, double ag)

{
double functval;

   functval=exp(-delGg(aT,ag)/(k*aT));

   return functval;
}
```

```
/*************************************************************************/

void conditions (void)

/* initial and boundary conditions */

{
//double T;
int i, j;

   g[0] = ginit;

   for (j = 0; j <= NJ; j++) {
//      T = Tinit + j*dT;
      f[0][j]=ng(Tinit,g[0]);
      g[NI] = gmax;
      f[NI][j]=0.0;
   }

//   f[NI][0]=ng(Tinit,g[NI]);

   for(i=1; i<NI; i++){
      g[i]=g[i-1]+dg;
      f[i][0]=ng(Tinit,g[i]);
   }
}

/*************************************************************************/

void TriDSolver(double argP[iMax], double argQ[iMax], double argR[iMax],
          double argRHS[iMax], double argx[iMax], int argn)

/* This subroutine solves a tri-diagonal system given     */
/* the upper, middle, and lower diagonals. The solution   */
/* is returned in the array argx.                         */

{
double alph[iMax], SOL[iMax];
int i;

   alph[1] = argQ[1];
   SOL[1] = argRHS[1];

   for(i = 2; i <= (argn-1); i++) {
```

```
      alph[i] = argQ[i] - argP[i]*argR[i-1]/alph[i-1];
      SOL[i] = argRHS[i] - argP[i]*SOL[i-1]/alph[i-1];
   }

   argx[argn-1] = SOL[argn-1]/alph[argn-1];
   for(i = (argn-2); i >= 1; i--) {
      argx[i] = (SOL[i] - argR[i]*argx[i+1])/alph[i];
   }
}

/*************************************************************************/

void solve (void)

{
double Told, Tnew, dg2, dTXi;
double pTnew[iMax], qTnew[iMax], rTnew[iMax];
double pTold[iMax], qTold[iMax], rTold[iMax];
double RHS[iMax], xs[iMax], RHSTerm;
int i, j;

   dg2 = dg*dg;
   dTXi = dT/Xi;
   Told = Tinit;
   for(j=0; j<NJ; j++) {
      Tnew = Tinit+(j+1)*dT;
      for(i=1; i<NI; i++) {
         pTnew[i] = dTXi*(A(Tnew,g[i])/(2.0*dg2) -
                B(Tnew,g[i])/(4.0*dg));
         qTnew[i] = dTXi*(-A(Tnew,g[i])/dg2 +
                C(Tnew,g[i])/2.0) - 1.0;
         rTnew[i] = dTXi*(A(Tnew,g[i])/(2.0*dg2) +
                B(Tnew,g[i])/(4.0*dg));

         pTold[i] = -dTXi*(A(Told,g[i])/(2.0*dg2) -
                B(Told,g[i])/(4.0*dg));
         qTold[i] = -dTXi*(-A(Told,g[i])/dg2 +
                C(Told,g[i])/2.0) - 1.0;
         rTold[i] = -dTXi*(A(Told,g[i])/(2.0*dg2) +
                B(Told,g[i])/(4.0*dg));

         if (i == 1) {
            RHSTerm = -pTnew[i]*f[i-1][j];
         }
         else if (i == (NI-1)) {
```

```
            RHSTerm = -rTnew[i]*f[i+1][j];
        }
        else {
            RHSTerm = 0.0;
        }

        RHS[i] = (pTold[i]*f[i-1][j] + qTold[i]*f[i][j] +
                rTold[i]*f[i+1][j] + RHSTerm);
    }

    TriDSolver(pTnew, qTnew, rTnew, RHS, xs, NI);

    for(i=1; i<NI; i++){
        f[i][j+1]=xs[i];
                Inew[i][j+1]=-Kg(Tnew,g[i])*((f[i+1][j+1]-f[i-1][j+1])
                                        /2.0/dg+f[i][j+1]*partialdelGg(Tnew,g[i])/k/Tnew);
    }


    Told = Tnew;
  }
}

/*************************************************************************/

double I0 (double aT)

{
double functval, atc;

  atc = aT - 273.16;

        functval = alphc*pv(aT)/k/aT*pow(2.0*m*sig(atc)/M_PI,1.0/2.0)/
                                rhol(atc)*exp(-
16.0*M_PI/3.0*pow(sig(atc)/k/aT,3.0)*pow(Vl(atc)/
          log(S),2.0));

return functval;
}

/*************************************************************************/

void printdataf (void)

{
```

```
double T;
int i, j;
FILE *fp;

   if ((fp = fopen(outFileStr, "w")) == NULL)
   {  /* begin if */
      printf("cannot open file");
      exit(1);
   }  /* end if */

   fprintf(fp,"TITLE=Solution of f at Tsat=323\n");
   fprintf(fp,"VARIABLES=\n");
   fprintf(fp,"g\n");

   for(j=0; j<=NJ; j=j+25) {
      T = Tinit + j*dT;
      fprintf(fp,"f%.6le\n", T);
   }

   for(i=0; i<=NI; i++) {
      fprintf(fp,"%.5lf ", g[i]);
      for(j=0; j<=NJ; j=j+25) {
         fprintf(fp,"%.15le ", f[i][j]);
      }
      fprintf(fp,"\n");
   }

   fclose(fp);
}

/*************************************************************************/

void printdataI (void)

{
double T;
int i, j;
FILE *fp;

   if ((fp = fopen(outFileStr, "w")) == NULL)
   {  /* begin if */
      printf("cannot open file");
      getchar();
      exit(1);
   }  /* end if */
```

```
    fprintf(fp,"TITLE=Solution of I/I0 at Tsat=323\n");
    fprintf(fp,"VARIABLES=\n");
    fprintf(fp,"T\n");

    for(i=0; i<=NI; i++) {
        if((g[i]>=14.0)&&(g[i]<=15.0))
      fprintf(fp,"I%.6le\n", g[i]);
    }

    for(j=0; j<=NJ; j++) {
        T=Tinit+j*dT;
        fprintf(fp,"%.5lf ", T);
        for(i=0; i<=NI; i++) {
                if((g[i]>=14.0)&&(g[i]<=15.0))
                fprintf(fp,"%.15le ", Inew[i][j]/I0(T));
      }
      fprintf(fp,"\n");
    }

    fclose(fp);
}

/*************************************************************************/
/************************** MAIN PROGRAM *************************/
/*************************************************************************/

void main(void)
{
    printf("\n");
    printf("Reading input.dat...\n");
    printf("\n");
    initparam();    /* call initial parameters */

    printf("\n");
    printf("Setting up initial conditions...\n");
    printf("\n");
    conditions();   /* call initial and boundary conditions subroutine */

    printf("\n");
    printf("Solving system over temperature range...\n");
    printf("\n");
    solve();        /* call solve routine */

    printf("\n");
```

```
    printf("Writing to output data file...\n");
    printf("\n");
    printdataf();    /* print solutions */

    printf("\n");
    printf("Press enter to exit program.\n");
    getchar();
}
```

VITA

Jessica L. Gass was born December 28, 1972 in Fayetteville, Tennessee. She grew up in Cowan, Tennessee and attended Franklin County High School where she graduated in 1991. In 1990, Jessica started working at The University of Tennessee Space Institute for the Center for Laser Applications in Tullahoma and remained there until graduating from Middle Tennessee State University in Murfreesboro in May 1996 with a Bachelor of Science degree in Applied Mathematics. Since she liked working at The University of Tennessee Space Institute, she decided to pursue a Master's degree there. She was appointed a Graduate Research Assistantship to Dr. Frank Collins with whom she did research in condensation theory. Jessica received her Master of Science degree in August of 1999. She accepted a job with Lockheed Martin Astronautics as a Software Engineer and moved with her husband of three years to Denver, Colorado in September 1999.