UNIVERSITY OF CALIFORNIA, SAN DIEGO

Free-Jet Expansion of Supercritical CO_2

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Imane Ghaleb Khalil

Committee in charge:

Professor David R. Miller, Chair
Professor Robert Continetti
Professor Keiko Nomura
Professor Kalyanasundaram Seshadri
Professor John Weare

2003

The dissertation of Imane Khalil is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2003

Table of Contents

List of Figures and Tables

VITA

1991-1994	B.S. Mechanical Engineering University of California, San Diego
1994-1996	M.S. Mechanical Engineering University of California, San Diego
1996-1997	Design Engineer, Composite Optics Inc. Design of composite lightweight structures used in aerospace industry. Testing of composite structures
1997-2001	Analysis Engineer, Hamilton Sundstrand Conduct structural analyses of rotating and static components for small gas turbine engines. Conduct engine tests in support of development activities
2001-2003	Ph.D. Mechanical Engineering University of California, San Diego

PUBLICATION

"The Interaction of a Supercritical Fluid Free-Jet Expansion with a Flat Surface," I. Khalil and D. Miller, Proc. of the 6th Int. Symp. on Supercritical Fluids, Vol. 1, page 655, Versailles, France, 2003.

"The Structure of Supercritical Fluid Free-Jet Expansion," I. Khalil and D. Miller, submitted to the AIChE Journal, 2003.

ABSTRACT OF THE DISSERTATION

Free-Jet Expansion of Supercritical CO_2

by

Imane Ghaleb Khalil

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2003

Professor David R. Miller, Chair

Numerical and experimental analyses of the structure of supersonic free-jet expansions of supercritical CO_2 into the atmosphere and impacting on a flat plate are presented. Such expansions are important to technology used by chemical engineers in materials processing, known as the Rapid Expansion of Supercritical Solutions (RESS).

Numerical calculations for the axisymmetric, two-dimensional expansion are based on a time-dependent finite difference method known as the two-step predictor-corrector Lax-Wendroff technique, incorporating the Redlich-Kwong or Peng-Robinson equations of state to model supercritical CO_2. A quasi-one-dimensional convergent-divergent nozzle approximation is also studied to test our numerical methods for the axisymmetric free-jet, to study the vibrational relaxation of CO_2, and to study the clustering and condensation during the expansion.

Experimental mass flow rates for the free-jet expansion from two different types of sources (orifice and capillary) are studied and compared to one-dimensional numerical results. Optical shadowgraph measurements of the axisymmetric free-jet and associated shock wave structure are compared to the axisymmetric numerical results. Impact pressure and temperature along the flat plate are measured and compared to numerical results, as is a temperature probe of the free-jet expansion.

We find that the numerical methods are able to reproduce the flow rate and free-jet structure very well. The temperature and pressure profiles agree reasonably well, except for temperature along the impacted plate. The disagreement with temperature at the plate is due to neglect of heat transfer at the plate and, at higher pressures, to condensation in the expansion. Our results suggest that approximations based on ideal gases and quasi-one-dimensional flow analysis, often used by RESS researchers, may be useful for these supercritical fluid flows. The preliminary calculations and experiments indicate that condensation is important in these flows at high pressures, but it will require considerable further study to understand and predict the associated effects.

CHAPTER 1

INTRODUCTION AND BACKGROUND

A supersonic free-jet expansion occurs in the expansion of a compressible fluid from high pressure into the atmosphere or a vacuum through a restriction such as an orifice or capillary tube. The expansion begins from a stagnation reservoir where the gas velocity is assumed to be zero. As the fluid flows through the orifice or capillary tube it accelerates from subsonic flow, reaching sonic conditions (Mach number equal to one) at the exit. The fluid then expands into a highly compressible, two-dimensional, supersonic free-jet that is bounded by shock waves which occur because the fluid pressure is adjusting to ambient conditions. The inset to figure 1.1 shows a general picture of the flow field in a free-jet expansion (Miller 1988). Because the expansion is not confined by diverging nozzle walls, as found in rocket nozzles, the expansion is called a free-jet. As the gas expands in the supersonic region, the pressure, temperature, and density decrease rapidly, while the velocity increases. Several rigorous numerical calculations of axisymmetric free-jet supersonic expansions for ideal gases have been used in molecular beam research since the 1960s (Miller 1988). The primary objective of this thesis is to extend the free-jet analysis using non-ideal gas assumptions for the expansions of fluids in a supercritical state under stagnation conditions, an important technological area referred to as Rapid Expansion of Supercritical Solutions (RESS).

Supercritical fluids are very important topics in chemistry and chemical engineering because they are often the processing medium in the regime intermediate between gases and liquids where they exhibit liquid-like solubilities and gas-like diffusivities and viscosity. The high characteristic density makes it possible for supercritical fluids to dissolve compounds that would typically be relatively insoluble under sub-critical or ideal gas conditions (Levit *et al.* 2002), or to dissolve temperature sensitive materials at low temperatures.

The supercritical fluid exists as a single phase beyond the critical point of any substance. Figure 1.2 is a phase diagram for carbon dioxide, one of the most prevalent and useful supercritical solvents; it is the fluid we study in this thesis. The critical temperature for CO_2 is 31.1°C and its critical pressure is 73 bar. These are mild critical conditions, which is why CO_2 is widely used in supercritical fluid applications. In addition, carbon dioxide is inexpensive, leaves no toxic residue, and is non-flammable (Clifford 1999). One of the earliest applications of supercritical CO_2 as a solvent was to decaffeinate coffee, while today it is used for such diverse applications as proteins encapsulation for drug delivery (Mishima *et al.* 2000). The second most widely used supercritical fluid is water with a critical temperature of 374°C and a critical pressure of 221 bar. Supercritical water has been used in toxic waste cleanup in combination with combustion of hydrocarbons (Tester *et al.* 1993; Maharrey and Miller 2001), as well as to dissolve and grow ceramic alumina thin films in a RESS apparatus (Brand and Miller 1989). There are several texts now available which review the properties and wide array of applications of supercritical fluids (McHugh and Krukonis 1986; Clifford 1999; Arai 2001; Sun 2002).

Supercritical fluids were found to dissolve low-vapor pressure solids more than a century ago by Hanney and Hogarth (1879). Solubility in supercritical fluids depends on two main factors: vapor pressure of the solute and the solvating effect of the fluid, both which depend on density. As an example, the behavior of the solubility of naphthalene in CO_2 as a function of pressure at constant temperature is shown in figure 1.3, where our calculations were made using the Peng-Robinson equation of state (defined below) and the experimental data were taken from Tsekhanskaya *et al.* (1964). The solubility rises sharply near the critical point of the solvent due to solvation arising from attractive forces between the solvent and the solute molecules. The sensitivity of the solubility with both temperature and pressure is shown more vividly in figure 1.4, also calculated with the Peng-Robinson equation of state. It is this ability to manipulate the solubility of solutes at relatively moderate temperatures and pressures that renders the supercritical fluid so important to chemical engineers. It is also this sensitivity that makes it very important to understand the characteristics of any flow field in which a supercritical fluid solvent and its solutes move because the precipitation of solutes is a primary factor in most industrial processes, especially RESS. The rapid expansion of supercritical solutions has been studied for nearly twenty years since the early papers of Smith and coworkers (Matson *et al.* 1986; Matson *et al.* 1987) to fabricate nano-scale particles, growth thin films, and as an interface to instrumentation such as mass spectrometers and gas chromatographs. A schematic of a typical RESS experiment is shown in figure 1.1. A solute is dissolved in a supercritical solvent at high pressure and the solution is then expanded in a free-jet expansion during which the solvent density and pressure decrease rapidly causing small clusters of solutes to come out of the solution and deposit onto a surface. An excellent overview of the RESS process, including the fluid mechanics of the subsonic and supersonic flows, is provided by Weber and Thies (2002). RESS can be used to generate ceramic precursors (Matson *et al.* 1986; Peterson *et al.* 1986), polymeric powders and fibers (Lele and Shine 1992; Tom *et al.* 1994), and fine powders of pharmaceutical or organic compounds with a narrow size distribution (Chang and Randolph 1989; Tom and Debenedetti 1991; Ksibi *et al.* 1995). We are also interested in using the RESS process to form small clusters for molecular beam experiments. Although the RESS has been studied extensively, there has been little advancement in quantitative predictions for this complicated process. Not only are the gas dynamics difficult because the fluids are non-ideal and the flows involve shock waves, but the clustering formation kinetics, particle growth in the expansion, and subsequent surface interactions are not well understood. Some of the most detailed analyses are those by Debenedetti and coworkers (Kwauk and Debenedetti 1993; Weber *et al.* 2002) who incorporated classical condensation theory into an approximation of the free-jet expansion, and Reverchon and Pallado (1996) who also used one-dimensional approximations of the expansion and compared them with temperature measurements far out in the expansion. Türk (Helfgen *et al.* 2003) has also recently applied a one-dimensional flow model to study the formation of small organic particles in RESS expansions. All of these research groups examined the morphologies as a function of source geometry, pressure, and temperature.

The general guidelines for forming large or small particles by controlling the source geometry, such as capillary or sharp-edged orifice sources, and source conditions including solute concentration, pressure, and temperature are beginning to be understood qualitatively. However, there have been no measurements of the extent and nature of the solute clustering within RESS expansions. Nearly all studies collect the products on a surface and examine the subsequent films.

The most difficult part of RESS fluid mechanics is modeling the supersonic free-jet expansion. There have been several axisymmetric calculations of free-jet supersonic expansions assuming ideal gas conditions, but none for a real gas. The schemes used previously include the method of characteristics and time marching techniques such as MacCormack, Lax-Wendroff (Sinha *et al.* 1971; Forney 1991), and piece-wise linear Godunov methods (Colella and Woodward 1984; Saito *et al.* 1986). Because the method of characteristics cannot calculate the flow field beyond the region of the shock, time marching methods are needed to capture the shock wave structure for the entire flow. They also have the ability to deal with the subsonic regions of the flow field.

The primary goal of this thesis is to solve the axisymmetric, two-dimensional, free-jet expansion of a supercritical fluid, characterized by a useful non-ideal gas equation of state, such as the Redlich-Kwong or Peng-Robinson equations of state which can be extended to multicomponent systems of interest to chemical engineers. These equations of state incorporate both the repulsive excluded volume and attractive potential real gas effects. We begin this research program by studying a pure supercritical fluid solvent, CO_2, without the added complication of solutes in order to assess the suitability of our numerical approach.

We have primarily studied the free-jet expansion of pure CO_2 originating from small orifices rather than capillary tubes in order to ignore the effects of viscosity, heat transfer, and clustering or condensation in the subsonic flow. The subsonic expansion from stagnation to sonic conditions in the orifice flow is essentially inviscid and adiabatic (hence isentropic) providing a well defined inlet boundary condition for our free-jet calculations and for our experiments.

Chapter 2 presents the primary result of this thesis research, the numerical method used to solve the axisymmetric free-jet expansion based upon the Redlich-Kwong and Peng-Robinson equations of state to model CO_2. We have made calculations for the expansion into the atmosphere and for the expansion impacting a flat plate to simulate the use of RESS to grow a thin film. Chapter 3 presents experimental data for mass flow rate, optical shadowgraph measurements of the free-jet and shock wave structure, impact pressure and temperature along the plate, and temperature in the expansion, for supercritical CO_2 and for argon, in order to provide a test of the calculations without the effects of the non-ideal gas thermodynamics. Chapter 4 presents some calculations and experiments for expansions from a capillary source, for comparison with the orifice source, because it is widely used in RESS technologies. Chapter 5 presents a preliminary analysis of the condensation of pure CO_2 in the free-jet expansion, and introduces the future work to be done with a time-of-flight mass spectrometer facility, directly coupled to a supercritical fluid nozzle source, to study the clustering of CO_2 solvent and solutes in RESS applications.

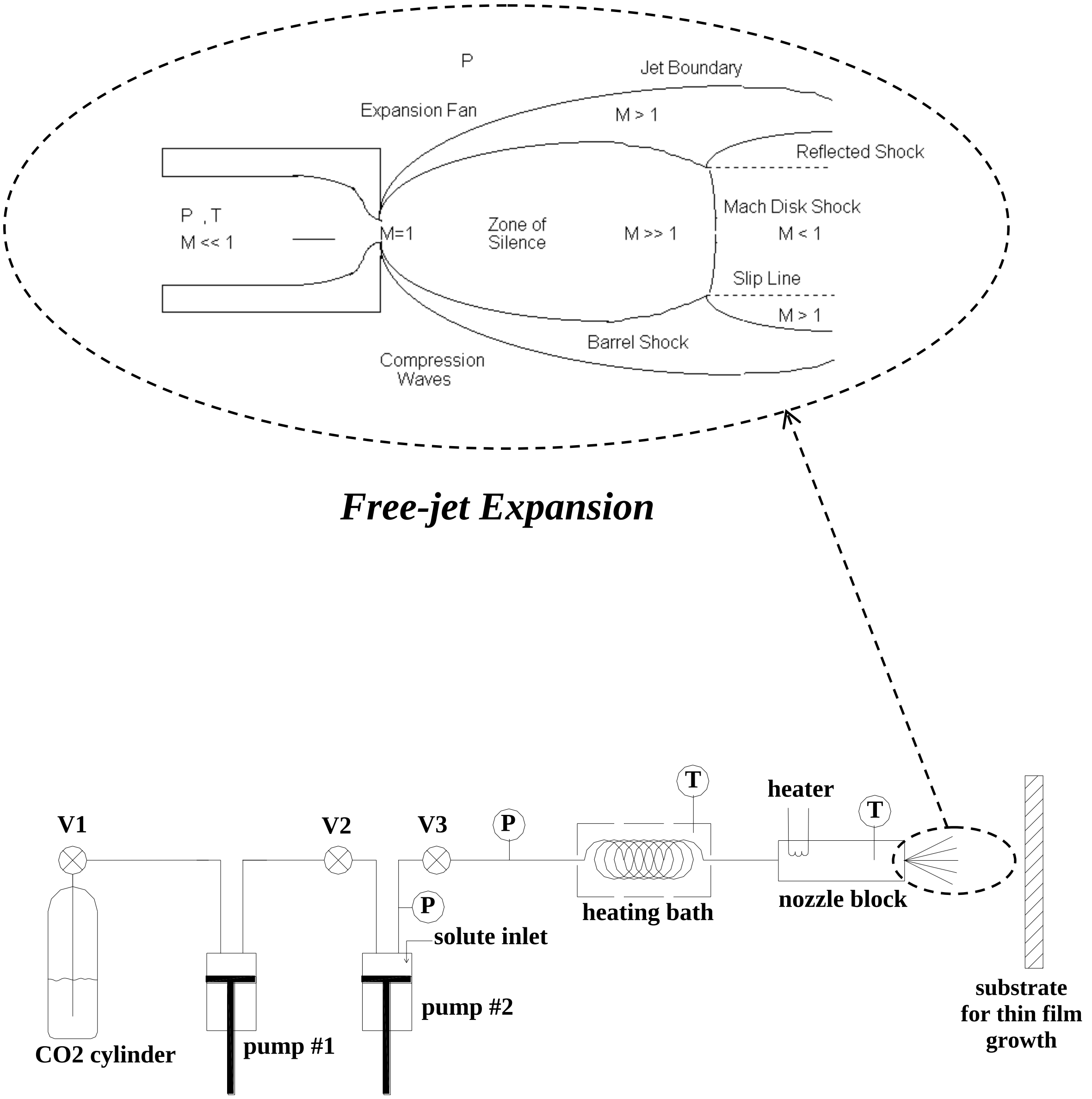


Figure 1.1: Schematic of RESS experiment and free-jet expansion

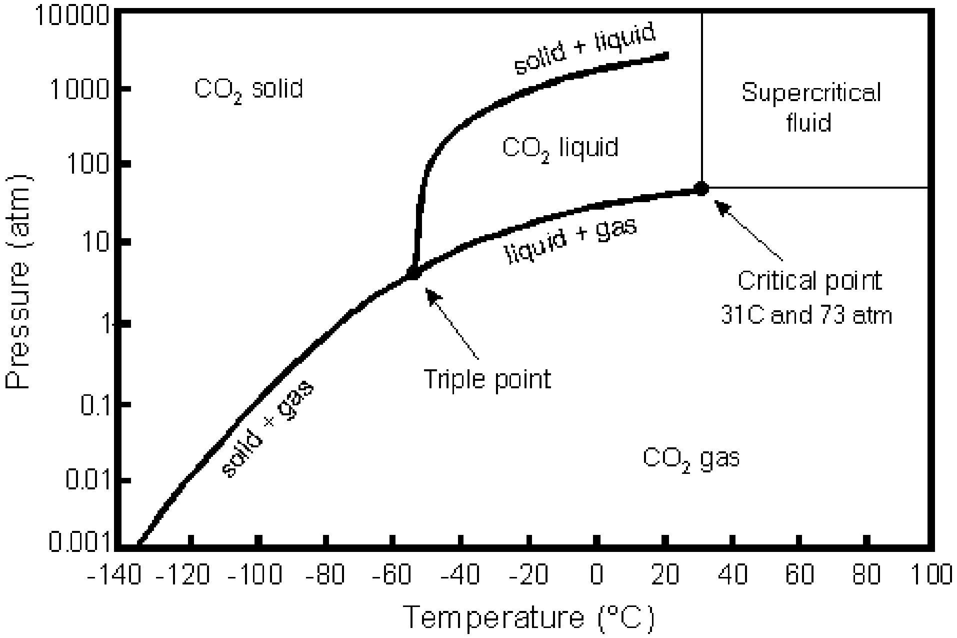


Figure 1.2: Phase diagram for carbon dioxide

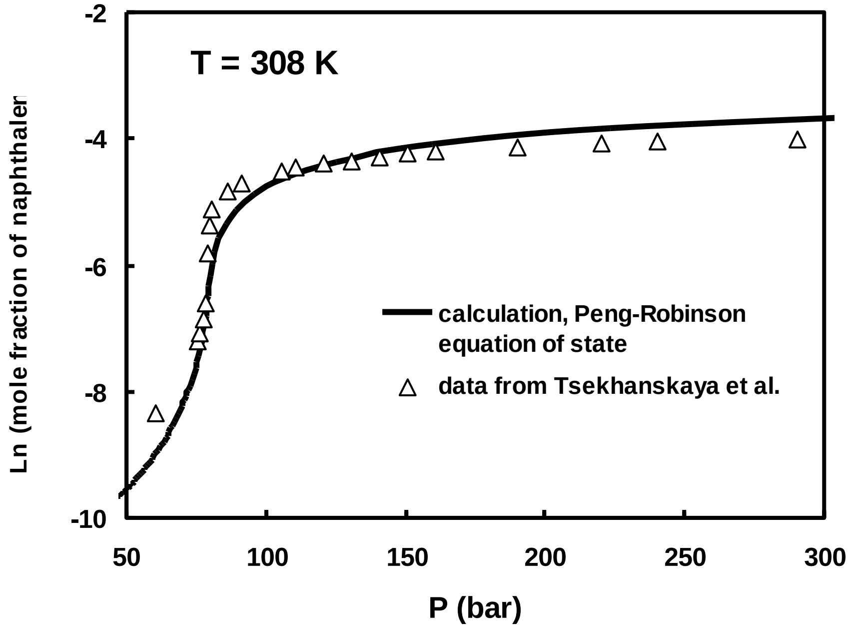


Figure 1.3: Solubility of naphthalene in carbon dioxide at constant temperature

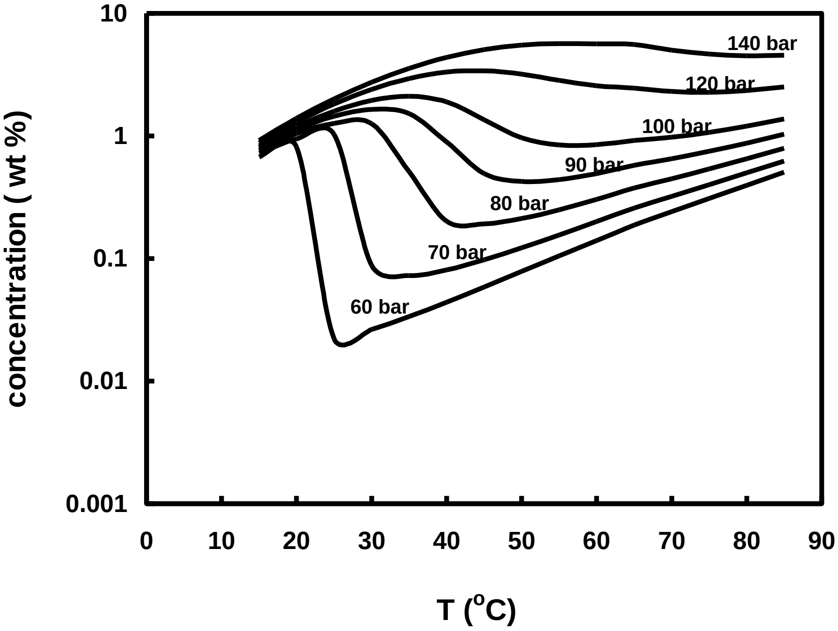


Figure 1.4: Effects of pressure and temperature on the solubility of naphthalene in carbon dioxide

REFERENCES

Arai, Y., Supercritical Fluids: Molecular Interactions, Physical Properties, and New Applications, Springer, New York (2001)

Brand, J., and Miller, D. R., Deposition of Thin Alumina Films from Supercritical Water Jets, Chemical Perspectives of Microelectronic Devices, Mat. Res. Soc. **131**, pp 617 (1989)

Chang, C. J., and Randolph, A. D., Precipitation of Microsize Organic Particles from Supercritical Fluids, AICHE J. **35**, pp 1876 (1989)

Clifford, T., Fundamentals of Supercritical Fluids, Oxford Science Publications (1999)

Colella, P., and Woodward, P., The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations., J. of Computational Physics **54**, pp 174 (1984)

Forney, L. J., Particle Impaction in Axially Symmetric Supersonic Flow., J. Aerosol Science and Technology **15**, pp 49 (1991)

Hanney, J. B., and Hogarth, J., Proc. R. Soc., London, 29, 324 (1879)

Helfgen, B., Türk, M., and Schaber, K., Hydrodynamic and Aerosol Modelling of the Rapid Expansion of Supercritical Solutions, J. Supercrit. Fluids **26**, pp 225 (2003)

Kim, J., Paxton, T., and Tomasko, D., Microencapsulation of Naproxen Using Rapid Expansion of Supercritical Solutions, Biotechnol. Prog., Ohio, **12**, pp 650 (1996)

Ksibi, H., Subra, P., and Garrabos, Y., Formation of Fine Powders of Caffeine by RESS, Advanced Powder Technol., **6**, pp 25 (1995)

Kwauk, X., and Debenedetti, P., Mathematical Modeling of Aerosol Formation by Rapid Expansion of Supercritical Solutions in a Converging Nozzle., J. Aerosol Science **24**, pp 445 (1993)

Lele, A. K., and Shine, A. D., Morphology of Polymers Precipitated from a Supercritical Solvent, **38**, pp 742 (1992)

Levit, N., Pestov, D., and Tepper, G., High Surface area Polymer Coatings for SAW-Based Chemical Sensor Applications, Sensors and Actuators B **82**, pp 241 (2002)

Maharrey, S., and Miller, D. R., A Direct Sampling Mass Spectrometer Investigation of Oxidation Mechanisms for Acetic Acid in Supercritical Water, J. Phys. Chem. A **105**, pp 5860 (2001)

Matson, D., Peterson, R., and Smith, R., Formation of Silica Powders from the Rapid Expansion of Supercritical Solutions, Adv. Cer. Mat., **1**, pp 232 (1986)

Matson, D., Peterson, R., and Smith, R., Production of Powders and Films by the Rapid Expansion of Supercritical Solutions., J. Materials Science **22**, pp 1919 (1987)

McHugh, M. and Krukonis, V., Supercritical Fluid Extraction: Principles and Practice, Butterworths, New York (1986)

Miller, D. R., Free Jet Sources, in Molecular Beam Methods, Vol. 1, pp 14, Scoles, G., Oxford University Press, New York (1988)

Mishima, K., Matsuyama, K., Tanabe, D., Yamauchi, S., Young, T. J., and Johnston, K. P., Microencapsulation of Proteins by Rapid Expansion of Supercritical Solution with a Nonsolvent, AIChE J. **46**, pp 857 (2000)

Peterson, R. C., Matson, D., and Smith, R. D., Rapid Precipitation of Low Vapor Pressure Solids from Supercritical Fluid Solutions: The Formation of Thin Films and Powders, J. Am. Chem. Soc., **108**, pp 2100 (1986)

Reverchon, E., and Pallado, P., Hydrodynamic Modeling of the RESS Process, J. Supercritical Fluids **9**, pp 216 (1996)

Saito, T., Nakatsuji, H., Teshima, K., Numerical Simulation and Visualization of Free-Jet Flow Fields, Trans. Jpn. Soc. Aero. Space Sci., **28**, No. 82, pp 240 (1986)

Sinha, R., Zarkay, V., and Erdos, J., Flowfield Analysis of Plumes of Two-Dimensional Underexpanded Jets by a Time-Dependent Method, AIAA Journal **9**, pp 2363 (1971)

Sun, Y., Supercritical Fluid Technology in Materials Science and Engineering, New York (2002)

Tester, J., Holgate, H., Armellini, F., Webley, P., Killilea, W., Hong, G., and Barner, H., Supercritical Water Oxidation Technology: Process Development and Fundamental Research, in Emerging Technologies in Hazardous Waste Management III, ACS Symp. Ser. 518, pp 35 (1993)

Tom, J. W., and Debenedetti, P., Formation of Bioerodible Polymeric Microspheres and Microparticles by Rapid Expansion of Supercritical Solutions, Biotechnol. Prog., **7**, pp 403 (1991)

Tom, J. W., Debenedetti, P., and Jerome, R., Precipitation of Poly(L-lactic acid) and Composite Poly(L-lactic acid)-Pyrene Particles by Rapid Expansion of Supercritical Solutions, J. Supercrit. Fluids, **7**, pp 9 (1994)

Tsekhanskaya, Y., Iomtev, M., Mushkina, E., Solubility of Naphthalene in Ethylene and Carbon Dioxide under Pressure, Russ. J. Phys. Chem., **38**, pp 1173 (1964)

Weber, M., Russell, L., and Debenedetti, P., Mathematical Modeling of Nucleation and Growth of Particles Formed by the Rapid Expansion of a Supercritical Solution under Subsonic Conditions, J. Supercrit. Fluids, **23**, pp 65 (2002)

Weber, M., and Thies, M., Undestanding the RESS Process, in Supercritical Fluid Technology in Materials Science and Engineering, edited by Y. Sun, pp387, New York (2002)

CHAPTER 2

THEORY AND NUMERICAL CALCULATIONS

2.1 Introduction and Approximations

As we described above, the fluid flow into the free-jet expansion begins at essentially stagnation conditions in a source chamber at T_o and P_o , and then undergoes an acceleration to sonic conditions at the source exit, or throat, of the expansion. The fluid then exits and undergoes the supersonic free-jet expansion, which is the principal topic of this thesis. The free-jet flow is supersonic because the source pressures are high such that the expansion pressure at the source exit is still well above the ambient pressure of the atmosphere. In order to decrease the pressure further, the flow into an increasing area must be supersonic. The expansion must ultimately adjust itself to the ambient conditions through shock waves, which are accounted for in the numerical method. The exception to this adjustment, relevant to molecular beam sources, is expansion into a vacuum where no shocks occur but the flow passes from continuum flow to free molecular flow, requiring kinetic theory analyses.

There are two limiting geometries of subsonic nozzle sources wherein the initial subsonic acceleration to the exit is accomplished; a short converging nozzle and a long constant diameter capillary tube. The converging nozzle source is approximated as an isentropic acceleration because the viscosity and heat conduction effects are small if the nozzle is short. There is a thin viscous boundary layer effect which can be calibrated as an effective diameter, or nozzle discharge coefficient, but the main core of the flow is adiabatic and isentropic. Discharge coefficients are typically between 0.9 and 1.0 for the range of Reynolds numbers encountered. At the other extreme, the capillary source is dominated by a viscous pressure drop along the capillary and requires a more sophisticated calculation to predict the nozzle exit conditions.

Regardless of how the flow is accelerated to the sonic exit conditions, the free-jet calculation of interest in this thesis begins at the nozzle exit, where we will assume uniform nozzle exit conditions that provide uniform entrance conditions for our free-jet calculation. This is an approximation which is not exact for capillary sources nor thin orifices, but it has been shown for ideal gases that the approximation does not effect the flow significantly beyond one or two source diameters (Murphy and Miller 1984). Although we also investigate free-jets from capillary sources (chapter 4 below), most of this thesis concentrates on the isentropic orifice source because it provides the most well defined initial conditions for the free-jet, both experimentally and theoretically. This source therefore provides the best check on our calculations and measurements.

Except near solid boundaries and in the jet boundary shear layers, the neglect of viscosity and heat conduction is a good approximation. For the supercritical fluids and conditions considered here, the more serious approximation is that we assume a homogeneous fluid, neglecting clustering and condensation effects. An introduction to the type of analysis required to incorporate condensation is given in chapter 5.

Finally, the real gas equations of state are selected to represent ones utilized by chemical engineers in the supercritical fluid field, especially those readily adaptable to mixtures of supercritical fluid solvents with solutes. The primary task of this thesis was to include such equations of state into the rigorous numerical analysis of the supersonic free-jet. As we discuss below, this requires care in selecting initial stagnation conditions (e.g. avoiding expansions for which pressures can become negative) determining the best strategy to establish the initial guess for the numerical time marching technique, and accurate evaluation of thermodynamic properties including the speed of sound.

2.2 Numerical Method Used in the Free-Jet Calculations

To calculate the flow, a shock capturing method that automatically finds and obtains the location of the Mach disk was chosen. The capturing method finds where the shock will occur by solving the governing differential equations. The disadvantage of the capturing method is that the shock is spread over a finite number of points on the computational grid which means that the location of the shock is not precise to within a few grid points. The calculated shock's thickness is therefore larger than the actual thickness of the shock.

When using the shock capturing method, experience shows that the solution using the conservative form of the differential equations is stable and smooth, while the non-conservative form oscillates before and after the shock, and the shock location may appear in the wrong location (Anderson 1995). This property is due to the fact that the conservative form conserves the flux of mass, ρ, momentum, ρu, and ρv, and energy, $\rho(e+1/2(u^2 + v^2))$, and the difference equations automatically satisfy the Rankine-Hugonot relations across each cell including those of the shock.

Since we assume the flow is inviscid, the equations solved are Euler's equations. Steady state Euler's equations are hyperbolic when the flow is supersonic, and elliptic when the flow is subsonic. Introducing the time variable into the differential equations makes them hyperbolic with respect to time, whether the flow is subsonic or supersonic, and allows us to solve a steady state problem using an unsteady form of Euler's equations. These time-dependent equations are solved until the solution converges to the steady state solution.

Although the method of characteristics is regarded as the most accurate for the isentropic supersonic free-jet expansion, the time-dependent methods, in addition to correctly capturing the shock wave structure and including the kinetic effects, are needed to extend the calculation to the viscous subsonic flow region. Many texts provide recipes for and summarize the advantages and disadvantages of various finite difference numerical methods applicable to the free-jet supersonic expansion (Hirsch 1990; Hoffman 1992; Chung 2002). One of the difficulties of incorporating real gas effects into the supersonic compressible flow calculation is the need to calculate the speed of sound at each grid point and time step. This evaluation for the speed of sound is especially important in methods which incorporate compression and rarefaction waves, such as the piecewise-linear method (Saito *et al.* 1986). This method is a sequel to Godunov's flux splitting method based on the concept of utilizing localized solutions of the one-dimensional Rieman problem to estimate the flow behavior in two-dimensional flow. Often for hypersonic nozzles, wherein flows originate from high temperature and pressure, it is sufficient to consider only the finite volume repulsive potential correction to ideal gases (Candler 2003). However, the supercritical fluid flows of this thesis originate from high pressure and sufficiently low temperatures that it is necessary to consider both the finite volume and the attractive potential corrections to ideal gas behavior. We have chosen the finite difference, two-step Lax-Wendroff method because it has the following advantages: It is suitable for the hyperbolic partial differential equations of the free-jet; it has been utilized successfully for ideal gas free-jet expansions; the real fluid equations of state are easily incorporated; and the speed of sound enters explicitly only in the stability criteria for the time increment.

2.3 Quasi-One-Dimensional Approximation

An approximation to the two-dimensional axisymmetric equations are the quasi-one-dimensional equations (QOD). The QOD equations are often used in compressible flow to study subsonic-to-supersonic expansions in nozzles where a physical nozzle area A(x) is prescribed (Miller 1988; Anderson 1990). The equations are quasi-one-dimensional because the properties are assumed to change only in the flow coordinate x along the nozzle centerline, and to be constant normal to this direction. The QOD equations are obtained rigorously from the exact axisymmetric equations by integrating over the direction normal to the centerline. They can be easily solved as exact algebraic equations or with the time marching Lax-Wendroff method (Anderson 1995). This QOD problem can therefore serve as a test for the numerical solution of the time-dependent method.

Unfortunately the free-jet has no physical nozzle boundary so the area A(x) is not prescribed. However, a common approach is to solve the axisymmetric free-jet (ASFJ) problem for Mach number along the centerline, and then use these rigorous results to work backwards with the QOD equations to identify an effective A(x) which will mimic the ASFJ. For our QOD studies we have taken an A(x) which closely mimics the ideal gas expansion with constant specific heat ratio $\gamma = 1.4$, which as we show below is a useful approximation for CO_2. In addition, we have identified stream tubes from our ASFJ calculations below to determine an effective A(x), by tracing a streamline back from the Mach disk to the inlet, thus defining a stream tube or effective diverging nozzle contour in which the central core of the expansion occurs (figure 2.28). The QOD calculations can also be easily extended into the subsonic regime upstream of the sonic exit. We obtained an A(x) approximation for this regime based on early pitot tube data for real nozzles (Ashkenas and Sherman 1966).

We use the QOD results to compare equations of state, to test our time marching numerical calculation, and to make preliminary assessments of kinetic effects, such as vibrational relaxation (Anderson 1970) or condensation. Investigators in Rapid Expansion of Supercritical Solutions (RESS) have used the QOD to investigate the expansion (Reverchon and Pallado 1996; Weber and Thies 2002).

2.4 Isentropic Calculations

The inlet conditions for the free-jet are established by an exact isentropic calculation from stagnation conditions (P_o, T_o, h_o, s_o) to sonic conditions at the orifice exit. Together with the thermodynamic relations for enthalpy, h, entropy, s, and the speed of sound, c, this requires the simultaneous solution of three equations for the three unknowns, ρ, T, and u:

$$\begin{aligned} &h(\rho,T) + u^2/2 = h_o(\rho_o,T_o) \\ &s(\rho,T) = s_o(\rho_o,T_o) \\ &u = c(\rho,T) \end{aligned} \qquad (2.1)$$

Together with the exit orifice diameter this calculation also provides the flow rate. We have assumed no streamline curvature at the inlet to the free-jet, which is an approximation best achieved by a short converging nozzle (Miller *et al.* 1982).

2.5 Governing Differential Equations

2.5.1 Free-Jet Equations

As stated before, the preferred method of solving the governing differential equations is the conservative form of Euler's equations. The conservative form preserves the flux of all quantities, ρ, ρu, ρv, and $\rho(e+1/2(u^2+v^2))$. The conservative, two-dimensional axisymmetric, inviscid flow equations are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial r} + \frac{\rho v}{r} = 0 \qquad (2.2)$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2 + P}{\partial x} + \frac{\partial \rho u v}{\partial r} + \frac{\rho u v}{r} = 0 \qquad (2.3)$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho u v}{\partial x} + \frac{\partial \rho v^2 + P}{\partial r} + \frac{\rho v^2}{r} = 0 \qquad (2.4)$$

$$\frac{\partial\, \rho\left\{e + 1/2\,(u^2 + v^2)\right\}}{\partial t} + \frac{\partial\, u\left[\rho\left\{e + 1/2\,(u^2 + v^2)\right\} + P\right]}{\partial x}$$
$$+ \frac{\partial\, v\left[\rho\left\{ e + 1/2\,(u^2 + v^2)\right\} + P\right]}{\partial r} + \frac{v\left[\rho\left\{ e + 1/2\,(u^2 + v^2)\right\} + P\right]}{r} = 0 \qquad (2.5)$$

The first equation (2.2) represents the conservation of mass. The second and third equations (2.3) and (2.4) are the conservation of momentum in the axial and radial directions respectively. The fourth equation (2.5) is the conservation of energy.

In these equations, ρ is the density of the fluid, u is the velocity in the axial direction, v is the velocity in the radial direction, P is the pressure, and e is the internal energy per unit mass. With additional equations of state for P(ρ,T) and e(ρ,T), we have six total differential equations for six unknowns, ρ, u, v, e, T, and P. The equations are non-dimensionalized with respect to the properties at the stagnation chamber, T_o, P_o, ρ_o, and with respect to the orifice diameter D, as follows:

$$\tilde{P} = P/P_o;\ \tilde{e} = e/T_oR;\ \tilde{\rho} = \rho/\rho_o;\ \tilde{u} = u\Big/\sqrt{T_oR};\ \tilde{v} = v\Big/\sqrt{T_oR};$$
$$\tilde{t} = t\sqrt{T_oR}\Big/D;\ \tilde{x} = x/D;\ \tilde{r} = r/D$$

where R is the gas constant per unit mass.

The resulting non-dimensional conservative equations, for axisymmetric, inviscid flow, are then written in vector form as follows:

$$\frac{\partial E}{\partial \tilde{t}} + \frac{\partial F}{\partial \tilde{x}} + \frac{\partial G}{\partial \tilde{r}} + \frac{H}{\tilde{r}} = 0 \tag{2.6}$$

$$E = \begin{vmatrix} \tilde{\rho} \\ \tilde{\rho}\tilde{u} \\ \tilde{\rho}\tilde{v} \\ \tilde{\rho}\left\{\tilde{e} + 1/2\left(\tilde{u}^2 + \tilde{v}^2\right)\right\} \end{vmatrix}, \qquad F = \begin{vmatrix} \tilde{\rho}\tilde{u} \\ \tilde{\rho}\tilde{u}^2 + \tilde{P}Z_o \\ \tilde{\rho}\tilde{u}\tilde{v} \\ \tilde{u}\left[\tilde{\rho}\left\{\tilde{e} + 1/2\left(\tilde{u}^2 + \tilde{v}^2\right)\right\} + \tilde{P}Z_o\right] \end{vmatrix}$$

$$G = \begin{vmatrix} \tilde{\rho}\tilde{v} \\ \tilde{\rho}\tilde{u}\tilde{v} \\ \tilde{\rho}\tilde{v}^2 + \tilde{P}\tilde{Z}_o \\ \tilde{v}\left[\tilde{\rho}\left\{\tilde{e} + 1/2\left(\tilde{u}^2 + \tilde{v}^2\right)\right\} + \tilde{P}Z_o\right] \end{vmatrix}, \qquad H = \begin{vmatrix} \tilde{\rho}\tilde{v} \\ \tilde{\rho}\tilde{u}\tilde{v} \\ \tilde{\rho}\tilde{v}^2 \\ \tilde{v}\left[\tilde{\rho}\left\{\tilde{e} + 1/2\left(\tilde{u}^2 + \tilde{v}^2\right)\right\} + \tilde{P}Z_o\right] \end{vmatrix}$$

where “E” is the solution vector, “F” and “G” are the flux vectors and “H” is the source term. Z_o is the compressibility at the stagnation condition, $Z_o = P_o / \rho_o R T_o$, and for an ideal gas $Z_o = 1$.

The flow is axisymmetric and therefore only one-half of the flow is calculated. Figure 2.1 shows the grid used for the free-jet impacting a vertical plate, with the plate placed at a distance Xp = 9.4D, where D is the orifice diameter. The grid is rectangular with 226 x 240 points. The indices i and j denote equally spaced grid points in the axial and radial directions, respectively ($\Delta x = \Delta r = D/24$). The numerical algorithm can be adjusted for an arbitrary plate location, and for our results, Xp was varied from 5D to 15D. We found that for the free-jet into the atmosphere, the grid in the axial direction needs to be at least equal to 20D in order for the numerical calculation to reach a steady state solution. The first 12 grid points on the left side of the grid are assigned to the orifice radius. To avoid the singularity of the equations at $r = 0$, we shift the grid points nearest to the axis of symmetry by $\Delta r/2$ from the exact location of the axis, and use a symmetry boundary condition.

The two-step Lax-Wendroff method proposed by Sinha (Sinha *et al.* 1971) and modified later by Forney (Forney 1991) was chosen for the calculation. We assume that the flow properties at time t are known and we use a Taylor series expansion in time to get the new flow field at $t + \Delta t$. The numerical solution will directly give us E_1, E_2, E_3, and E_4 for each time step and then we solve for ρ, u, v, e, T and P as follows:

$$\begin{aligned}
\rho &= E_1\\
u &= E_2/E_1\\
v &= E_3/E_1\\
e &= \frac{E_4}{\rho} - \frac{1}{2}(u^2 + v^2)\\
T &= T(\rho, e)\\
P &= P(\rho, T)
\end{aligned} \tag{2.7}$$

Once ρ, u, v, e, T, and P are calculated, we can calculate the elements of the vectors "F", "G" and "H".

A full time step is used in the predictor step with forward differencing:

$$\begin{aligned}
E^{p}(\tilde{x}, \tilde{r}, \tilde{t} + \Delta\tilde{t}) = E^{*}(\tilde{x}, \tilde{r}, \tilde{t})\\
-\frac{\Delta\tilde{t}}{\Delta\tilde{x}}\left[F(\tilde{x} + \Delta\tilde{x}, \tilde{r}, \tilde{t}) - F(\tilde{x}, \tilde{r}, \tilde{t})\right]\\
-\frac{\Delta\tilde{t}}{\Delta\tilde{r}}\left[G(\tilde{x}, \tilde{r} + \Delta\tilde{r}, \tilde{t}) - G(\tilde{x}, \tilde{r}, \tilde{t})\right]\\
-\frac{\Delta\tilde{t}}{\tilde{r}} H(\tilde{x}, \tilde{r}, \tilde{t}) + S^{t}_{i,j}
\end{aligned} \tag{2.8}$$

where E^{*} is a spatial smoothing term :

$$\begin{aligned}
E^{*}(\tilde{x}, \tilde{r}, \tilde{t}) = \frac{1}{4}\Big[E(\tilde{x} + \Delta\tilde{x}, \tilde{r}, \tilde{t})\\
+ E(\tilde{x} - \Delta\tilde{x}, \tilde{r}, \tilde{t})\\
+ E(\tilde{x}, \tilde{r} + \Delta\tilde{r}, \tilde{t})\\
+ E(\tilde{x}, \tilde{r} - \Delta\tilde{r}, \tilde{t})\ \Big]
\end{aligned} \tag{2.9}$$

and $S_{i,j}$ is the artificial viscosity. The artificial viscosity we used is known to be successful with the two-step Lax-Wendroff technique (Anderson 1995):

$$S^t_{i,j} = \frac{C_x\,|\,p^t_{i+1,j} - 2p^t_{i,j} + p^t_{i-1,j}\,|}{p^t_{i+1,j} + 2p^t_{i,j} + p^t_{i-1,j}}\left(E^t_{i+1,j} - 2E^t_{i,j} + E^t_{i-1,j}\right)$$
$$+\frac{C_y\,|\,p^t_{i,j+1} - 2p^t_{i,j} + p^t_{i,j-1}\,|}{p^t_{i,j+1} + 2p^t_{i,j} + p^t_{i,j-1}}\left(E^t_{i,j+1} - 2E^t_{i,j} + E^t_{i,j-1}\right) \qquad (2.10)$$

The arbitrary parameters, Cx and Cy, are chosen after observing their effect on the solution. Typical values of Cx and Cy are between 0.01 and 0.3, and Cx= Cy=0.15 are used for this paper.

Once the predicted values of the elements of vector “E” are calculated, we calculate the predicted ρ^p, u^p, v^p, e^p, T^p, and P^p as described in equation (2.7). We then calculate the predicted elements of vectors “F”, “G”, and “H”, henceforth called “F^p”, “G^p”, and “H^p”. This process is repeated for all interior grid points shown in figure 2.1. Unlike the internal points, the value of the elements of vectors “F”, “G”, and “H” at the boundary are imposed at all time steps. They are then used with the internal values to calculate the corrected quantities in the next corrector step.

A full time step is used in the corrector step with backward differencing:

$$E(\tilde{x},\tilde{r},\tilde{t}+\Delta\tilde{t}) = \frac{1}{2}\left[E(\tilde{x},\tilde{r},\tilde{t}) + E^p(\tilde{x},\tilde{r},\tilde{t})\right]$$
$$-\frac{\Delta\tilde{t}}{2\Delta\tilde{x}}\left[F^p(\tilde{x},\tilde{r},\tilde{t}) - F^p(\tilde{x}-\Delta\tilde{x},\tilde{r},\tilde{t})\right]$$
$$-\frac{\Delta\tilde{t}}{2\Delta\tilde{r}}\left[G^p(\tilde{x},\tilde{r},\tilde{t}) - G^p(\tilde{x},\tilde{r}-\Delta\tilde{r},\tilde{t})\right]$$
$$-\frac{\Delta\tilde{t}}{2\tilde{r}}H^p(\tilde{x},\tilde{r},\tilde{t}) + S^t_{i,j} \qquad (2.11)$$

Each step is first order accurate since the expansion contains only first order terms in the relevant Taylor series. It becomes second order accurate when these two steps are combined. This reduces the amount of computing time, while maintaining the accuracy of the computation.

The smoothing term E^* and the artificial viscosity term $S_{i,j}$ were used separately in the difference equations. The smoothing term E^* was first used to find a steady state solution of the problem. However, the barrel shock and the Mach disk appear much sharper when the calculation is extended by using the artificial viscosity $S_{i,j}$ for several additional time steps. In agreement with Forney (1991), we found that the smoothing technique was sufficient to locate the shock and damp out oscillations, and that running with artificial viscosity will actually cause the solution to degenerate after 10,000 time steps. By trial and error, and by comparing with known ideal gas results, we therefore only used the artificial viscosity term for about 1000 additional time steps to improve the accuracy of the smoothing solution, discussed below.

The method is explicit, which means that the time step used in the calculation can not be arbitrarily chosen but is constrained by stability requirements. The Courant Friedrich and Lewy (CFL) criteria for stability is used to calculate the time step (Hirsch 1990):

$$\Delta t < \frac{\Delta x \Delta y}{|u|\Delta y + |v|\Delta x + c\sqrt{\Delta x^2 \Delta y^2}} \tag{2.12}$$

This limitation on Δt requires the calculation of the speed of sound, c, at each grid point at the end of each time iteration in order to determine the new time step for the next iteration. The speed of sound $c(\rho,T) = (dP/d\rho)_s^{1/2}$ enters explicitly into the calculation for the stability constraint on the time step. In order to achieve the accuracy required using any of our equations of state, we numerically evaluated the derivative $(dP/d\rho)_s$, at any ρ, and T by selecting a small $\Delta\rho$ about the local ρ and calculating the associated ΔP for an isentropic process using both $P(\rho,T)$ and $s(\rho,T)$ equations of state. Since for the CO_2 free-jet the vibrational degrees of freedom are frozen, we essentially compute the frozen speed of sound which is an appropriate speed for these numerical methods.

We set initial conditions for ρ, u, v, e, T and P, for all the interior grid points in order to start the finite difference solution. Although for fixed boundary conditions the steady state solution of CO_2 is independent of initial conditions, we found that there is a narrow range of acceptable initial values for which convergence can be achieved when using the real gas equations of state. By trial and error we found that a realistic set of initial conditions come from the steady state solution of the ideal gas. The solution of the ideal gas free-jet, with $\gamma = 1.4$, is used as the initial conditions for the real gas CO_2 calculation. For this initial ideal gas expansion, sonic conditions u^*, ρ^* and T^* are applied as the initial condition for all points in the axial direction i =1 to imax when j =1 to12. Ambient conditions are applied for all points in the axial direction when j =13 to jmax, as follows:

$$
\begin{array}{ll}
1 \le i \le \infty,\ 1 \le j \le 12: & 1 \le i \le \infty,\ 13 \le j \le \infty: \\
u(i, j, t) = u^* & u(i, j, t) = 0 \\
v(i, j, t) = 0 & v(i, j, t) = 0 \\
\rho(i, j, t) = \rho^* & \rho(i, j, t) = \rho_{amb} \\
T(i, j, t) = T^* & T(i, j, t) = T_{amb}
\end{array}
\qquad (2.13)
$$

The boundary conditions are the governing elements of the entire computation, and there is a set of boundary conditions that works with each method. Guided by others (Sinha *et al.* 1971), we found that the following set of boundary conditions converge to a steady and stable solution. For the two types of problems we are solving, free-jet impacting the plate and free-jet into the atmosphere, the boundary conditions were the same except for the far right hand side of the grid boundary.

For the boundary conditions on the inflow, j =1 to 12 are the grid points that correspond to the orifice. This is where the jet exits the orifice and sonic conditions were applied:

$$
\begin{array}{l}
1 \le j \le 12: \\
u(1, j, t) = u^* \\
v(1, j, t) = 0 \\
\rho(1, j, t) = \rho^* \\
T(1, j, t) = T^*
\end{array}
\qquad (2.14)
$$

The jet conditions, ρ^*, u^*, and T^* were obtained by solving the isentropic expansion from the stagnation chamber to the orifice exit, where Mach number equals one, as shown in section 2.7.1 below.

The boundary conditions for all grid points above the orifice are based on ambient conditions, with zero velocity in both axial and radial directions:

$$\begin{aligned}&13 \leq j \leq \infty :\\&u(1, j, t) = 0\\&v(1, j, t) = 0\\&\rho(1, j, t) = \rho_{amb}\\&T(1, j, t) = T_{amb}\end{aligned} \tag{2.15}$$

For the boundary conditions on the top boundary, ambient conditions are used for the density and the temperature. The velocity in the radial direction is calculated by linear extrapolation using the two adjacent interior grid points. The velocity in the axial direction is set equal to zero:

$$\begin{aligned}&u(i, j\max, t) = 0\\&v(i, j\max, t) = 2v(i, j\max - 1, t) - v(i, j\max - 2, t)\\&\rho(i, j\max, t) = \rho_{amb}\\&T(i, j\max, t) = T_{amb}\end{aligned} \tag{2.16}$$

For the boundary conditions on the outflow and the free-jet into a plate, a reflection boundary is used (Moretti 1968):

$$\begin{aligned}&u(i\max, j, t) = -u(i\max, j, t)\\&\frac{\partial v}{\partial x}(i\max, j, t) = 0\\&\frac{\partial \rho}{\partial x}(i\max, j, t) = 0\\&\frac{\partial T}{\partial x}(i\max, j, t) = 0\end{aligned} \tag{2.17}$$

For the boundary conditions on the outflow and the free-jet into the atmosphere, we apply the condition of zero-gradient of the flow parameter:

$$
\begin{aligned}
&u(i\max,\ j,t) = u(i\max - 1, j, t)\\
&v(i\max,\ j,t) = v(i\max - 1, j, t)\\
&\rho(i\max,\ j,t) = \rho(i\max - 1, j, t)\\
&T(i\max,\ j,t) = T(i\max - 1, j, t)
\end{aligned}
\tag{2.18}
$$

For the boundary conditions on the centerline, symmetry boundary conditions are used:

$$
\begin{aligned}
&\frac{\partial u}{\partial r}(i,1,t) = 0\\
&v(i,1,t) = -v(i,3,t)\\
&\frac{\partial \rho}{\partial r}(i,1,t) = 0\\
&\frac{\partial T}{\partial r}T(i,1,t) = 0
\end{aligned}
\tag{2.19}
$$

2.5.2 Quasi-One-Dimensional Equations

The quasi-one-dimensional, inviscid, equations are written in the conservative form as follows:

$$
\begin{aligned}
&\frac{\partial\,(\rho A)}{\partial t} + \frac{\partial\,(\rho A u)}{\partial x} = 0\\
&\frac{\partial\,(\rho A u)}{\partial t} + \frac{\partial\,(\rho A u^2 + PA)}{\partial x} = P\frac{\partial A}{\partial x}\\
&\frac{\partial\left[\rho(e + 1/2\,u^2)A\right]}{\partial t} + \frac{\partial\left[\rho(e + 1/2\,u^2)Au + PAu\right]}{\partial x} = 0
\end{aligned}
\tag{2.20}
$$

Together with the equations of state, P = P(ρ,T) and e = e(ρ,T), we have a total of five equations, to solve for five unknowns ρ, u, e, T, and P, assuming the geometry A(x) is specified.

Again, the equations are non-dimensionalized with respect to the properties at the stagnation chamber, T_o, P_o, ρ_o, and c_o, and the throat diameter D, as follows:

$\tilde{P} = P/P_o$; $\tilde{e} = e/e_o$; $\tilde{\rho} = \rho/\rho_o$; $\tilde{T} = T/T_o$; $\tilde{u} = u/c_o$; $\tilde{t} = tc_o/D$; $\tilde{x} = x/D$;
$\tilde{A} = 4A/\pi D^2$

The non-dimensional equations of the conservative, quasi-one-dimensional, inviscid flow are written in vector form as follows:

$$\frac{\partial E}{\partial \tilde{t}} + \frac{\partial F}{\partial \tilde{x}} = G \tag{2.21}$$

$$E = \begin{vmatrix} \tilde{\rho}\tilde{A} \\ \tilde{\rho}\tilde{A}\tilde{v} \\ \tilde{\rho}\left(\tilde{e}k_2 + 1/2\ \tilde{v}^2\right)\tilde{A} \end{vmatrix}, \quad F = \begin{vmatrix} \tilde{\rho}\tilde{A}\tilde{v} \\ \tilde{\rho}\tilde{A}\tilde{v}^2 + \tilde{P}\tilde{A}k_1 \\ \tilde{\rho}\left(\tilde{e}k_2 + 1/2\ \tilde{v}^2\right)\tilde{A}\tilde{v} + \tilde{P}k_1\tilde{A}\tilde{v} \end{vmatrix},$$

$$G = \begin{vmatrix} 0 \\ \tilde{P}k_1 \dfrac{d\tilde{A}}{d\tilde{x}} \\ 0 \end{vmatrix}$$

where $k_1 = P_o/\rho_o c_o^2$, and $k_2 = e_o/c_o^2$, are dimensionless constants.

The convergent-divergent area used in our calculations is shown in figure 2.2. As noted above, it is selected to approximate the rate of expansion of the ASFJ and is discussed below. A grid of 510 points is used to model the convergent-divergent area. The first grid point, labeled point 1, corresponds to the inlet from the stagnation chamber. The last point on the grid, labeled imax, corresponds to the exit of the nozzle. The points are equally spaced. The distance between two points is denoted by Δx and is equal to D/40, where D is the diameter at the throat.

The numerical difference method used is the same as the method discussed above for the axisymmetric free-jet calculation except that for this one-dimensional approximation we were able to use the artificial viscosity damping directly, avoiding the use of the smoothing term.

For the boundary conditions at the inlet of the nozzle all the properties, except the velocity, are specified as constant inlet conditions. The velocity must be unspecified to permit the calculations to determine the flow rate of the choked flow, and is calculated by linear extrapolation using two adjacent interior grid points:

$$\begin{gathered} u(1) = 2u(2) - u(3),\ \rho(1) = \rho_o,\ T(1) = T_o, \\ e(1) = e(\rho_o, T_o),\ P(1) = P(\rho_o, T_o) \end{gathered} \tag{2.22}$$

At the exit of the nozzle, the pressure is specified and the rest of the properties are calculated by linear extrapolation using the two adjacent interior grid points. The value of P set as a boundary condition at the exit determines the shock location inside the nozzle.

Artificial viscosity was discussed for the axisymmetric free-jet above. For the QOD calculations, the artificial viscosity term added to the difference equations is (Anderson 1995):

$$S^t_{i,j} = \frac{C_x \,| p^t_{i+1} - 2p^t_i + p^t_{i-1} |}{p^t_{i+1} + 2p^t_i + p^t_{i-1}} \left(E^t_{i+1} - 2E^t_i + E^t_{i-1} \right) \tag{2.23}$$

The arbitrary parameter Cx was taken as Cx = 0.2 in the QOD calculations.

2.6 Thermodynamics and Equations of State

The additional equations needed to complete the calculations above are the equations of state for the thermodynamic variables: $P = P(\rho,T)$, $e = e(\rho,T)$, $h = h(\rho,T)$, and $s = s(\rho,T)$, where P is the pressure, e is the internal energy, h is the enthalpy, and s is the entropy of the fluid.

For an ideal gas, with constant specific heats, Cp and Cv, these relations would be: $P = \rho RT$, $e = CvT$, $h = CpT$, and $\Delta s = Cp \ln(T_2/T_1) - R \ln(P_2/P_1)$, where R is the gas constant per unit mass. The complexity greatly increases for supercritical CO_2, which is a real gas, rather than an ideal gas. We have used several equations of state in our calculations, including Redlich-Kwong and Peng-Robinson cubic equations (Sandler 1999), as well as the 27-parameter equation (Huang *et al.* 1984). The cubic equations of state are readily adaptable to many mixtures of solutes with solvents because they have been well studied by chemical engineers and the mixture interaction parameters are available (Clifford 1999; Sandler 1999; Poling *et al.* 2000).

The Redlich-Kwong equation of state is:

$$P = \frac{RT}{v-b} - \frac{a}{T^{1/2} v(v+b)} \qquad (2.24)$$

$$a = 0.42784 \frac{R^2 T_c^{2.5}}{P_c}; b = 0.08664 \frac{RT_c}{P_c}$$

where T_c is the critical temperature of the fluid and P_c is its critical pressure.

The Peng-Robinson equation of state is:

$$P = \frac{RT}{v-b} - \frac{a(T)}{v(v+b)+b(v-b)} \tag{2.25}$$

$$a(T) = 0.45724\frac{R^2T_c^2}{P_c}\alpha(T); b = 0.07780\frac{RT_c}{P_c}$$

$$\sqrt{\alpha} = 1+\kappa\left(1-\sqrt{\frac{T}{T_c}}\right) \text{ and } \kappa = 0.37464+1.54226\omega-0.26992\omega^2$$

where ω is the acentric factor.

An additional equation of state for carbon dioxide which we have used, the 27-parameter expansion derived by Huang *et al.* (1984) is:

$$\begin{aligned} Z = \frac{P}{\rho RT} = \; & 1+b_2\rho'+b_3\rho'^2+b_4\rho'^3+b_5\rho'^4+b_6\rho'^5+b_7\rho'^2\exp[-c_{21}\rho'^2] \\ & +b_8\rho'^4\exp[-c_{21}\rho'^2]+c_{22}\rho'\exp[-c_{27}(\Delta T)^2] \\ & +c_{23}\frac{\Delta\rho}{\rho'}\exp[-c_{25}(\Delta\rho)^2-c_{27}(\Delta T)^2] \\ & +c_{24}\frac{\Delta\rho}{\rho'}\exp[-c_{26}(\Delta\rho)^2-c_{27}(\Delta T)^2] \end{aligned} \tag{2.26}$$

where $T' = T/T_c$, $\rho' = \rho/\rho_c$, $\Delta T = 1-T'$, $\Delta = 1-\rho'$. The parameters b_1,…..,b_8, and c_{21},…..,c_{27} are given in the Appendix A.

Calculations of e, h, and s are taken from standard texts (Wark 1995; Sandler 1999). The enthalpy and entropy, using the Redlich-Kwong relation, become:

$$h = h_{IG} + h_R = h_{IG}(T) + RT\left\{ \frac{3a}{2bRT^{3/2}} \ln(1 + b/v) - \frac{Pv}{RT} + 1 \right\}$$

$$s = s_{IG} + s_R = s_{IG}(T,P) + R\ \left\{ -\ln\left[\frac{Pv}{RT}\left(1 - \frac{b}{v}\right) \right] + \frac{a}{2bRT^{3/2}} \ln(1 + b/v) \right\} \qquad (2.27)$$

The enthalpy and entropy using the Peng-Robinson equation of state become:

$$h = h_{IG} + h_R = h_{IG}(T) + RT(Z - 1) + \frac{T(da/dT) - a(T)}{2\sqrt{2}b} \ln\left(\frac{Z + (1 + \sqrt{2})B}{Z + (1 - \sqrt{2})B} \right)$$

$$s = s_{IG} + s_R = s_{IG}(T,P) + R\ln(Z - B) + \frac{(da/dT)}{2\sqrt{2}b} \ln\left(\frac{Z + (1 + \sqrt{2})B}{Z + (1 - \sqrt{2})B} \right) \qquad (2.28)$$

where B = Pb/RT

The enthalpy and entropy using the 27-parameter equation of state become:

$$\begin{aligned} h = h_{IG} + h_R = h_{IG}(T) + RT\Big[\ Z - 1 &+ \left(\frac{c_2}{T'} + 2\frac{c_3}{T'^2} + 3\frac{c_4}{T'^3} + 4\frac{c_5}{T'^4} + 5\frac{c_6}{T'^5} \right)\rho' \\ &+ \frac{1}{2}\left(\frac{c_8}{T'} + 2\frac{c_9}{T'^2} \right)\rho'^2 + \frac{1}{3}\left(\frac{c_{11}}{T'} \right)\rho'^3 + \frac{1}{4}\left(\frac{c_{13}}{T'} \right)\rho'^4 \\ &+ \frac{1}{5}\left(\frac{c_{14}}{T'} \right)\rho'^5 + \frac{1}{2c_{21}^2}\left(3\frac{c_{18}}{T'^3} + 4\frac{c_{19}}{T'^4} + 5\frac{c_{20}}{T'^5} \right) \\ &\times \left\{ 1 - \left(1 + c_{21}\rho'^2\right)\exp\left(-c_{21}\rho'^2\right) \right\} - 2c_{22}c_{27}T'\Delta T\rho' \exp\left\{ -c_{27}(\Delta T)^2 \right\} \\ &+ \frac{c_{23}c_{27}}{c_{25}} T'\Delta T \exp\left\{ -c_{25}(\Delta\rho)^2 - c_{27}(\Delta T)^2 \right\} \\ &+ \frac{1}{2c_{21}}\left(3\frac{c_{15}}{T'^3} + 4\frac{c_{16}}{T'^4} + 5\frac{c_{17}}{T'^5} \right)\left\{ 1 - \exp\left(-c_{21}\rho'^2\right) \right\} \\ &+ \frac{c_{24}c_{27}}{c_{26}} T'\Delta T \exp\left\{ -c_{26}(\Delta\rho)^2 - c_{27}(\Delta T)^2 \right\}\ \Big] \end{aligned}$$

$$\begin{aligned} s = s_{IG} + s_R &= s_{IG}(T) - \left(c_1 - \frac{c_3}{T'^2} - 2\frac{c_4}{T'^3} - 3\frac{c_5}{T'^4} - 4\frac{c_6}{T'^5} \right)\rho' \\ &- \frac{1}{2}\left(c_7 - \frac{c_9}{T'^2} \right)\rho'^2 - \frac{1}{3}c_{10}\rho'^3 - \frac{1}{4}c_{12}\rho'^4 \\ &+ \frac{1}{2c_{21}}\left(2\frac{c_{15}}{T'^3} + 3\frac{c_{16}}{T'^4} + 4\frac{c_{17}}{T'^5} \right)\left\{ 1 - \exp\left(-c_{21}\rho'^2 \right) \right\} \\ &+ \frac{1}{2c_{21}^2}\left(2\frac{c_{18}}{T'^3} + 3\frac{c_{19}}{T'^4} + 4\frac{c_{20}}{T'^5} \right) \\ &\times \left\{ 1 - \left(1 + c_{21}\rho'^2 \right)\exp\left(-c_{21}\rho'^2 \right) \right\} - c_{22}\left(1 + 2c_{27}T'\Delta T \right)\rho' \exp\left\{ -c_{27}(\Delta T)^2 \right\} \\ &+ \frac{c_{23}}{2c_{25}}\left(1 + 2c_{27}T'\Delta T \right)\exp\left\{ -c_{25}(\Delta\rho)^2 - c_{27}(\Delta T)^2 \right\} \\ &+ \frac{c_{24}}{2c_{26}}\left(1 + 2c_{27}T'\Delta T \right)\exp\left\{ -c_{26}(\Delta\rho)^2 - c_{27}(\Delta T)^2 \right\} \end{aligned} \tag{2.29}$$

We have used three approaches to compute the ideal gas contribution for h_{IG} and s_{IG}. An empirical heat capacity $Cp_{,o}(T)$ (Wark 1995) is used to determine the enthalpy and the entropy changes of an ideal gas between T_1 and T_2 as follows:

$$\begin{aligned} \Delta h_{IG} &= \int_{T1}^{T2} \frac{C_{p,o}}{T} dT \\ \Delta s_{IG} &= \int_{T1}^{T2} \frac{C_{p,o}}{T} dT - R \ln\frac{P_2}{P_1} \end{aligned} \tag{2.30}$$

where (Wark 1995) $cp_{,o} = R\,(2.401 + 8.735e^{-3}T - 6.607e^{-6}T^2 + 2.002e^{-9}T^3)$

Over the range of temperatures encountered in our supercritical CO_2 expansions (below 350°C) only the degenerate $\nu_2 = \nu_3$ bending modes of CO_2 are excited (Tien and Lienhard 1971; Hecht 1990). As suggested in figure 2.3, at 350°C only 6% of the molecules are in the first excited state of this bending mode. Therefore, in addition to use of an empirical heat capacity Cp(T) to determine Δh_{IG}, we have also used the rigorous statistical mechanics relation (Anderson 1970; Tien and Lienhard 1971; Hecht 1990) to incorporate the contribution to h, s, and e of these modes for most of our calculations, and in particular to establish the inlet conditions to the free-jet expansion.

Using statistical mechanics results, the ideal gas contribution to the enthalpy h_{IG} is given by:

$$h_{IG} = \frac{7}{2}RT + R\left(\theta_v + \frac{2\theta_v}{e^{\theta_v/T} - 1}\right) + h_{adj} \tag{2.31}$$

The constant h_{adj} is chosen such that h_{IG} is equal to zero at T= 273 K. θ_v is the characteristic vibrational temperature which for carbon dioxide is $\theta_v = 960.1$ K (Hecht 1990).

Similarly, the ideal gas contribution to the entropy s_{IG} is given by:

$$s_{IG} = \frac{7}{2}R\ln\frac{T}{T_{ref}} - R\ln\frac{P}{P_{ref}} + 2R\left(\frac{\theta_v / T}{e^{\theta_v/T} - 1} - \ln\left(1 - e^{-\theta_v/T}\right)\right) + s_{adj} \tag{2.32}$$

where, $T_{ref} = 273$ K, $P_{ref} = 1$ bar, and s_{adj} is chosen such that $s_{IG} = 0$ at T= 273 K and P= 1 bar.

Figure 2.4 shows the calculation of the enthalpy versus temperature for carbon dioxide using the Redlich-Kwong equation of state with the empirical Cp= Cp(T), and compares it to the Redlich-Kwong enthalpy calculated using the exact statistical mechanics expression for h_{IG}. The two different methods of calculating enthalpy give similar results to within 4%.

Finally, as we show below the vibrational bending mode freezes on the expansion very near the nozzle throat. Therefore a good approximation for the free-jet is to assume a constant Cp= 7/2 R. This is the third approach we have used to calculate h_{IG} and s_{IG}.

Often for many steps in the calculations, we needed to invert an equation of state such as h(ρ, T) or s(ρ,T) in order to calculate T, given h and ρ, or s and ρ. Subroutines that use the secant method (Press *et al.* 1994) provide such a root finding calculation and an example subroutine is given in the Appendix B. The initial guess in these subroutines is critical in order to find the correct solution and it was chosen based on the latest available solution in the expansion.

In order to verify the accuracy and consistency of our subroutines for the thermodynamic properties, we evaluated the differential equation of state relation $de - Tds = \int P dv$ which should be satisfied rigorously for a constant composition fluid. We verified the accuracy or our subroutines by integrating the relation over a range of conditions relevant to our CO_2 expansion and we found that the equation was satisfied to within 0.1%.

In the next section we present the results of our calculations. We first discuss results using the QOD approximation, which guided us in the modeling used for the full two-dimensional axisymmetric calculations. The rigorous results are then presented for the axisymmetric expansions, which are the primary results of this thesis.

2.7 Computational Results

2.7.1 Quasi-One-Dimensional

The running time for the QOD calculation , using real gas equations of state for supercritical CO_2, on a Dell PC with a P4 CPU 1.80 GHz at 1 GB RAM, is 2 minutes. The solution reaches steady state after 4000 time steps at which point the flow field variables stop changing. The results presented in this section are for the following stagnation conditions: T_o= 343 K, P_o= 80 bar. The pressure at the exit, P_{exit} , was fixed and set equal to 1 bar. The nozzle throat diameter D is 50 μm, and the length of the nozzle in the supersonic region is 11.5 nozzle diameters.

Figures 2.5, 2.6 and 2.7 compare the solution for the temperature and pressure profiles and for the Mach number for a CO_2 expansion using the Redlich-Kwong equation of state to that assuming an ideal gas, $\gamma = 1.4$. These results show that the ideal gas approximation is remarkably good except for temperature after the shock. The Peng-Robinson equation of state gave similar results to those of the Redlich-Kwong equation of state and some results are summarized in table 2.1 which is discussed further below. We found that the 27-parameter equation of state yields a negative pressure upstream of the shock and we were therefore unable to achieve convergence to a steady state solution in the divergent part of the nozzle for this equation of state. For the isentropic expansion to the throat, the 27-parameter equation of state gives similar results as Peng-Robinson or Redlich-Kwong.

Figure 2.8 shows the temperature profile for the Redlich-Kwong equation of state, using the exact statistical mechanics equations in the thermodynamic properties, with and without the vibrational modes. These results show, as discussed above, that the vibrational modes make only a small contribution to the energy of the expansion. Calculations for vibrational relaxation below support this approximation even further by showing that the vibrational bending modes freeze downstream of the throat. Therefore, a reasonable approximation is to assume that only the rotational and translational degrees of freedom contribute to the ideal gas enthalpy in the free-jet. Therefore when comparing the rigorous equation of state results for supercritical CO_2 to ideal gas results, we use $h_{IG} = 3.5RT$ or the $\gamma = 1.4$ results.

Figure 2.9 shows the trajectory of the expansion when P_o= 80 bar and P_o= 137 bar. Both expansions enter into the two-phase region. The expansion at P_o= 137 bar may actually cross the spinodal curve, where super-saturation is unstable and any perturbation should cause condensation to occur. This condensation could only be inhibited at very low densities where there are insufficient collisions, which should not be the case for this expansion under supercritical conditions. This suggests that our assumption of no condensation may be a poor one under these conditions.

To test the accuracy of the numerical method used, properties at three different locations along the nozzle were calculated analytically and compared to the numerical solution. The three positions were at the throat, before the shock, and after the shock. For an ideal gas, the QOD equations can be simplified to algebraic relations which can be solved analytically, and the properties of the flow can be calculated at any location along the nozzle (Anderson 1990). Unfortunately, for the real gas there is no exact algebraic solution.

To calculate exactly the properties of CO_2 at the throat of the QOD where sonic conditions occur and the mass flow rate is determined, we had to solve the conservation equations (2.1), for isentropic flow from the stagnation source to the throat. These equations were solved using the following iterative algorithm at the sonic throat conditions:

1. guess $\dot{m}$
2. guess ρ_2 ($\rho_2 = \rho_o - \Delta$)
3. $v_2 = \dot{m} / \rho_2 A_2$
4. $h_2 = h_o - {v_2}^2 / 2$
5. $T_2 = T_2(h_2, \rho_2)$
6. $P_2 = P_2(\rho_2, T_2)$
7. $s_2(\rho_2, T_2) \Rightarrow s_2$, if $s_2 \neq s_o$, go back to #2.
8. ${c_2}^2 = \left.\frac{dP}{d\rho}\right)_{S,2} = \frac{P_2 - P_2'}{\rho_2 - \rho_2'}$
9. $\text{Mach} = \frac{v_2}{c_2}$, if Mach$\neq$1, go back to #1 and adjust $\dot{m}$

To calculate the properties before the shock, specified at the calculated A(x) for the flow, in the divergent section of the nozzle, the same set of equations (2.1) are solved using the same algorithm with elimination of the first step, since the mass flow rate was determined by the sonic conditions at the throat.

To calculate the properties after the shock, we solved the conservation equations for a normal shock, known as the Rankine-Hugonot relations between properties upstream 2 and downstream 3:

$$\begin{aligned} &\rho_2 v_2 = \rho_3 v_3 \\ &P_2 + \rho_2 {v_2}^2 = P_3 + \rho_3 {v_3}^2 \\ &h_2 + {v_2}^2 / 2 = h_3 + {v_3}^2 / 2 \\ &P = P\,(\rho, T) \\ &h = h\,(\rho, T) \end{aligned} \tag{2.33}$$

The algorithm used to solve Rankine-Hugonot equations is:

1. guess ρ_3 ($\rho_3 = \rho_2 - \Delta$)
2. $v_3 = \dot{m} / \rho_3$
3. $h_3 = h_2 + v_2^2 / 2 - v_3^2 / 2$
4. $T_3 = T(\rho_3, h_3)$
5. $P_3 = P(\rho_3, T_3)$
6. check if $P_2 + \rho_2 v_2^2 = P_3 + \rho_3 v_3^2$ is satisfied, if not go back to #1

Table 2.1 compares exact and time marching numerical solutions of the QOD equations at the three positions described above, using Redlich-Kwong, Peng-Robinson, and ($\gamma = 1.4$) ideal gas equations of state. These results show that the time marching calculation works well but is least reliable downstream of the shock. Because the different equations of state give similar results, most of our ASFJ calculations below were done with the simplest equation of state, the Redlich-Kwong, in order to probe the supercritical fluid effects in the supersonic free-jet expansion.

To further understand the contribution of the vibrational modes, we examined the vibrational relaxation (Anderson 1970) for the degenerate $\nu_{2,3}$ modes of CO_2, using the QOD expansion. The approach to equilibrium of the vibrational energy is (Bradley 1962):

$$\frac{dE_v}{dt} = u\frac{dE_v}{dx} = \frac{E_{v,eq} - E_v}{\tau} \tag{2.34}$$

where $E_{v,eq}$ is the equilibrium vibrational energy evaluated at the local temperature of the gas, and τ is the vibrational relaxation time. Two different relations of τ for CO_2 were taken from the literature (Anderson 1969; Yamazaki *et al.* 1980; Vincenti 1986):

$$\log(\tau P)_{CO_2-CO_2} = 17.8(T^{-1/3}) - 1.808$$

$$\text{or } \tau = \frac{Z}{\nu} = \frac{\exp(86.8T^{-1/3} - 2.1)}{\bar{c}/\lambda}; \quad \bar{c} = \left(\frac{3P}{\rho}\right)^{1/2} \text{ and } \lambda = \frac{1}{\sqrt{2}\pi d^2 n} \qquad (2.35)$$

where Z is the collision number for vibrational to translational energy transfer, ν is the binary collision frequency, $\bar{c}$ is the mean molecular speed, λ is the mean free path, d is the diameter of the CO_2 molecule, and n is the number density of molecules per unit volume.

Equation 2.34 is integrated along the nozzle from the inlet to the exit. A plot of the fraction of initial vibrational energy remaining, $E_{vib} / E_{vib,o}$, versus distance, x/D, for the same 80 bar and 343 K expansion is shown in figure 2.10. This result shows that vibrational energy stops cooling, freezes, and has negligible contribution to the expansion beyond the sonic exit. Hence, as already suggested above, a reasonable ideal gas approximation for CO_2 under our conditions is expected to be a constant γ = 1.4 free-jet expansion. Further, when using the real gas equation of state, a reasonable Cp for the ideal gas contribution to enthalpy and entropy is simply Cp = 3.5R to account for the translational and rotational modes.

2.7.2 Two-Dimensional Axisymmetric Free-Jet Impacting a Plate

The process of obtaining a solution for an axisymmetric free-jet using the Redlich-Kwong equation of state consisted of three steps: first, the ideal gas codes were run for 2000 time steps to obtain initial conditions for Redlich-Kwong, as stated above. Then the Redlich-Kwong codes using the smoothing term were also run for 2000 time steps. Finally, the Redlich-Kwong codes with the artificial viscosity term was used to finish the calculation (using the solution obtained from the smoothing codes as initial conditions) and were run for 1000 time steps. The total computational time for all three runs for CO_2 ASFJ, on a Dell PC with a P4 CPU 1.80 GHz at 1 GB RAM, is 3 hours.

Figure 2.11 shows our calculated density contours for a supercritical CO_2 ASFJ expansion, at 70^{o}C and 80 bar, directed at a flat plate, and indicates the rapid changes through the barrel and normal shock waves. Figure 2.12 is the density contours for a supercritical CO_2 expansion at 70°C and 137 bar. A comparison of the graphs shows that the Mach disk moves downstream at the higher pressure and that the diameter of the Mach disk increases at higher pressure. This is consistent with ideal gas results developed by Bier and Schmidt (1961). Figure 2.13 is the density contours for the supercritical CO_2 expansion at 70°C and 137 bar, when the plate is set closer to the jet at Xp = 7.5D. It shows that the Mach disk becomes more rounded when the plate is closer to the source.

As a comparison to an ideal gas, a plot of the calculated density contours of an ideal gas with both $\gamma = 1.4$ and $\gamma = 5/3$ are shown in figure 2.14 and figure 2.15 respectively. As expected the shape of the shock structure of CO_2 (figure 2.11) more closely resembles the ideal gas, $\gamma = 1.4$, expansion.

Figures 2.16 and 2.17 show the axisymmetric profile for temperature and pressure, respectively, along the centerline for the supercritical CO_2 at 70°C, for three different values of P_o= 80, 137, and 200 bar. The results demonstrate the ability of the numerical method to capture the shock waves. The source pressure effect on the temperature downstream of the shock due to the real gas effects is significant, and important to RESS experiments.

Figures 2.18 and 2.19 show the centerline temperature and pressure calculated using only smoothing and then adding numerical viscosity for a supercritical CO_2 expansion at 70°C and 80 bar. Additionally, on this plot, is the ideal gas $\gamma = 1.4$ calculation. The artificial viscosity improved the solution up to the shock but introduced numerical oscillation beyond that point. Just as we found from the QOD calculation, the ideal gas approximation is a reasonable first approximation, but the important result is that the real gas solution has a much lower temperature downstream of the shock compared to the ideal gas, which returns close to the stagnation temperature. This result is expected by analogy with the thermodynamic Joule-Thompson effect.

To test the accuracy of the numerical method for this ASFJ, we compared calculated profiles for an ideal gas, $\gamma = 1.4$, along the centerline up to the shock location with well known results from the method of characteristics (Miller 1988; Ashkenas and Sherman 1966). Figure 2.20 shows a comparison between the numerical solution for the temperature profile, using both the smoothing and the numerical viscosity techniques, and the exact solution from the method of characteristics. The Mach disk and flow properties are more accurately defined by including the numerical viscosity on the final time steps, and the Lax-Wendroff results are accurate within 8%.

As we noted above, most of our numerical calculations are based on the Redlich-Kwong equation of state because of its simplicity and because it is well used by chemical engineers to treat supercritical CO_2. Further, the QOD calculations (Table 2.1) showed that there is little difference between the Peng-Robinson and Redlich-Kwong equations of state. Our axisymmetric numerical codes are adaptable to any equation of state and, as an example, we compare pressure and temperature centerline profiles in figures 2.21 and 2.22 for both equations of state. Once again, under our conditions, we find that either equation of state is suitable. Therefore in this thesis, all of the following calculations are made using only the Redlich-Kwong equation of state. When solutes are introduced, there may be more important differences as the equations of state are extended to binary systems.

Figures 2.23 and 2.24 show the ASFJ profiles for pressure and temperature along the plate, normalized by source conditions, for the same three cases. The plate pressure and temperature dependence on source pressure is that expected by the strength of the shock which is determined by its location in the expansion. At the plate, there is a core central region of higher pressure and temperature which falls off at a radial distance of about five source diameters. Experimental data is compared with these results in chapter 3 below. For some cases, the pressure profiles at the plate have oscillations. These oscillations are not numerical but real oscillations observed experimentally and previously reported for ideal gases (Teshima 1990). A detailed example of these oscillations, after the solution has converged to steady state, is shown in figure 2.25 for pressure at the plate for a supercritical CO_2 expansion at 80 bar.

Figures 2.26 and 2.27 show our calculated velocity field at 70°C, 80 bar and 137 bar, for a plate placed at Xp = 9.4D. The flow after the shock is directed more strongly outward at higher pressure. Streamtubes from our ASFJ velocity solution were used to determine an effective area A(x) for the QOD calculations. These were calculated by integrating the equation for a streamline, $r = r_o + \int v/u \, dx$, from a point r_o at the sonic inlet to the Mach disk, where u and v are the axial and radial components of the velocity field. At each step, this equation finds the radial position along the streamline. This usually falls between grid points so the next step in the integration requires interpolation of u and v at the adjacent nodes.

Figure 2.28 shows the area profile, normalized by the sonic or throat area for the appropriate streamline, extracted from our ASFJ calculations for supercritical CO_2 at P_o= 80 and 137 bar as described, and compares it to the area profile obtained from the exact solution for an ideal gas, $\gamma = 1.4$, isentropic expansion result. The two area profiles are quite similar and this result supports the use of the ideal gas QOD approximation analysis which previous researchers have used to investigate supercritical free-jet expansions.

The above streamlines, shown in figure 2.29, can also be used to calculate the percent of the mass flow passing through the Mach disk. This is accomplished by tracing the last streamline, denoted as j = 6 in figure 2.29, which does not cross the disk to its radial position at the inlet. Using this position, which occurs near r = D/4, we conclude that only about 25% of the total flow passes through the Mach disk. This result agrees qualitatively with the estimates of 20% based on ideal gas expansions by Weber and Ties (2002).

2.7.3 Two-Dimensional Axisymmetric Free-Jet into Atmosphere

The same procedure used in calculating the solution for the free-jet impacting the flat plate is used for the free-jet into the atmosphere. Figure 2.30 shows our calculated density contours for a supercritical CO_2 ASFJ expansion into the atmosphere, at 70^oC and 80 bar. Figure 2.31 shows our calculated velocity field for the jet at the same conditions. Figures 2.32 and 2.33 show the ASFJ profiles for pressure and temperature along the centerline. By comparing the density fields of the expansion into the atmosphere to the jet impacting the flat plate, it is shown that the Mach disk has much less curvature when the plate is removed and becomes more rounded as the plate is placed closer to the source. By comparing the velocity field of the two expansions, we see that the flow field is bent outward when the plate is placed in front of the jet.

We had difficulties converging to a solution for the free-jet expansion into the atmosphere at a high stagnation pressure, P_o= 200 bar, where multiple shocks can appear. We also believe this numerical difficulty may be due to the fact that at such high pressures, the expansion has crossed the spinodal regime where the equation of state is unstable, and where condensation may be inevitable. In fact, under such conditions, our shadowgraph measurements below show a second shock forming in addition to CO_2 condensation downstream of the Mach disk. We are still investigating this condensation phenomenon.

In this chapter we have shown that our numerical method is able to converge to reasonable solutions for the free-jet expansions of supercritical fluids, capturing the required shock wave structure. In the next chapter, we describe experimental results which indicate that these calculations provide very reasonable results for the jet structure. We are therefore encouraged to extend the calculations to include solutes and possible condensation and/or precipitation effects, although as we show in chapter 5, the latter may be too difficult to deal with in a quantitative manner.

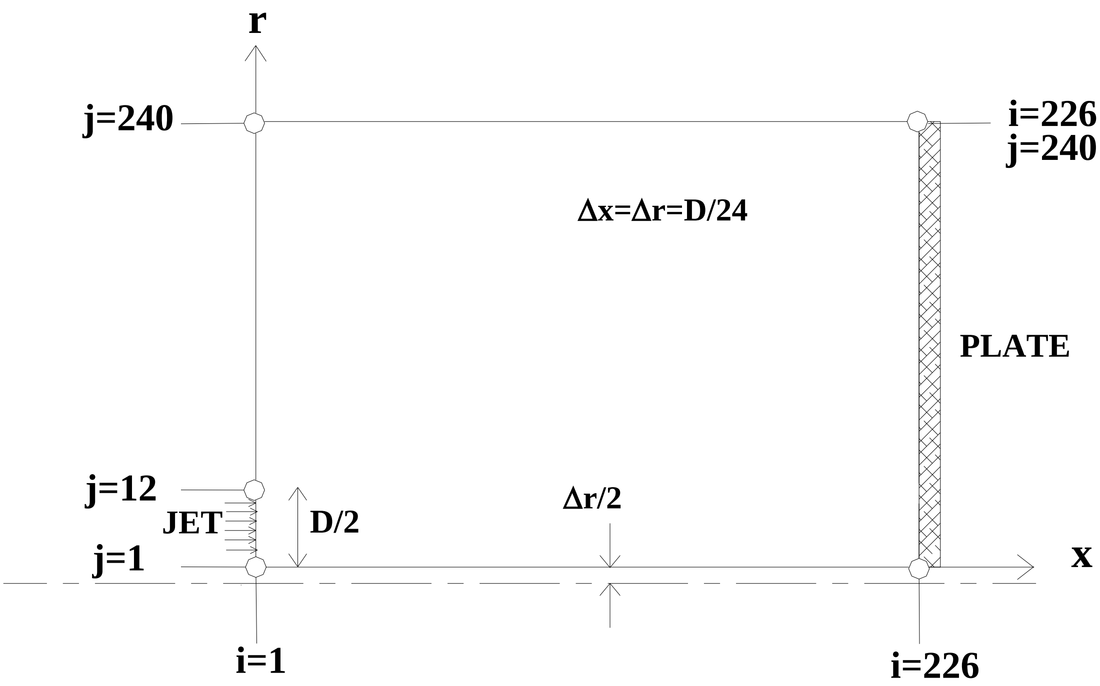


Figure 2.1: Rectangular computational grid

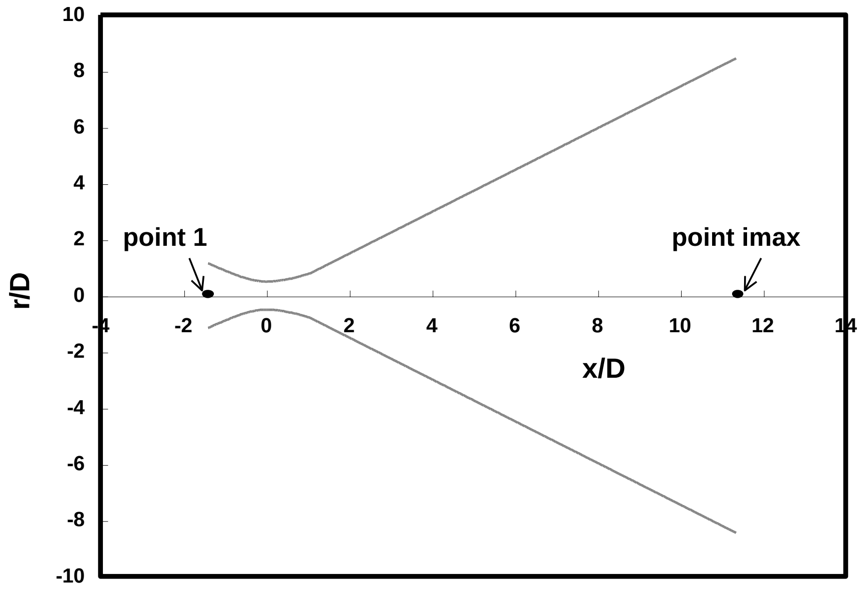


Figure 2.2: Convergent-divergent nozzle profile

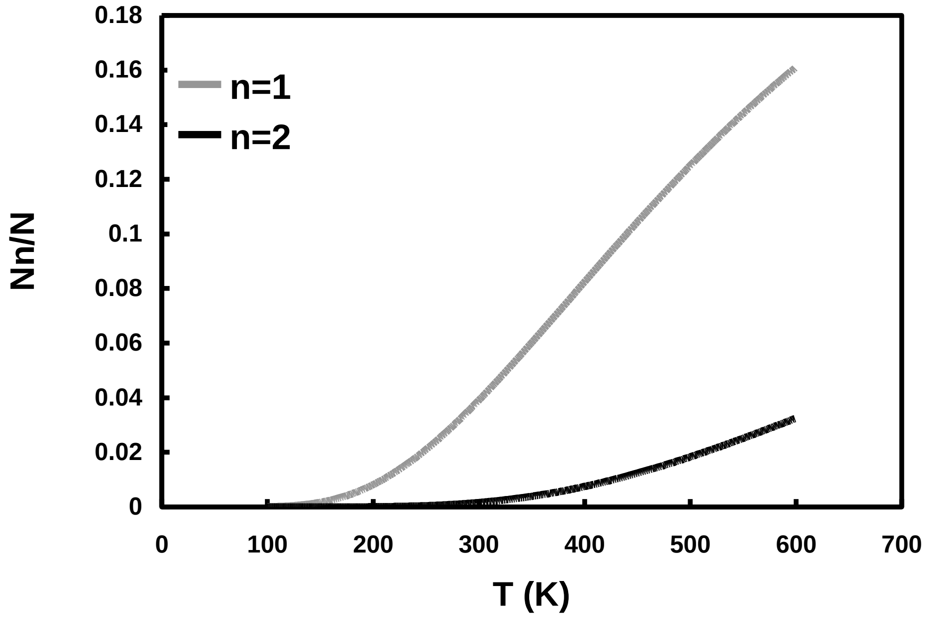


Figure 2.3: Fraction of CO_2 molecules in the n^{th} vibrational state of the degenerate bending mode

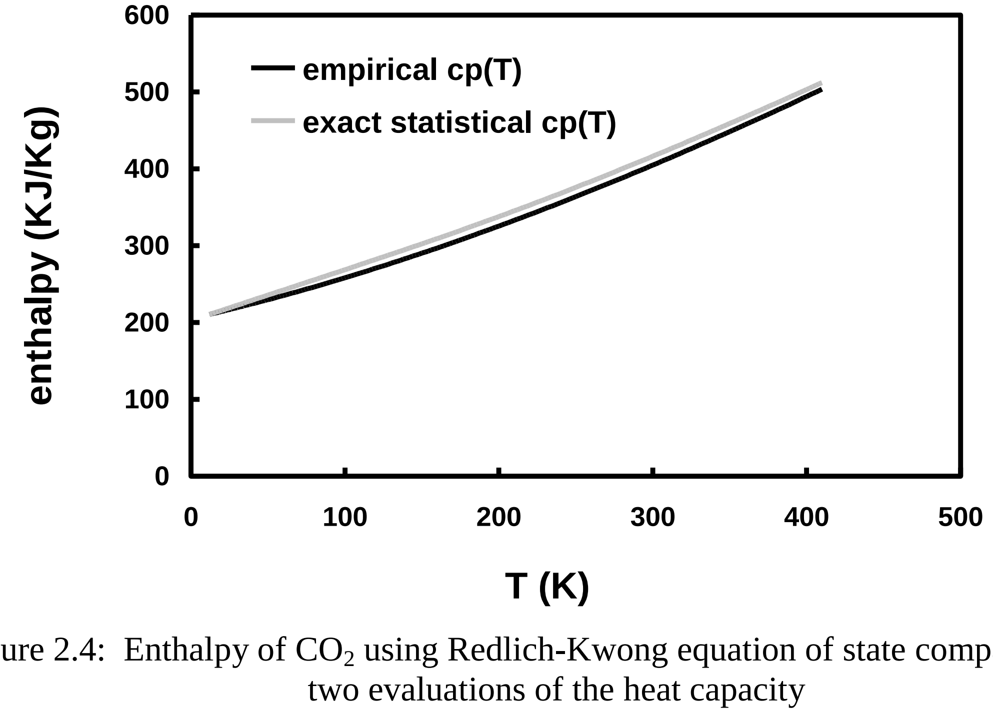


Figure 2.4: Enthalpy of CO_2 using Redlich-Kwong equation of state comparing two evaluations of the heat capacity

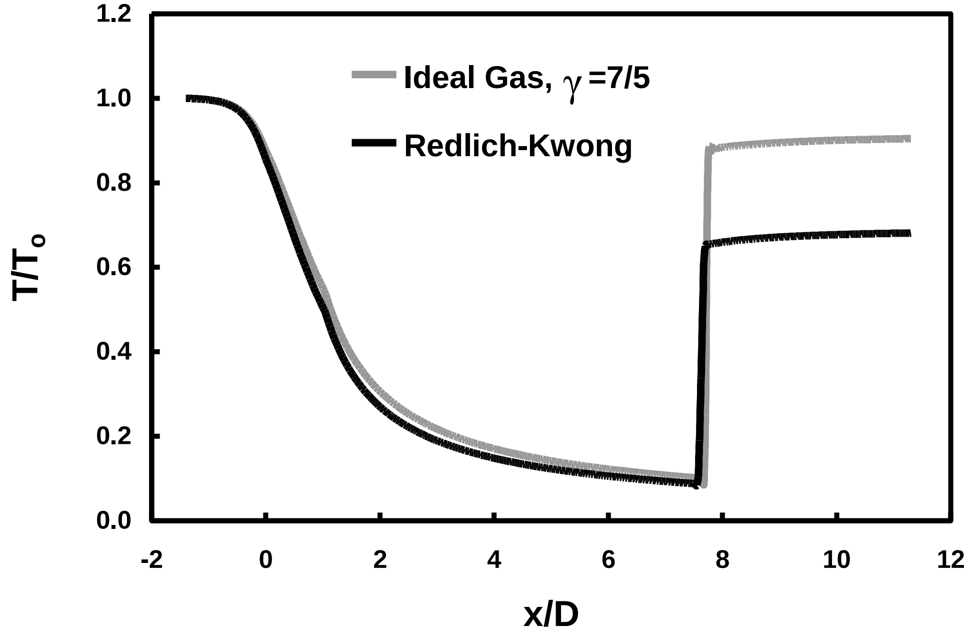


Figure 2.5: QOD temperature profile, CO_2 at T_o= 70°C P_o= 80 bar

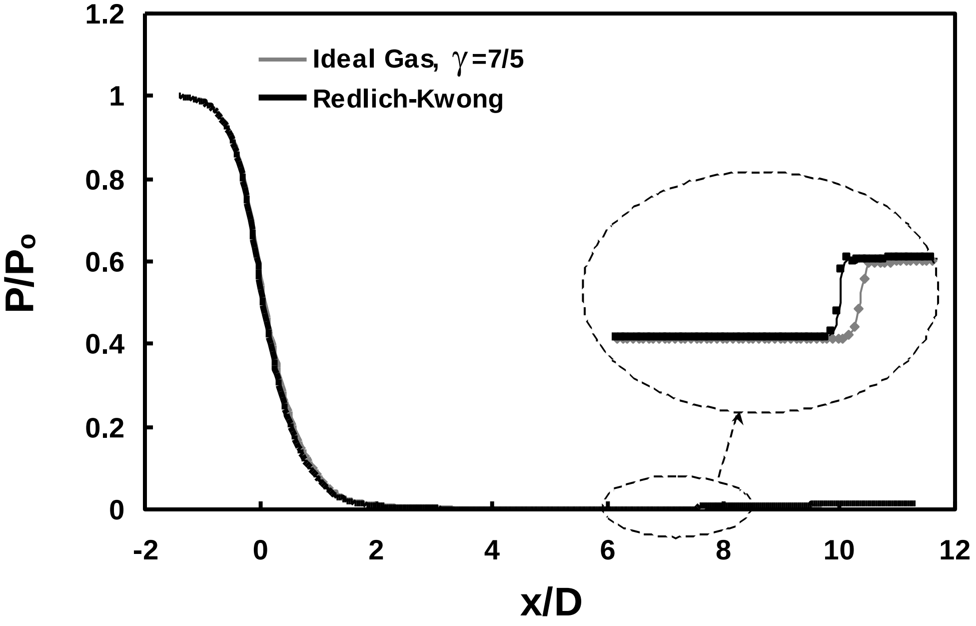


Figure 2.6: QOD pressure profile, CO_2 at T_o= 70°C P_o= 80 bar

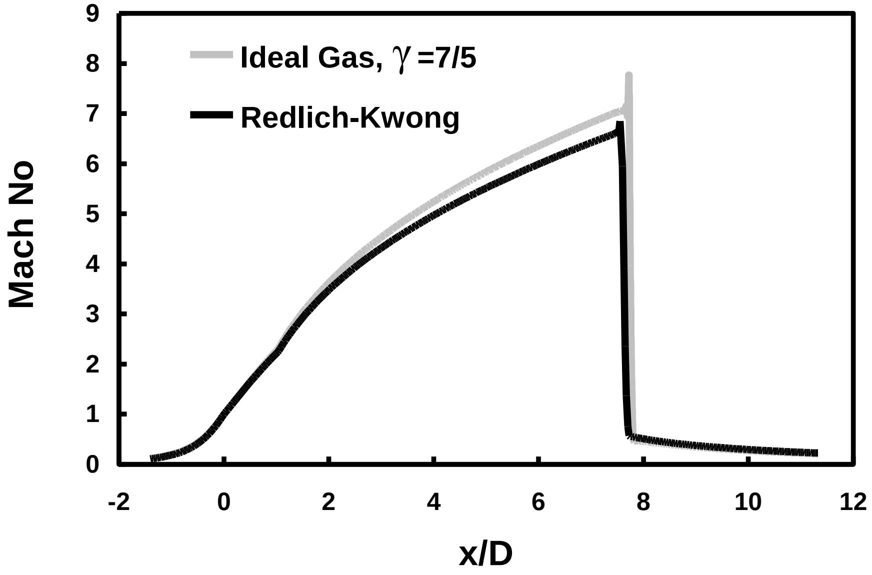


Figure 2.7: QOD Mach number, CO_2 at T_o= 70°C P_o= 80 bar

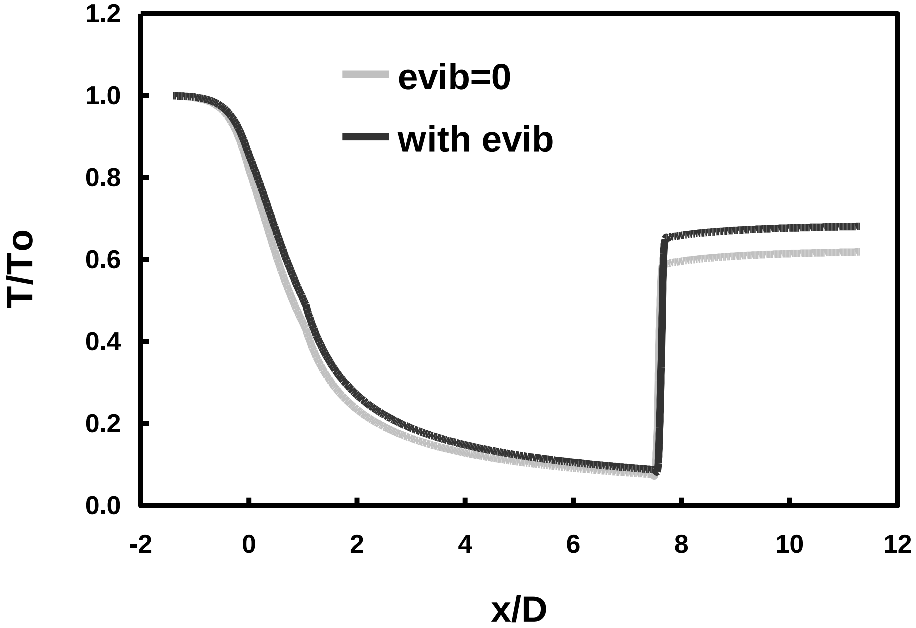


Figure 2.8: QOD temperature profile with and without vibrational modes, CO_2 at T_o= 70°C P_o= 80 bar

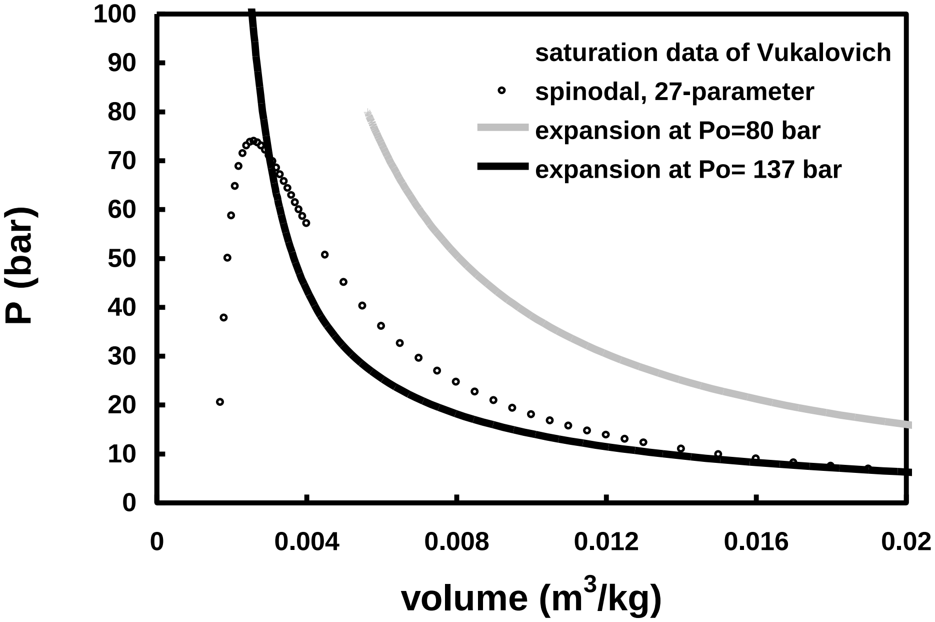


Figure 2.9: QOD expansion PV trajectory

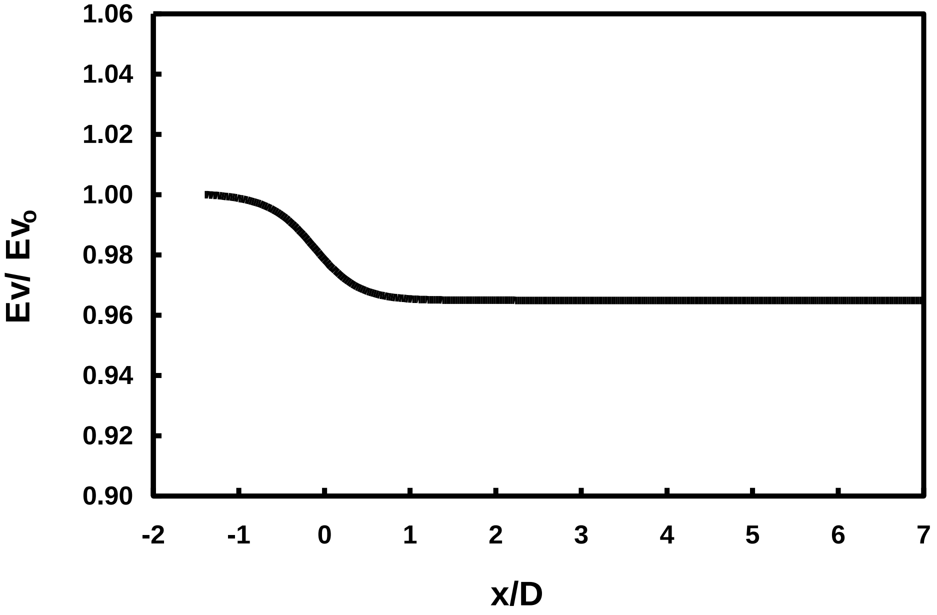


Figure 2.10: Vibrational relaxation of CO_2

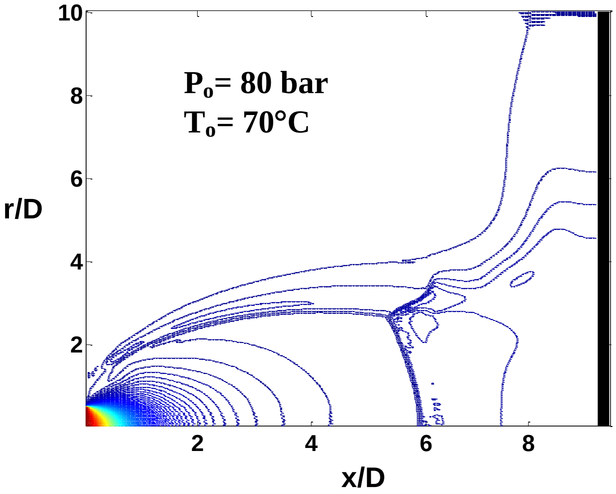


Figure 2.11: ASFJ CO_2 computed density contours, Xp= 9.4D

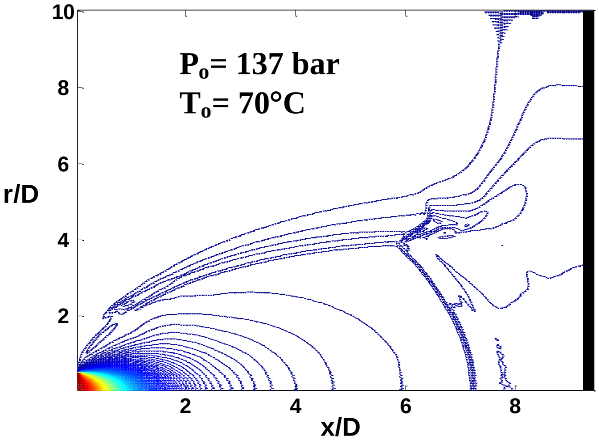


Figure 2.12: ASFJ CO_2 computed density contours, Xp= 9.4D

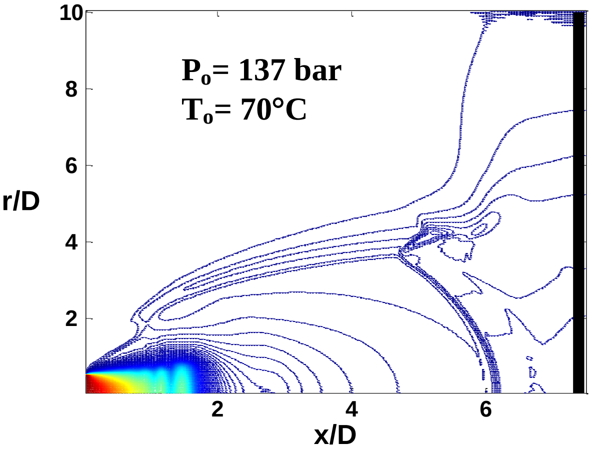


Figure 2.13: ASFJ CO_2 computed density contours, Xp= 7.5D

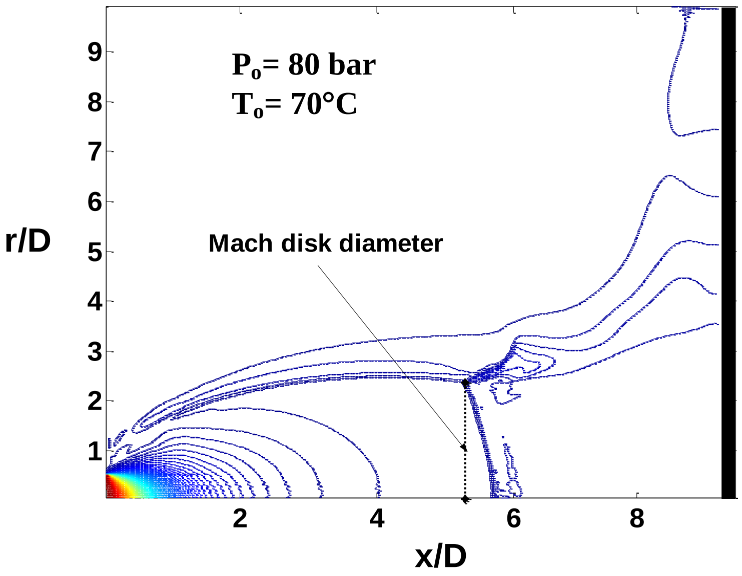


Figure 2.14: ASFJ ideal gas, γ =7/5, computed density contours, Xp= 9.4D

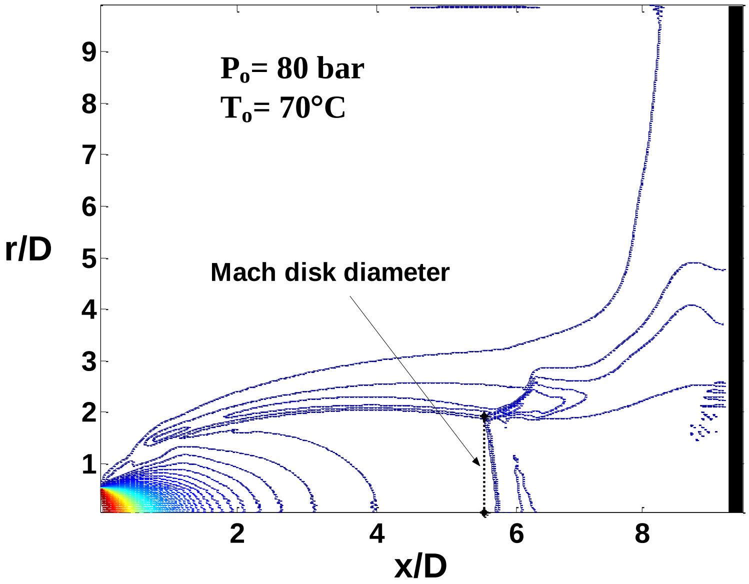


Figure 2.15: ASFJ ideal gas, γ =5/3, computed density contours, Xp= 9.4D

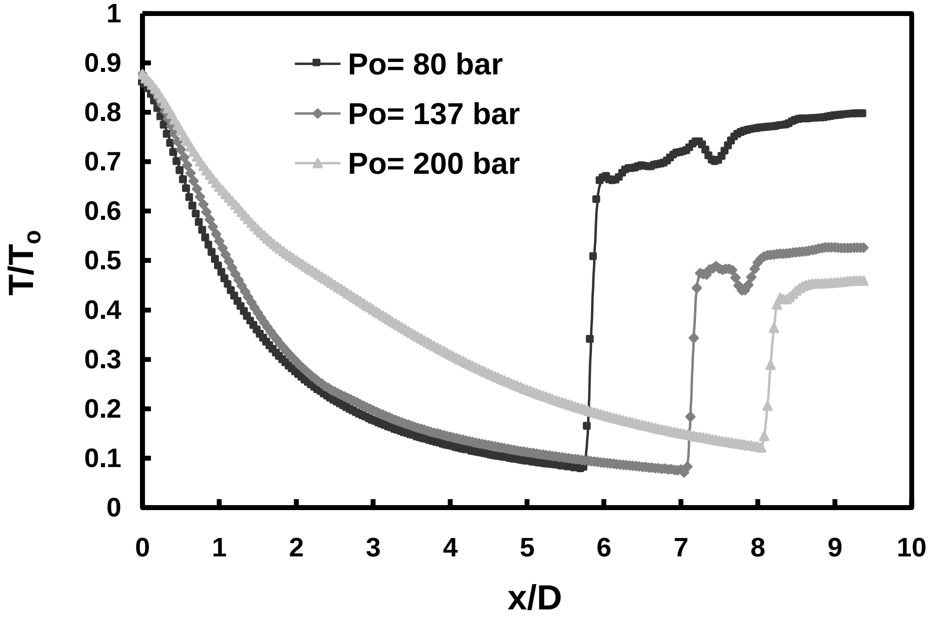


Figure 2.16: ASFJ CO_2 centerline temperature profile

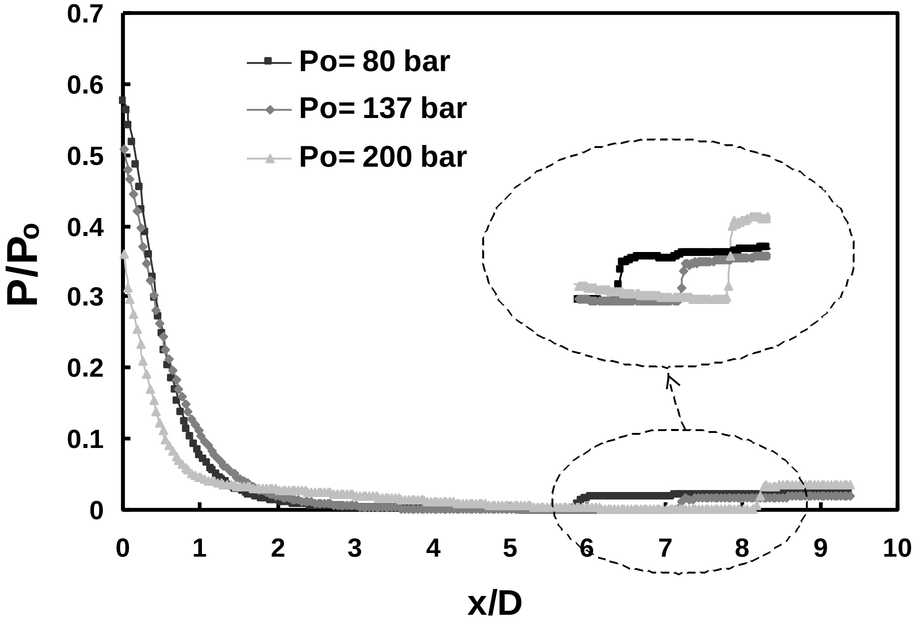


Figure 2.17: ASFJ CO_2 centerline pressure profile

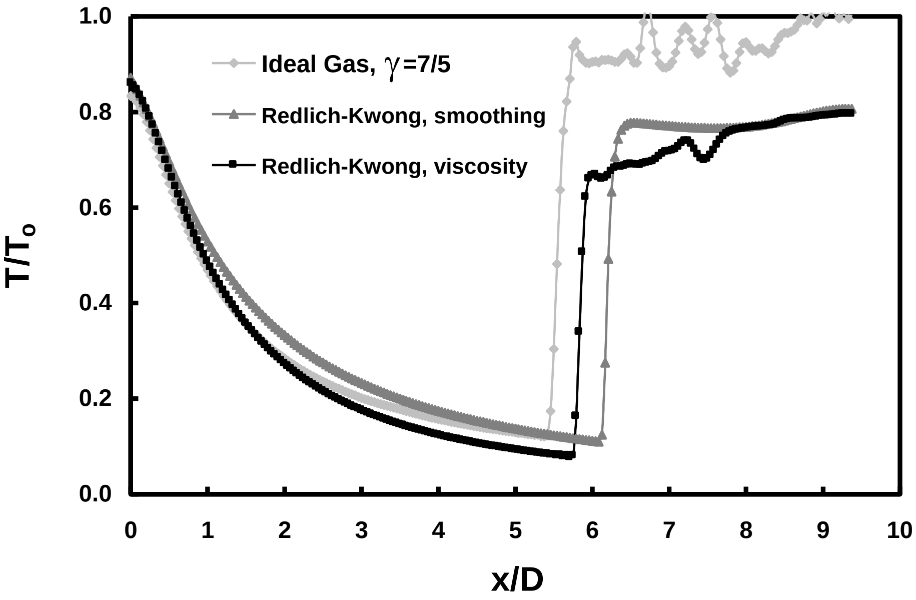


Figure 2.18: ASFJ CO_2 centerline temperature at P_o= 80 bar, T_o= 70°C

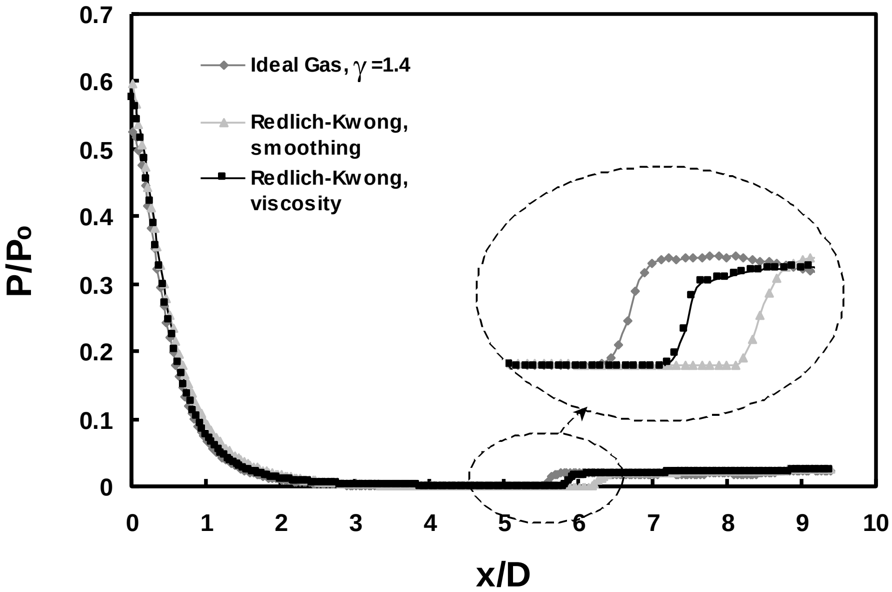


Figure 2.19: ASFJ CO_2 centerline pressure at P_o= 80 bar, T_o= 70°C

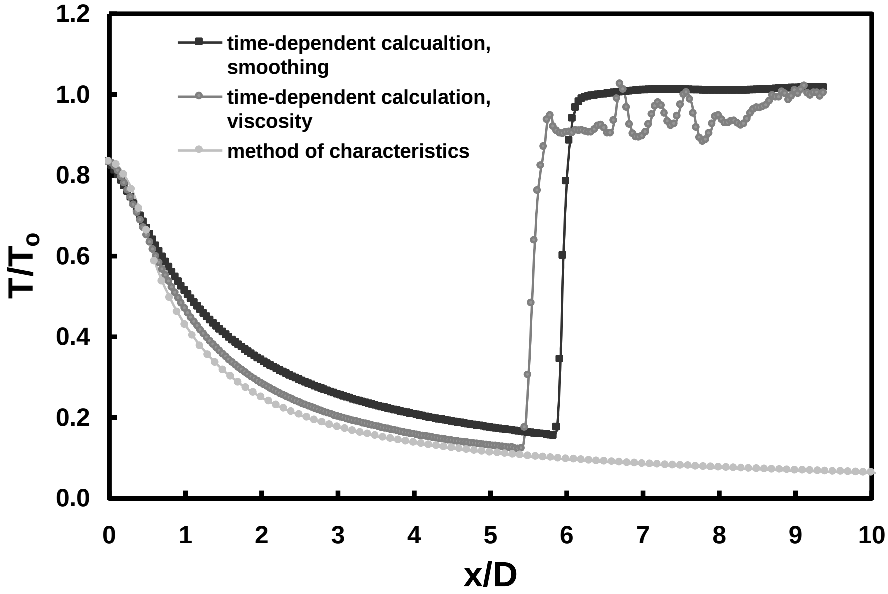


Figure 2.20: ASFJ for ideal gas, γ=7/5; centerline temperature at P_o= 80 bar, T_o= 70°C

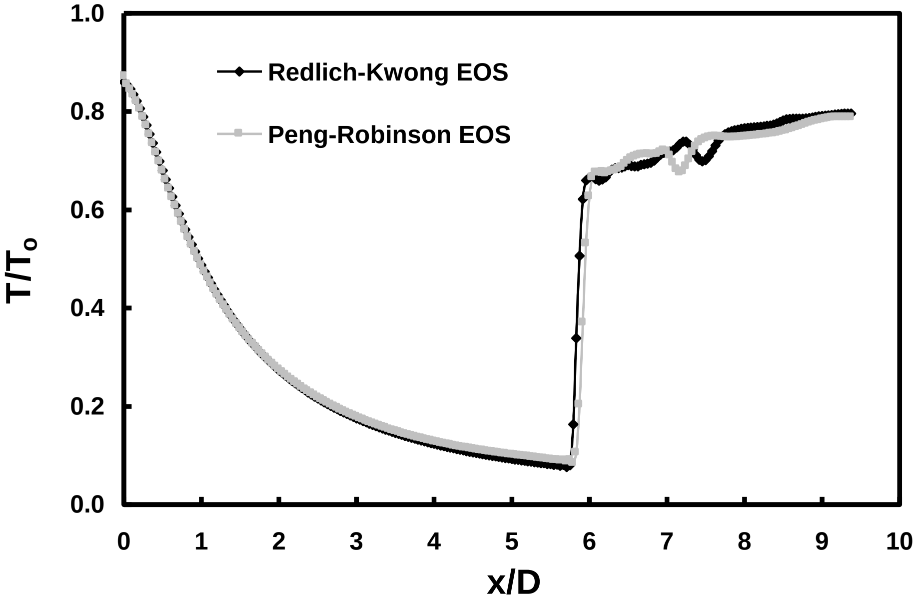


Figure 2.21: ASFJ CO_2 comparing Redlich-Kwong and Peng-Robinson equation of state; centerline temperature at P_o= 80 bar, T_o= 70°C

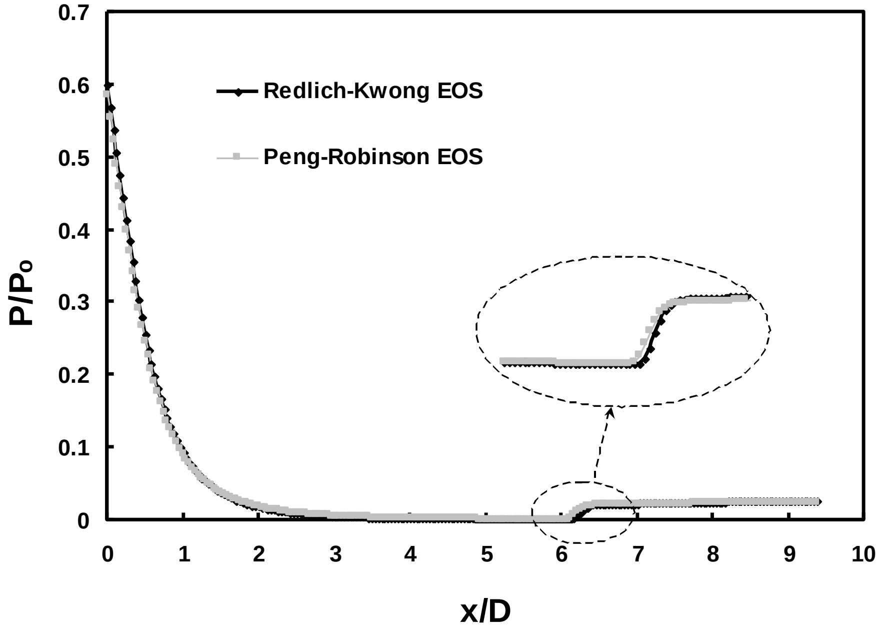


Figure 2.22: : ASFJ CO_2 comparing Redlich-Kwong and Peng-Robinson equation of state; centerline pressure at P_o= 80 bar, T_o= 70°C

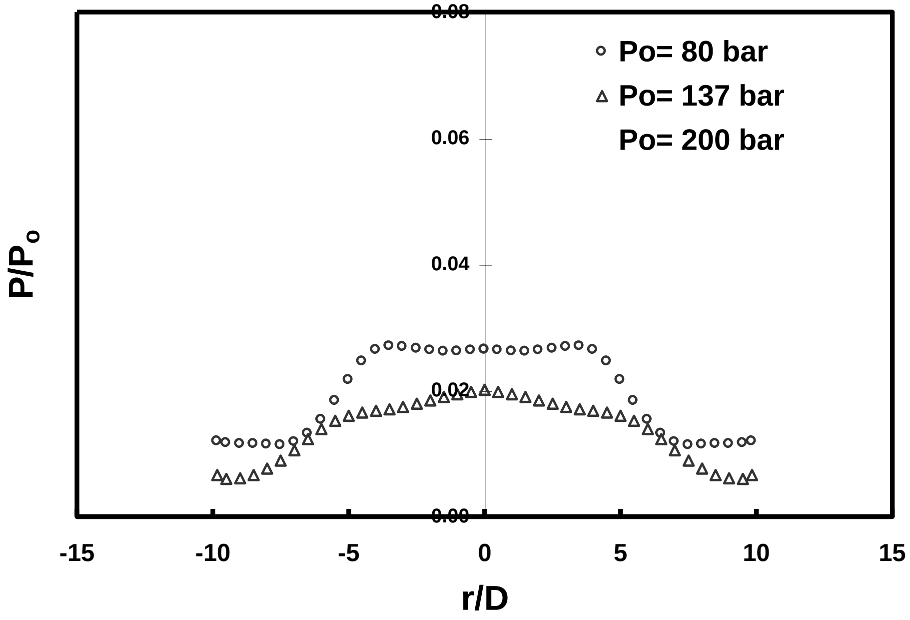


Figure 2.23: Calculated ASFJ CO_2 pressure profile at the plate

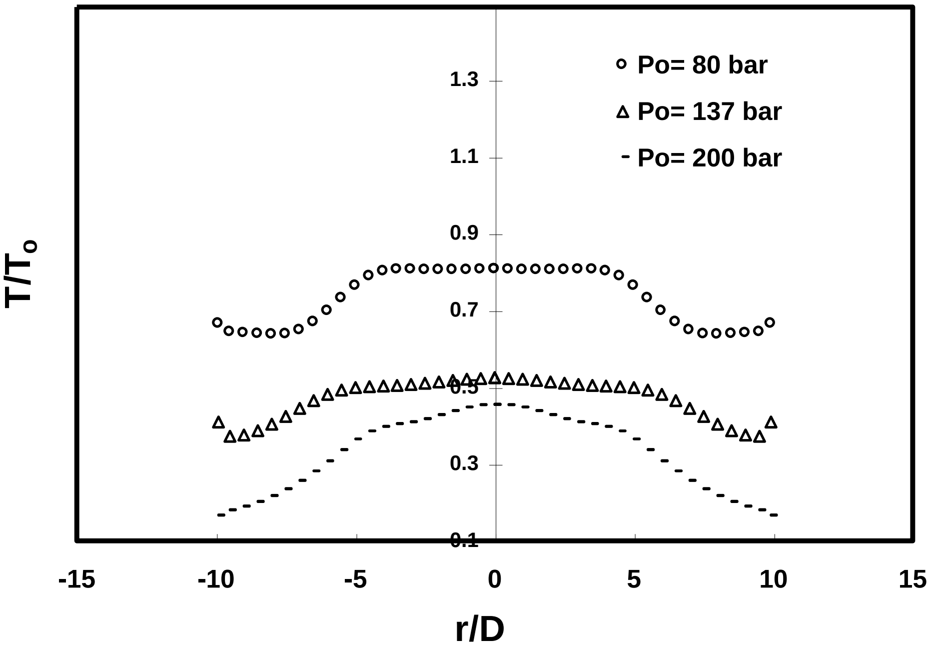


Figure 2.24: Calculated ASFJ CO_2 temperature profile at the plate

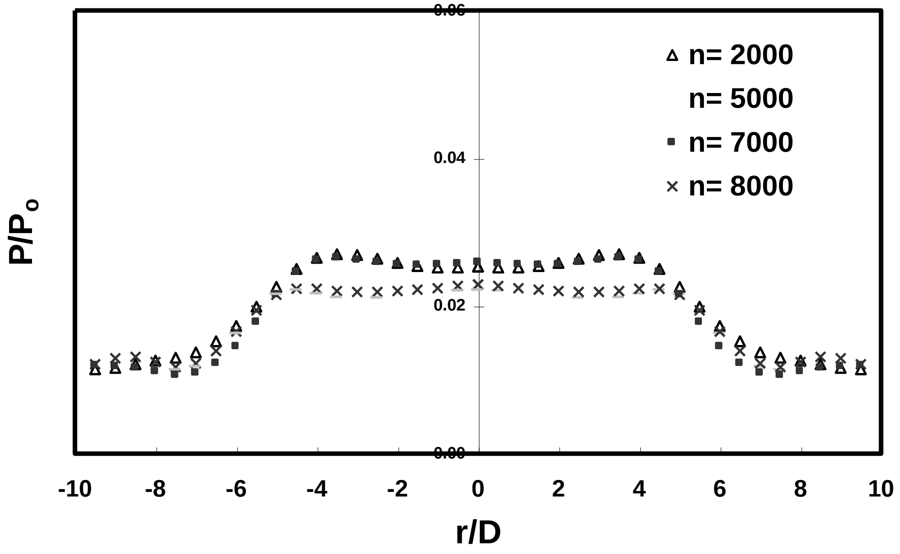


Figure 2.25: Calculated pressure oscillation at the plate at P_o= 80 bar, T_o= 70°C

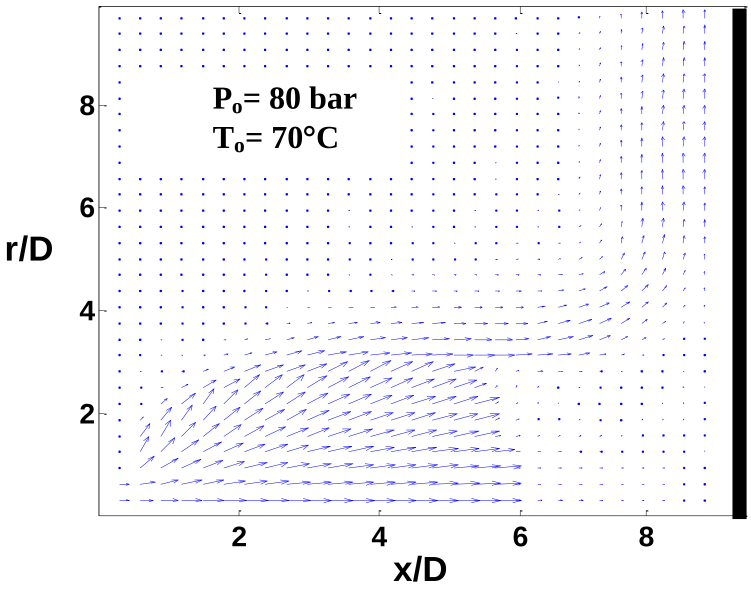


Figure 2.26: CO_2 computed velocity vectors at P_o= 80 bar

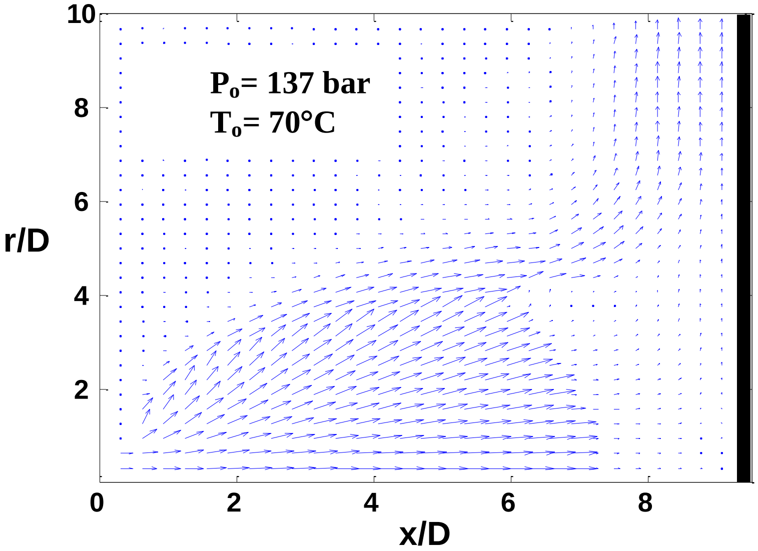


Figure 2.27: CO_2 computed velocity vectors at P_o= 137 bar

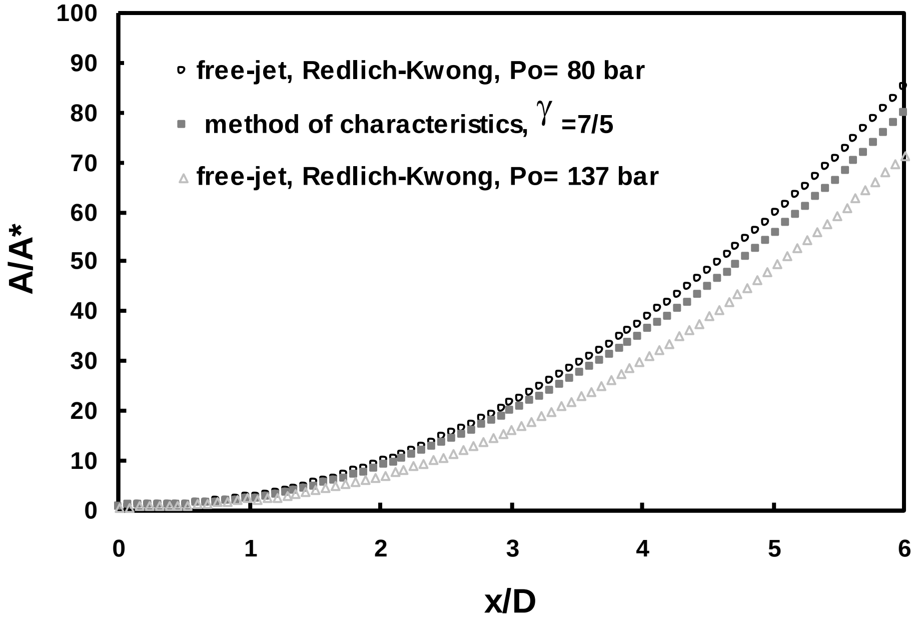


Figure 2.28: Effective QOD nozzle area

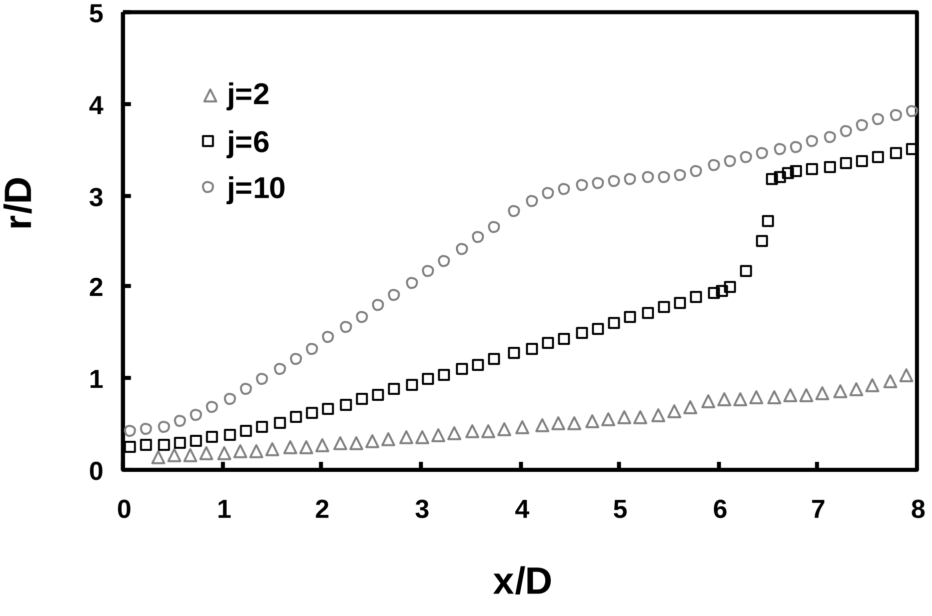


Figure 2.29: Streamlines extracted from ASFJ CO_2 at P_o= 80 bar, T_o= 70°C

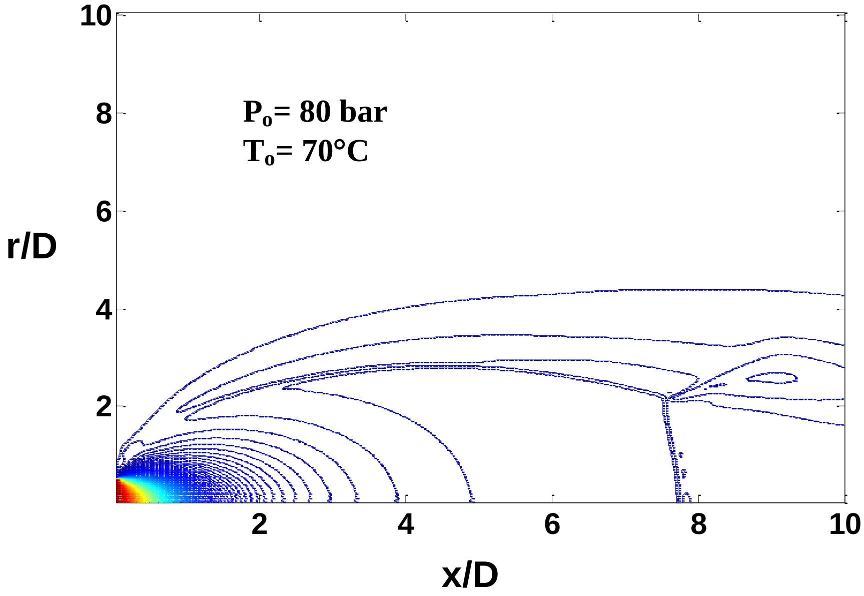


Figure 2.30: Computed density contours for CO_2 ASFJ expansion into atmosphere

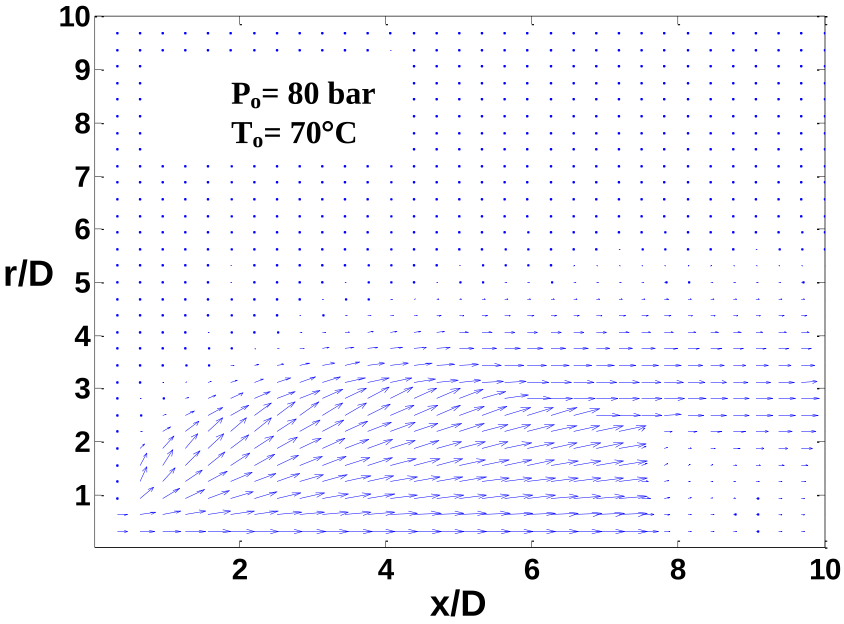


Figure 2.31: Computed velocity vectors for CO_2 ASFJ expansion into atmosphere

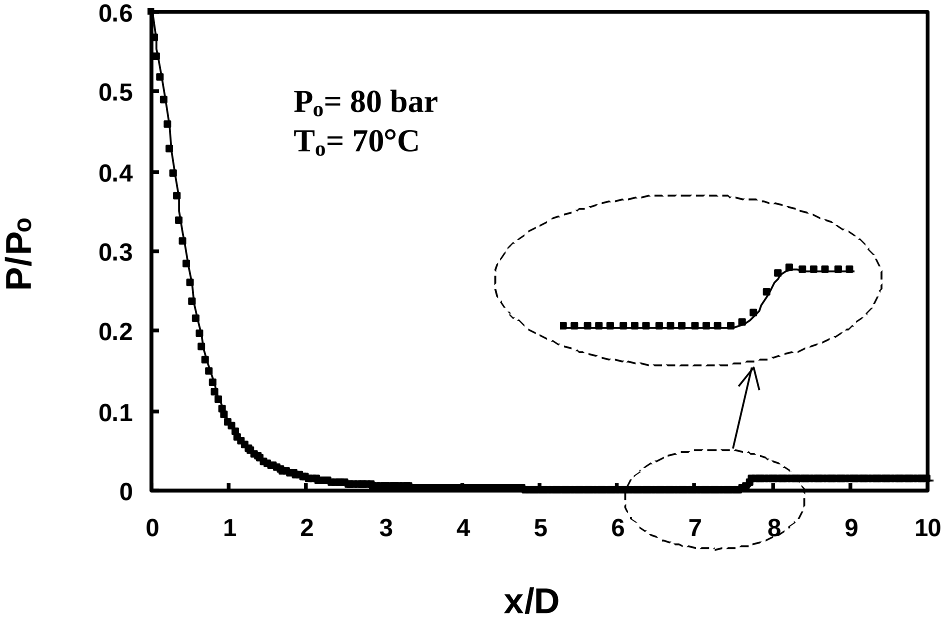


Figure 2.32: Centerline pressure for CO_2 ASFJ expansion into atmosphere

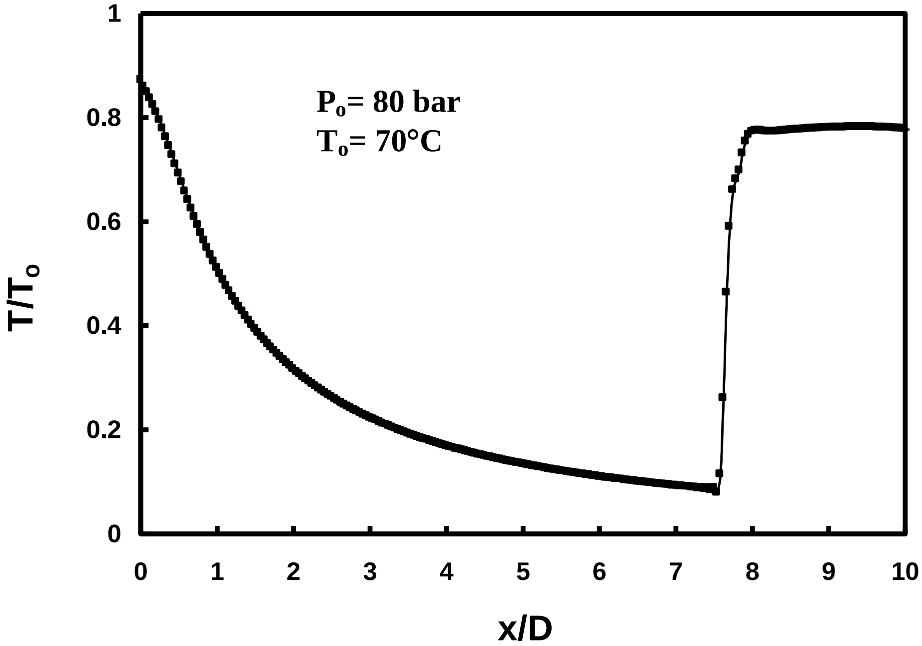


Figure 2.33: Centerline temperature for CO_2 ASFJ expansion into atmosphere

Table 2.1: Comparison between exact and numerical QOD solutions for Redlich-Kwong (RK), Peng-Robinson (PR), and Ideal Gas (IG)

		at throat	**before shock**	**after shock**
Pressure P/P_o	Exact RK	0.54	0.00025	0.012
	Lax-Wendroff RK	0.56	0.00025	0.011
	Exact PR	0.57	0.00024	0.012
	Lax-Wendroff PR	0.55	0.00024	0.012
	Exact IG	0.53	0.00020	0.012
	Lax-Wendroff IG	0.53	0.00024	0.011
Temperature T/T_o	Exact RK	0.83	0.088	0.74
	Lax-Wendroff RK	0.86	0.090	0.69
	Exact PR	0.86	0.084	0.72
	Lax-Wendroff PR	0.85	0.083	0.66
	Exact IG	0.83	0.088	0.97
	Lax-Wendroff IG	0.84	0.080	0.92

REFERENCES

Anderson, J. D., A Time Dependent Quasi-One-Dimensional Analysis of Population Inversions in an Expanding Gas, NOLTR 69-200 (1969)

Anderson, J. D., A Time Dependent Analysis for Vibrational and Chemical Nonequilibrium Nozzle Flows, AIAA Journal **8**, pp 545 (1970)

Anderson, J. D., Modern Compressible Flow, with Historical Perspective, McGraw-Hill (1990)

Anderson, J. D., Computational Fluid Dynamics, The Basics with Applications, McGraw-Hill (1995)

Ashkenas, H., and Sherman, F., The Structure and Utilization of Free Jets in Low Density Wind Tunnels, in Rarefied Gas Dynamics, Vol. 2, pp 84, edited by J. Deleeuw, Academic Press, New York (1966)

Bier, K., and Shmidt, E., Zur Form Der VerdichtungsstöBe in frei expandierenden Gasstrahlen. Zeitschrift für Angewandte Physik **13**(11), pp 493 (1961)

Bradley, J., Shock Waves in Chemistry and Physics, John Wiley, New York (1962)

Candler, G., Hypersonic Nozzle Analysis Using an Excluded Volume Equation of State, University of Minnesota, private conversation

Chung, T., J., Computational Fluid Dynamics, Cambridge University Press (2002)

Forney, L. J., Particle Impaction in Axially Symmetric Supersonic Flow., J. Aerosol Science and Technology **15**, pp 49 (1991)

Hecht, C., Statistical Thermodynamics and Kinetic Theory, Dover Publications (1990)

Hirsch, C., Numerical Computation of Internal and External Flows, John Wiley and Sons, New York (1990)

Hoffman, J. D., Numerical Methods for Engineers and Scientists, McGraw Hill (1992)

Miller, D. R., Free Jet Sources, in Molecular Beam Methods, Vol. 1, pp 14, Scoles, G., Oxford University Press, New York (1988)

Miller, D., Fineman, M., and Murphy, H., The Free-Jet Expansion from a Capillary Source, 13th Int. Symp. On Rarefied Gas Dynamics, Novosibirsk, Russia (1982)

Moretti, G., The importance of Boundary Conditions in the Numerical Treatment of Hyperbolic Equations, PIBAL Report 68-34, Polytechnic Institute of Brooklyn, New York (1968)

Murphy, H., and Miller, D., Off Axis Properties of Free-Jets from Different Subsonic Flows, in 14th Int. Symp. Rarefied Gas Dynamics, edited by H. Oguchi, pp 743, Univ. of Tokyo Press, Tokyo (1984)

Murphy, H., and Miller, D., Effects of Nozzle Geometry on Kinetics in Free-Jet Expansions, J. Physical Chemistry **88**, pp 4474 (1984)

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flanner, B. P., Numerical Recipes in Fortran, Cambridge University Press (1994)

Reverchon, E., and Pallado, P., Hydrodynamic Modeling of the RESS Process, J. Supercritical Fluids **9**, pp 216 (1996)

Saito, T., Nakatsuji, H., Teshima, K., Numerical Simulation and Visualization of Free-Jet Flow Fields, Trans. Jpn. Soc. Aero. Space Sci., **28**, No. 82, pp 240 (1986)

Sandler, S. I., Chemical and Engineering Thermodynamics, John Wiley (1999)

Sinha, R., Zarkay, V., and Erdos, J., Flowfield Analysis of Plumes of Two-Dimensional Underexpanded Jets by a Time-Dependent Method, AIAA Journal **9**, pp 2363 (1971)

Teshima, K., Experimental and Numerical Investigation of a Supersonic Free Jet Impinging on a Perpendicular Surface, in 7th Int. Symp. on Space Tech., Tokyo, pp 765 (1990)

Tien, C., and Lienhard, J., Statistical Thermodynamics, Holt, Rinehart, and Winston Inc. (1971)

Vincenti, W. G., Kruger, C. H., Introduction to Physical Gas Dynamics, Krieger Publishing (1986)

Vukalovich, M. P., and Altunin, V. V., Thermophysical Properties of Carbon Dioxide, London and Wellingborough (1968)

Wark, K., Advanced Thermodynamics for Engineers, McGraw-Hill (1995)

Weber, M., and Thies, M., Undestanding the RESS Process, in Supercritical Fluid Technology in Materials Science and Engineering, edited by Y. Sun, pp387, New York (2002)

Yamazaki, S., Taki, M., and Fujitani, Y., Rotational and Vibrational Relaxation In Free-JetExpansions of CO_2 at Stagnation Temperaturesranging form 300 K to 1300 K, in 12th Int. Symp. Rarefied Gas Dynamics, edited by S. Fisher, pp 802, Charlottesville, Va (1980)

CHAPTER 3

EXPERIMENTAL SET-UP AND RESULTS

3.1 Introduction

We noted above that we have selected CO_2 as the experimental supercritical fluid to investigate in this thesis because it is widely used as a supercritical solvent for Rapid Expansion of Supercritical Solutions (RESS) experiments, the equations of state are well established for CO_2 as well as for mixtures of solutes, and the critical properties are reasonably easy to achieve, $T_c = 31°C$ and $P_c = 73$ bar. The experimental set-up described below is straightforward, utilizing commercial components and diagnostics. Because of the high supercritical pressures, it is necessary to work with very small orifices, less than 100 μm, in order to maintain sufficiently low flow rates for the pumps to control and to have reasonable run times. The experiments are designed to validate the principal features of our calculations, including flow rate, shock structure, and temperature and pressure at the plate. This chapter deals with the orifice source, but we also report some preliminary results for capillary sources below in chapter 4. Once again, we have concentrated on the isentropic orifice source because it provides well defined inlet conditions for our free-jet calculations.

3.2 Experimental Apparatus

Figure 3.1 is a schematic of the experimental set-up, while figure 3.2 is a picture of the set-up. The fluid is compressed in an ISCO Model 100DX commercial piston driven syringe pump, capable of operating at pressures up to 10,000 psi with a 102 ml volume reservoir, and then directed into a VARIAN 8500 pump, capable of operating at 10,000 psi with a 250 ml volume piston. It is possible to operate the ISCO pump at a constant pressure or constant flow rate, while the VARIAN pump can only operate at constant volume flow rate. The piston of the VARIAN pump was redesigned to allow the addition of solute to the pump.

All pressures are measured with Omega strain gauges, accurate to within 0.25%. The fluid leaves the syringe pump and passes through a water bath pre-heater to heat the fluid to the desired supercritical temperature before it enters the source stagnation chamber where a resistive heater controls the final stagnation temperature. A thermocouple enters directly into the stagnation chamber, just upstream of the nozzle, and is used to record the correct stagnation temperature. Figure 3.3 is a schematic of the nozzle block assembly, while figure 3.4 is a picture of the nozzle block showing the high pressure swagelok fitting into which the actual nozzle orifices are directly machined. The small nozzle orifices were made by drilling an aperture in the swagelok cap. We have primarily studied flows from two orifice diameters, 100 μm and 50 μm. The pumping capacity permitted us to reach up to 100 bar with the 100 μm orifice and

200 bar with the 50 μm orifice. A typical length over diameter of the orifices is five. The connections tubing used are standard 1/8″ stainless steel with an inner diameter of 0.055″. All of the tubing connections are made with standard 1/8″ swagelok stainless steel fittings, and 7 μm filters are used to prevent any particles from clogging the orifice. The size of the nozzle orifice is a compromise between being large enough to permit some resolution of the shock structure below and being small enough to be able to reach high pressures at reasonable flows with the pump. The system is fully capable of reaching pressures up to 500 bar and temperatures up to 200°C. For the present CO_2 studies we have operated at temperatures near 70°C and pressures between 80 bar and 200 bar.

The experimental procedure is as follows: The water bath is heated to 70°C with an emersion resistive heater. A thermocouple is submerged inside the water bath to monitor continually the temperature of the bath. The length of the tubing that needed to be submerged in the hot water bath, in order to bring the temperature of flowing fluid from room temperature to within a few degrees of the desired source temperature 70°C, was calculated to be at least 100 cm. To maintain the temperature of CO_2 at 70°C between the exit of the bath and the inlet of the orifice, we use a resistive heater inserted inside the nozzle block. The voltage and current are supplied to the resistive heater with a Hewlett Packard 6038A system power supply. The temperature of the stagnation chamber in the nozzle block is also monitored constantly during operation, with a type K Omega thermocouple. Carbon dioxide, purity 99.9%, was supplied from a syphon type gas cylinder. After ensuring that all valves are closed (see figure 2.1) and heaters have been brought to the desired temperature, valve V_1 is opened to fill the ISCO pump and CO_2 is pressurized to the desired supercritical pressure. Valve V_1 is then closed and valve V_2 is opened to direct the high pressure CO_2 to the VARIAN pump. This procedure is repeated until the piston in the VARIAN pump is filled with the high pressure CO_2. Finally, we set the desired flow rate at the VARIAN pump and valve V_3 is opened to allow the pressurized CO_2 to flow inside the hot bath and to reach the desired stagnation temperature.

A simple shadowgraph system is focused on the free-jet expansion and utilizes a lamp light source, a 10 cm diameter convex-convex lens, and a Nikon camera with a 28-108 mm zoom lens which made it possible to take an enlarged image of the shock structure with a maximum f/3.5 aperture. A typical arrangement for the shadowgraph is shown in figure 3.5. The lens makes the light parallel before passing through the gas. The intensity of the light passing through the gas is a function of the density variation in the gas due to the density dependent refractive index (Liepman 1947). The shadowgraph technique is sensitive to the change in spacial gradient of density. The deflection of the light rays is proportional to the spacial gradient $\partial\rho/\partial x$. In a region where $\partial\rho/\partial x$ is constant, the deflection of the rays is constant and no change in light intensity results on the screen. However, in a region where $\partial\rho/\partial x$ is not constant, that is $\partial^2\rho/\partial x^2$ is not zero, the deflection of the rays is not constant. This occurs in particular at the shock wave where the density rises very suddenly causing a high curvature in the density variation and a dark line is produced on the screen.

The plate used in the experiment is a 6.5 cm by 6.5 cm aluminum plate, 1.5 cm thick. The face of the plate is polished and perfectly smooth and flat. The plate is attached to a micrometer device to allow sensitive positioning of the plate to the accuracy of 10 μm, or one tenth of a nozzle diameter, when D = 100 μm. The vertical plate can be put as close as three nozzle diameters from the source and then removed to study the free-jet into the atmosphere.

Figure 3.6 shows the experimental set-up for the pressure measurement at the plate. An Omega strain gauge low pressure transducer (model PX602) is inserted from the back of the plate and opened to the front surface through a 100 μm orifice, minimizing the dead space in front of the strain gauge. The voltage and current are supplied to the pressure transducer with a Hewlett Packard 6634A system DC power supply. The plate was moved vertically, horizontally, and axially in order to obtain a profile.

Measurement of temperature is much more difficult because of heat transfer effects, both at the plate and in the thermocouple bead itself. We did not make any corrections for heat transfer so our data must be considered to be qualitative, and as we show it is the least satisfactory data with which to compare our calculations. We did compare plastic plates with metal plates to ascertain the magnitude of the heat transfer effect. To measure plate temperature, a 250 μm sheath metal thermocouple (Omega KMTSS-010G) shown in figure 3.7 is inserted (slip-fit) from the rear until it was flush with the front of the plate surface. We also made measurements with the thermocouple protruding about 100 μm in front of the plate. Again, to obtain some measure of the heat transfer effect, a 50 μm type K thermocouple (Omega CHAL-002) was built by inserting the 2 wires inside a two-hole ceramic rod (Omega TRM-005132). The inside diameter of the holes is 0.005″ which allows the wires to be held tight in place in front of the free-jet. The outside diameter of the ceramic rod is 1/32″ and is inserted (press-fit) into the plate. The size of the thermocouple opening in the plate, together with the heat transfer effects, made it difficult to resolve the temperature profiles (see below). The resulting temperature profiles were similar for the three techniques and most of our results were taken with the flush mounted 250 μm thermocouple.

As we noted above, the calculations predict that the real gas effects are most significant for temperature downstream of the Mach disk shock. Therefore, in order to obtain some idea of the temperature of the flow in the free-jet, we suspended a small 100 μm thermocouple bead onto two 50 μm wires and moved it directly into the jet, probing through the Mach disk as shown in figure 3.8. Figure 3.9 shows a picture of this probe in which the bow shock structure in front of the probe is seen by the shadowgraph. We discuss the results for this probe below, but clearly heat transfer effects are also important for this measurement.

3.3 Experimental Results and Comparisons with Theory

As described above in chapter 2, the flow rate of supercritical CO_2 through the nozzle orifice is accurately calculated using any of our equations of state, by assuming isentropic flow from stagnation conditions to choked sonic flow at the orifice exit or throat of the expansion. This flow rate is also the most accurate experimental measurement because of the quality of the piston driven displacement syringe pump. Nozzle diameters were calibrated by measuring the flow rates for ideal gas argon expansions, for which the equation of state is exact, and comparing with exact calculations. Heat transfer is negligible in such short nozzles so that the flow is adiabatic. We verified this by comparing runs with the nozzles completely emerged in a constant temperature bath to runs with the nozzle suspended in the ambient air. It is also necessary to convert the flow rate at the pump, which is a displaced volume flow rate, into a mass flow rate by using the equation of state to convert the measured temperature and pressure into a CO_2 density. In all cases we did this consistently for each equation of state. That is, if our downstream analysis were based on the Redlich-Kwong equation of state, then we used the Redlich-Kwong equation to determine properties at all locations, including the pump, nozzle stagnation chamber, the nozzle exit or throat, and in the free-jet itself.

Even though the core of the nozzle flow is isentropic, there is a small viscous boundary layer which causes the flow diameter to be slightly less than the geometric diameter. In order to determine the effective flow diameter, we made accurate flow rate measurements with argon and compared them with the isentropic ideal gas calculation to obtain the flow diameter, the only unknown in the calculation. Typically this gave us a discharge coefficient between 0.9-1.0, which we then used in our CO_2 calculations. The obvious approximation is that the boundary layers are similar for the ideal gas and the real CO_2 gas. The Reynolds number for the supercritical CO_2 flows is of order 10^5 while for the argon flows is of order $5x10^5$. We also used the computed CO_2 flow rates and compared them with the measured rate in order to set a discharge coefficient. The two methods gave similar results and we feel our calibrations should be valid to within 5%.

Figures 3.10, 3.11 and 3.12 show data and theoretical comparisons for several source conditions and three different equations of state. The orifice diameters indicated have been corrected for the discharge coefficients and are nominally 100 μm and 50 μm in size. Clearly our calculations are quiet reasonable for the flow rates, and we feel confident that they provide us with good inlet conditions for our free-jet calculations. As with the QOD calculations the differences between the equations of state are not substantial and we have therefore proceeded to utilize the simplest Redlich-Kwong equation state for most of our axisymmetric free-jet (ASFJ) calculations. The accuracy of the experimental data does not warrant a more rigorous equation of state, like the 27-parameter model or the Peng-Robinson.

Figure 3.13 is a shadowgraph of supercritical CO_2 expansion from a 100 μm orifice into the atmosphere, showing the shock structure. From the shadowgraph pictures, we can measure the position $(x/D)_{MD}$ of the Mach disk shock waves to within 0.25 orifice diameters. To verify the accuracy of our measurements, we compared results for the Mach disk location for the argon free-jet expansions into the atmosphere without a plate to that of well established ideal gas results in figure 3.14. The Mach disk location for ideal gases is insensitive to the specific heat ratio and given by $(x/D)_{MD} = 0.67\ (P_o/P_b)^{1/2}$, where P_b is the background ambient pressure (Miller 1988).

Figure 3.15 shows the experimental Mach disk location for the supercritical CO_2 free-jet expansion into the atmosphere together with numerically calculated results, and the ideal gas correlation $(x/D)_{MD} = 0.67\ (P_o/P_b)^{1/2}$. The calculated results are good except at the highest source pressure. Remarkably, the ideal gas approximation appears to be a good first order approximation for the supercritical CO_2, a result useful to RESS researchers. Pictures of the expansion into the atmosphere at different stagnation pressures are shown in figure 3.16. Only the jet at $P_o = 200$ bar was visible, due to the considerable condensation behind the Mach disk when the pressure is high. For this high pressure, the shadowgraph shown in figure 3.17 also indicates a different shock structure than that at lower pressure. Condensation at these higher source pressures may account for some of the poor agreement with theory.

Figure 3.18 is a shadowgraph of a supercritical CO_2 expansion from a 100 μm orifice impacting a flat plate at Xp/D = 7. Figures 3.19 thru 3.21 show the experimental shock position as a function of the plate position for CO_2 together with our numerical results at three different source pressures, P_o = 80, 137 and 200 bar respectively. It is clear that the Redlich-Kwong CO_2, ASFJ numerical calculations provide reasonable results for the shock position, even at the highest pressure. Figure 3.22 is a shadowgraph of supercritical CO_2 expansion impacting a plate placed at 9.4 nozzle diameters and at stagnation pressure $P_o = 200$ bar showing clearly the condensation occurring downstream of the Mach disk. It is therefore somewhat surprising that the 200 bar case agrees so well for the shock location given that the observed condensation resulted in the calculations poorly matching the experimental data for pressure and temperature at the plate (see below).

Figure 3.23 compares the shock position as a function of plate distance, for argon and CO_2, at a stagnation pressures near $P_o = 80$ bar. At these lower source pressures, it again appears that ideal gas results should provide a reasonable approximation for supercritical CO_2.

Figure 3.24 shows examples of the pressure profiles at the plate for experimental and calculated results, at 80 and 137 bar, for the plate located at Xp = 9.4D. The agreement between experiment and calculation is reasonable at these pressure values and might be improved by the addition of viscous effects. Figure 3.25 shows the pressure profiles at 200 bar, where the experimental agreement with the calculated result is poor and, as mentioned above, is likely due to condensation effects observed downstream of the Mach disk in the subsonic flow impacting the plate. Because of its complexity, condensation has been ignored in our calculations, even though it is known that the expansion enters into the two phase regions of supercritical CO_2. We discuss our preliminary analysis of condensation in chapter 5.

Figure 3.26 shows the data for pressure profiles at two different plate distances, Xp = 7.5D and Xp = 9.4D for the expansion of supercritical CO_2 at $P_o = 137$ bar and $T_o = 343$ K. The pressure, as expected, increases when the plate is closer to the source, but the agreement with our calculations is reasonable at both plate locations.

Figures 3.27 and 3.28 show data comparing experimental and computed temperature profiles on the flat plate near the centerline. Again, there are no heat transfer corrections made to this data, so the calculated temperature is just that of the fluid and due to the effect of the plate on the expansion. Figure 3.27 shows the condition where there is the closest match and figure 3.28 shows the case of the poorest. Again, we believe that as the pressure increases, the agreement with the calculation deteriorates because of condensation effects which would warm the flow due to the heat release. We had very similar data using the two different sized thermocouples discussed above. We did find that using the plastic plate produces a colder temperature reading, which is closer by 10% to the Redlich-Kwong calculations for CO_2 at 200 bar, because the heat transfer with the plastic plate is reduced. A closer look at the details of the temperature profiles near the centerline at the plate show that the agreement is even poorer than shown by the broad profiles. Figure 3.29 shows more detail for the 80 bar case, indicating that the calculation does not reproduce the undulation indicated by the data near the centerline. It will be a challenge to both our calculations and experiments to improve and understand these plate profiles.

Again, to provide data for a fluid where the complications of the real gas effects are absent we also have measured and computed a few ideal gas argon plate profiles. Figures 3.30 and 3.31 show the pressure and temperature profiles at the plate for an ideal gas argon expansion at T_o= 22°C and P_o= 80 bar, and compares the experiments to the calculations for an ideal gas, $\gamma = 5/3$, at the same stagnation conditions. The pressure profiles agree nicely, as for the supercritical CO_2 expansions above. The calculated temperature at the centerline was equal to the stagnation temperature, as expected for an ideal gas, while the experimental measurements taken with the large diameter thermocouple recorded a lower value, likely due to heat transfer. As before, the off-axis profiles are in disagreement between the calculation and experiment.

The computed pressure and temperature plate profiles all have a circular core of high pressure and temperature which then fall to lower values near a radial distance of $r/D \sim 5$. Researchers have often reported rings of solute deposits in RESS type experiments. It is possible that the coupling of solute precipitation in the fluid phase to our temperature and pressure profiles could help explain such phenomenon.

Qualitative temperature measurements within the free-jet are especially difficult not only because of probe heat transfer effects and the small physical scale of these experiments, but because in the supersonic regime the thermocouple probe induces a bow shock in front of itself as shown in figure 3.9. Nevertheless, because the flow temperature depends critically on the supercritical fluid equation of state and possibly on condensation kinetics, it was important to obtain at least a qualitative picture of the temperature profile. Figure 3.32 shows a profile obtained by moving the thermocouple along the jet axis, as described above. The shadowgraph camera probe was used to examine the distances and to observe the bow shock, which typically was less than 0.25D in front of the thermocouple probe when the probe penetrated the Mach disk. The CO_2 expansion is for 70°C and 80 bar through the 100 μm orifice with the Mach disk at 5.8D. There are no corrections for probe heat transfer effects to the subsonic flow around the thermocouple, and we assume the small thermocouple bead equilibrates with the fluid. The calculated points upstream of the Mach disk assume the probe temperature would be close to the flow temperature just downstream of the induced bow shock along the jet axis. Using the experimental bow shock location, we then used our shock calculations to obtain the flow properties before and just after the shock; we did not solve for the two-dimensional flow downstream of the bow shock and around the probe. In the subsonic flow downstream of the Mach disk, it was assumed the probe reflected the numerically calculated local temperature along the jet axis. Clearly the data and calculations have several caveats but other than the displacement in axial position, which we do not yet understand, it qualitatively supports the calculations.

In summary, the calculations appear to reproduce the flow rate and jet structure reasonably well, including the pressure at the plate surface. The only deficiency is with the fluid temperature. While the limited data for temperature within the jet is encouraging, the data for the temperature at the plate is not satisfactory, and will require the computational acknowledgement of heat transfer between the fluid and the plate, and within the plate.

We conclude that our numerical treatment is able to include the real gas effects for supercritical fluid expansions. The extension to solutes using the equations of state we have tested should be straightforward. However, the important caveats associated with condensation are evident in our experiments and represent a future course of study. We provide some preliminary work in chapter 5.

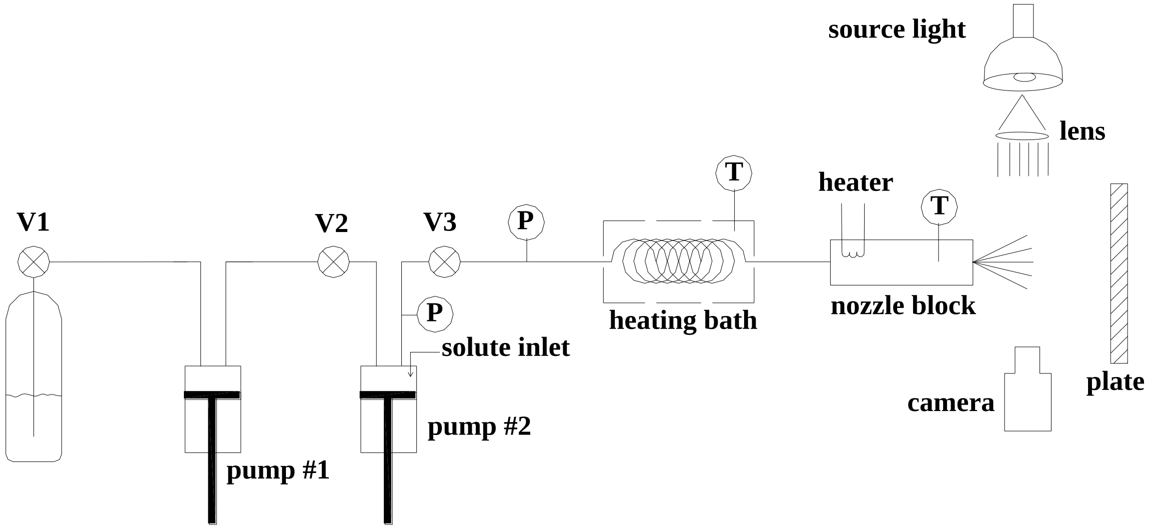


Figure 3.1: Schematic of experimental set-up

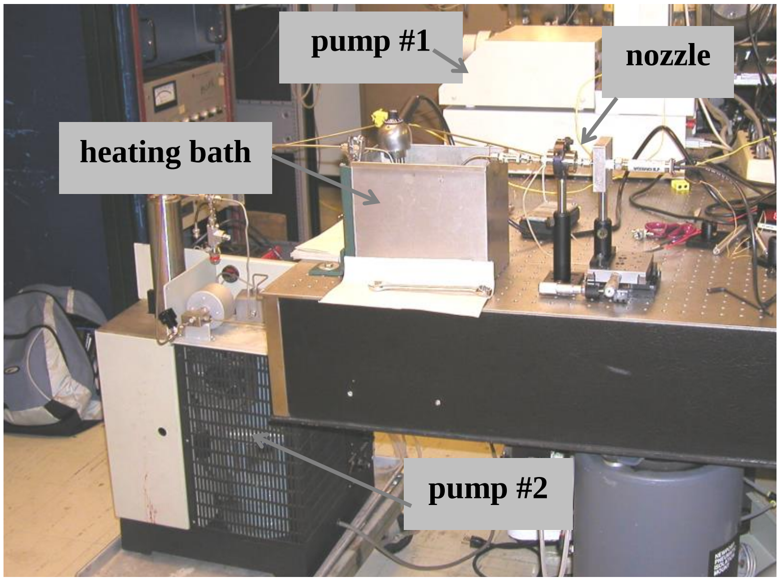


Figure 3.2: Picture of the experimental set-up

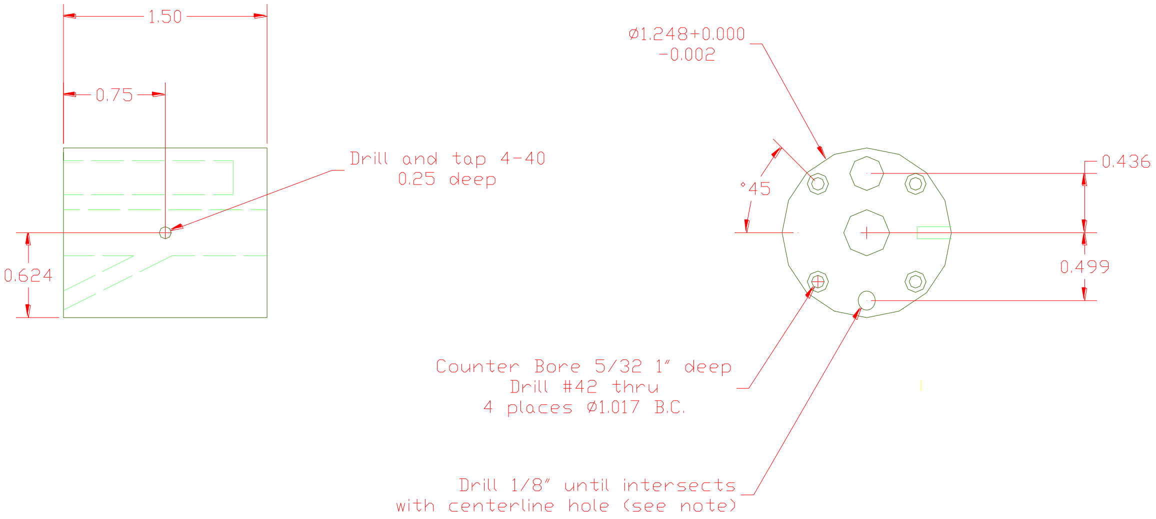


Figure 3.3: Mechanical drawing of the nozzle block

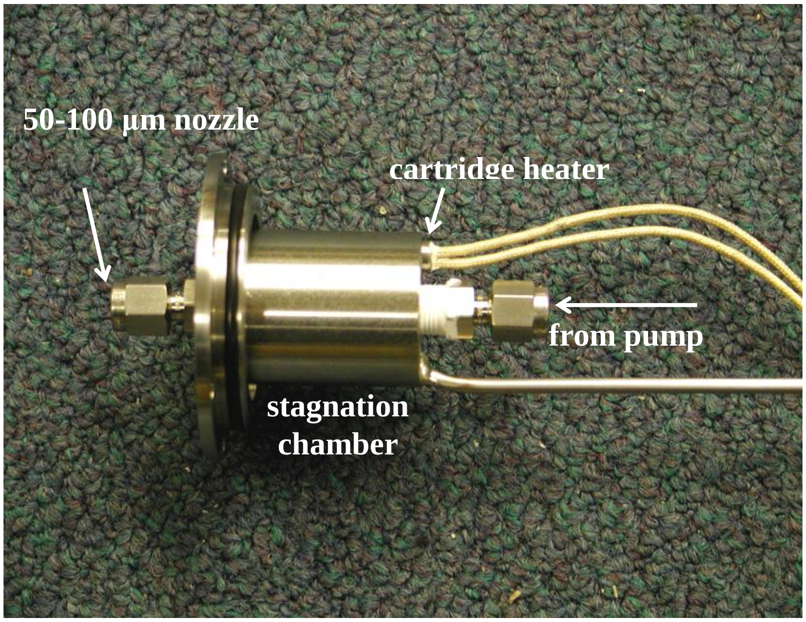
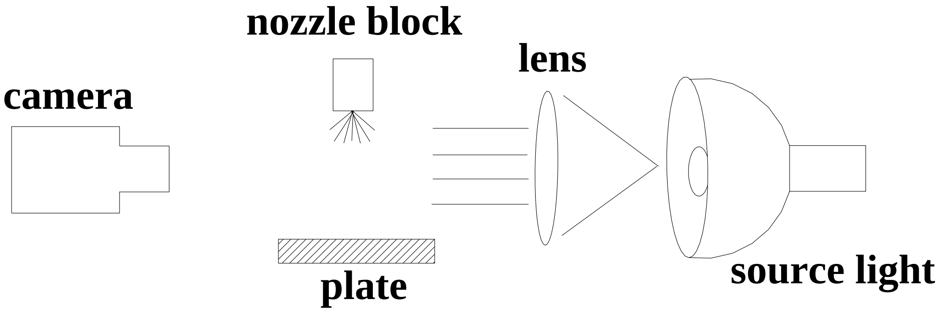


Figure 3.4: Picture of the nozzle block



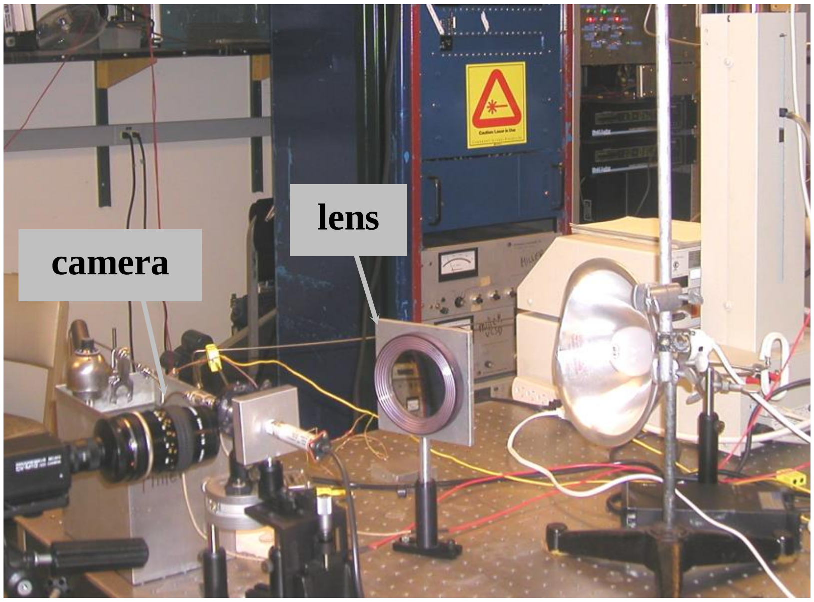


Figure 3.5: Shadowgraph technique

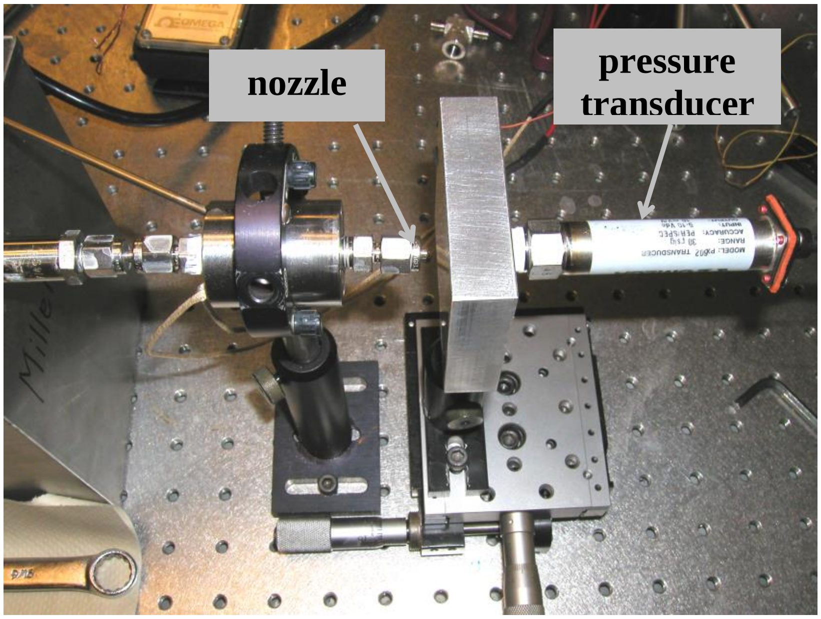


Figure 3.6: Set-up for pressure measurement at the plate

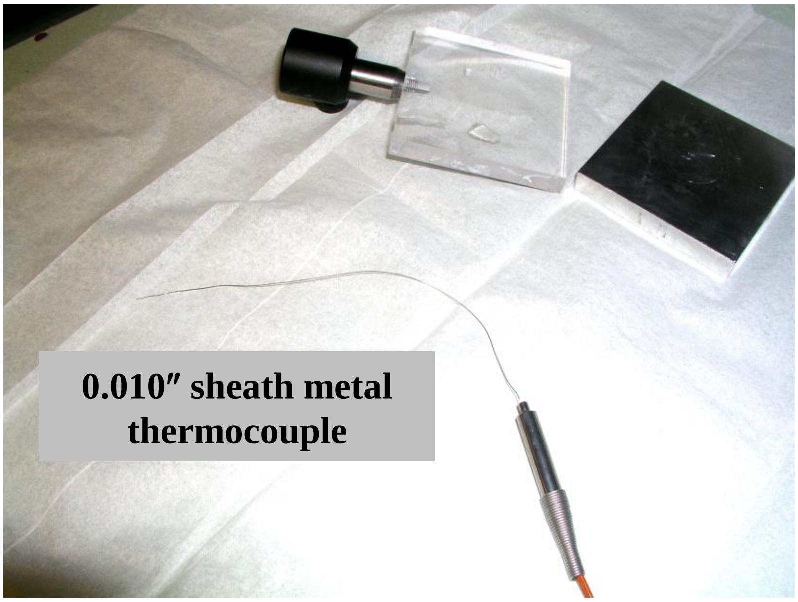


Figure 3.7: Set-up for temperature measurement at the plate

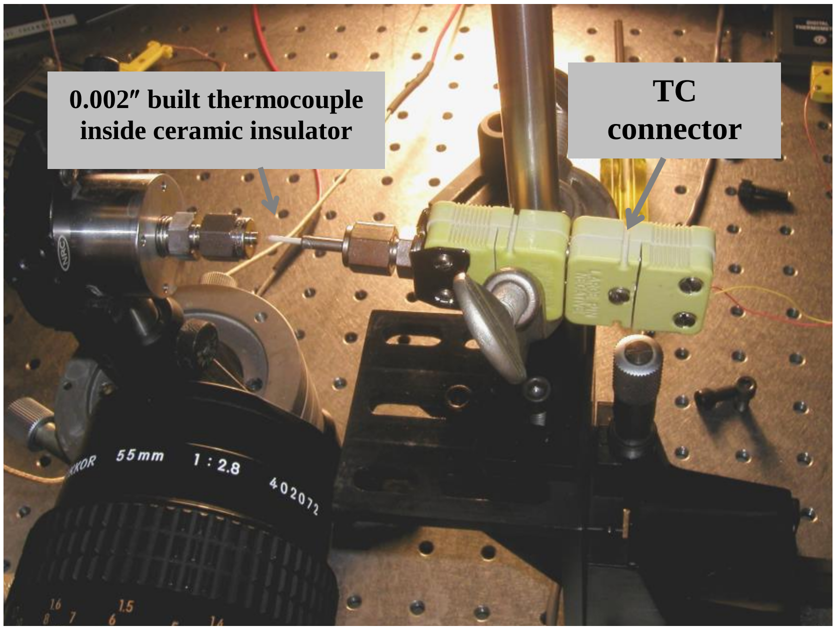


Figure 3.8: Set-up for temperature measurement of the ASFJ into the atmosphere

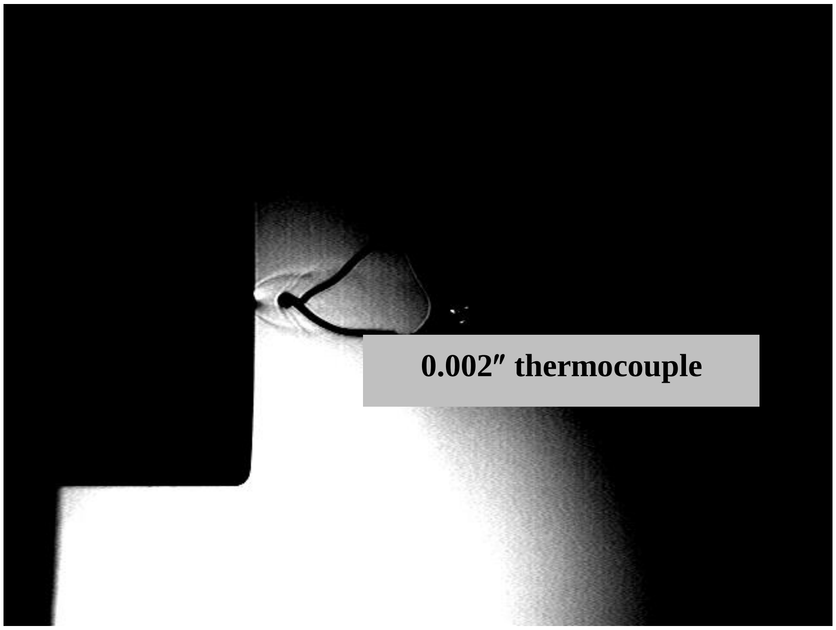


Figure 3.9: Shadowgraph of temperature measurement of the ASFJ into the atmosphere

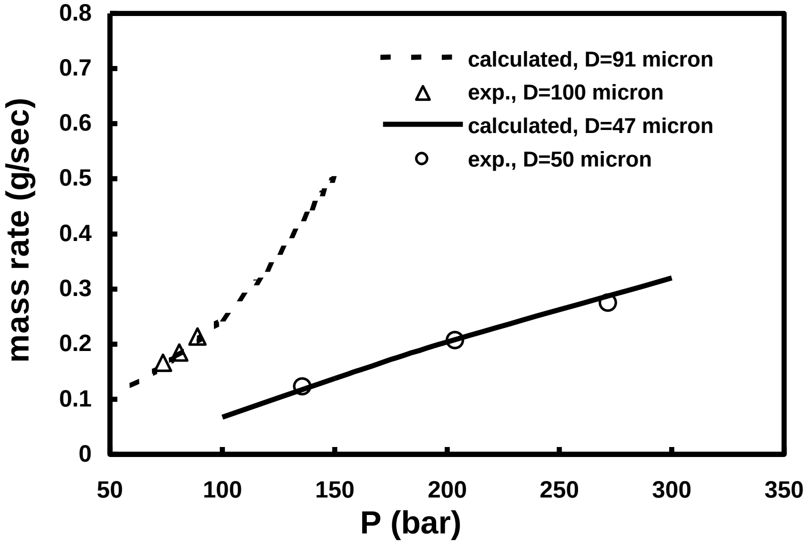


Figure 3.10: Experimental and calculated orifice mass flow rate using the 27-parameter equation of state

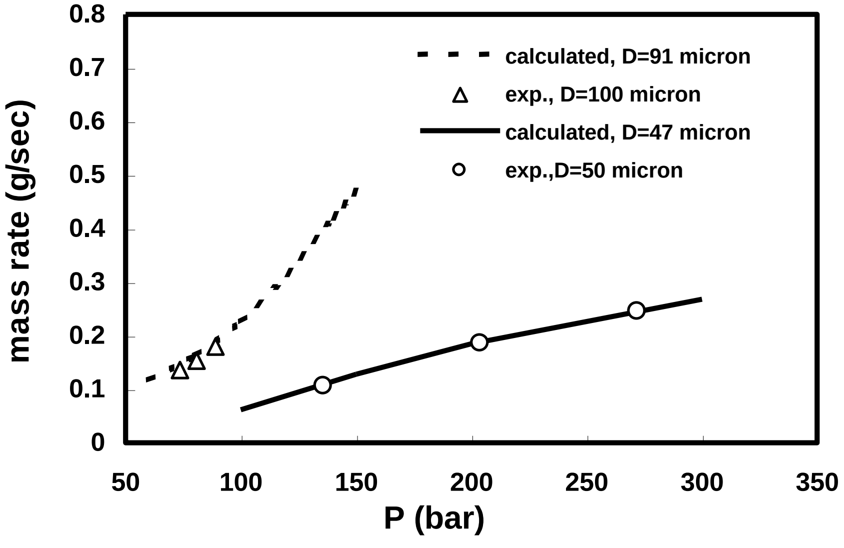


Figure 3.11: Experimental and calculated orifice mass flow rate using Redlich-Kwong equation of state

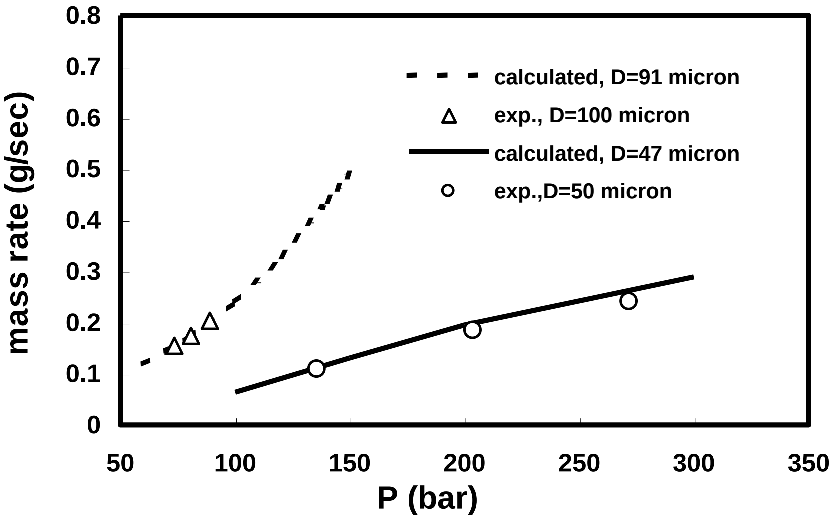


Figure 3.12: Experimental and calculated orifice mass flow rate using Peng-Robinson equation of state

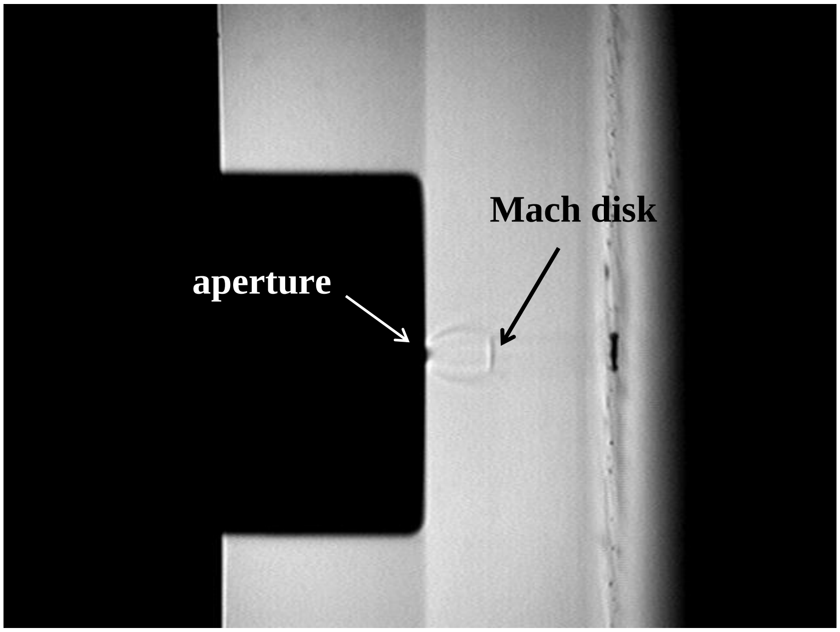


Figure 3.13: Shadowgraph of the free-jet of CO_2 into the atmosphere (P_o= 137 bar, T_o= 70°C)

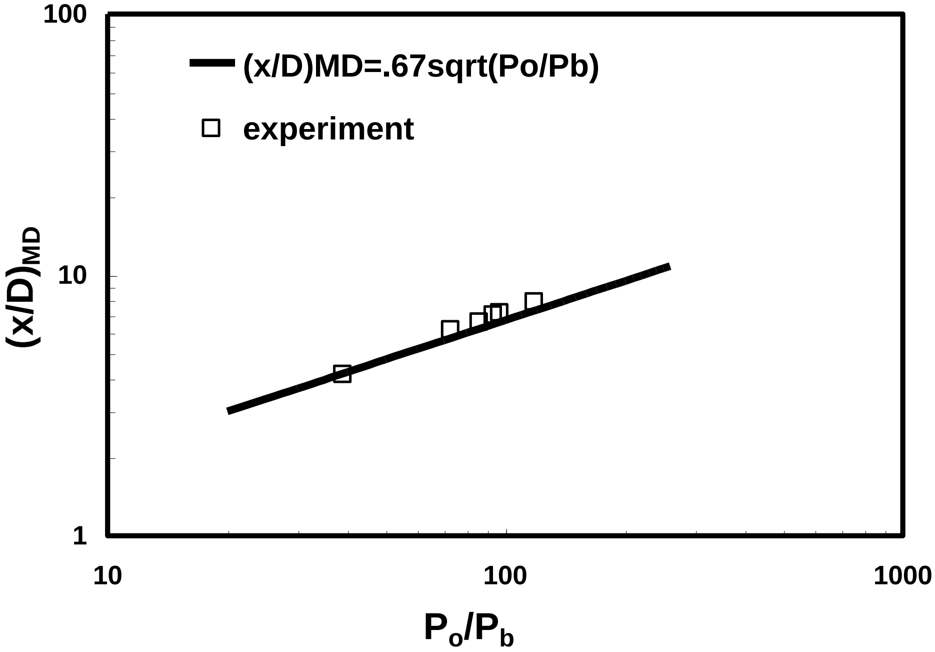


Figure 3.14: Experimental Mach disk location for argon free-jet

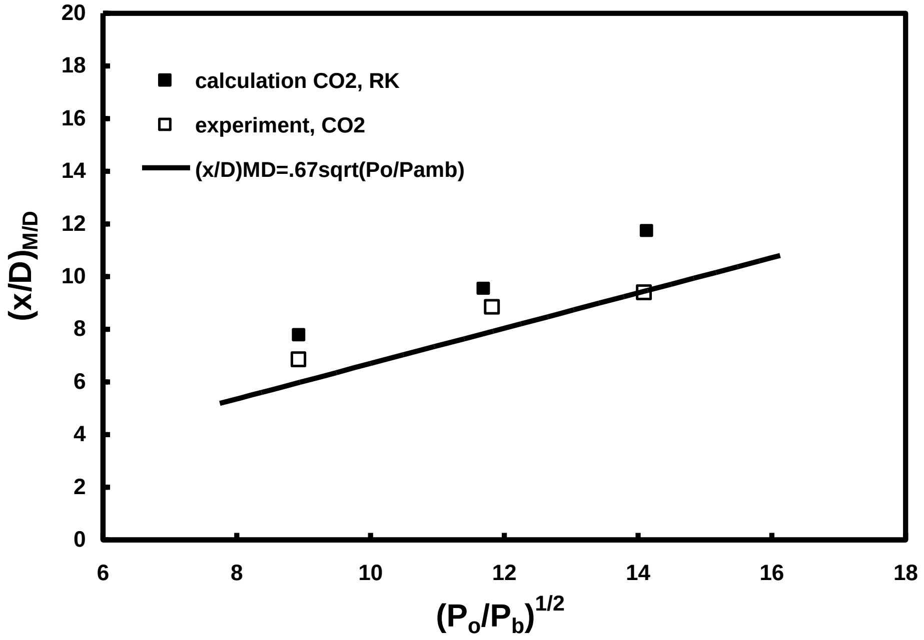


Figure 3.15: Experimental and calculated Mach disk location for CO_2 free-jet

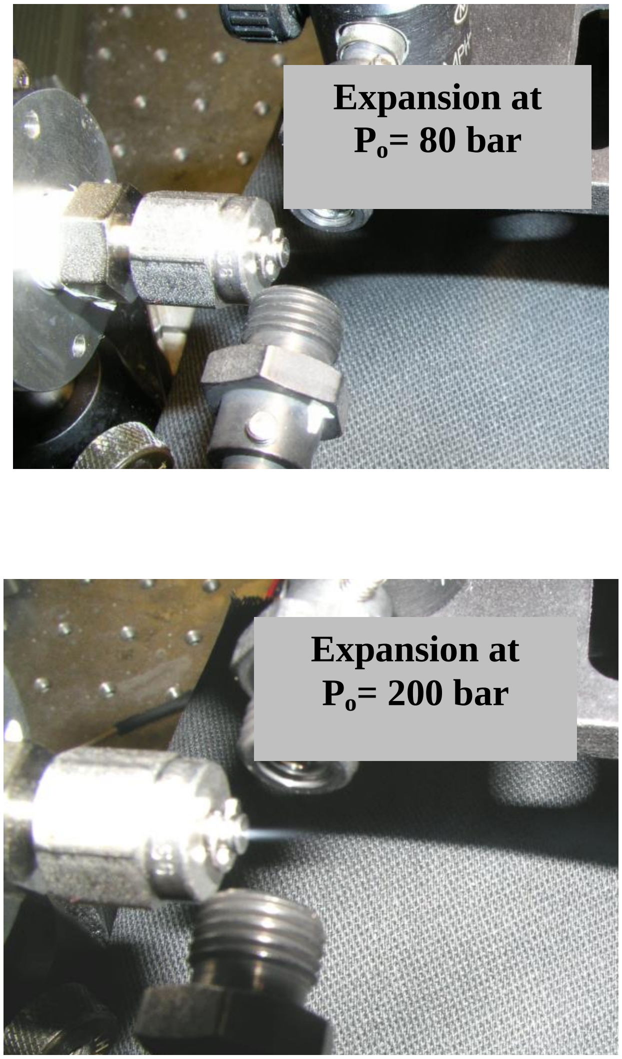


Figure 3.16: Effect of source pressure on the appearance of condensation in the CO_2 free-jet expansion

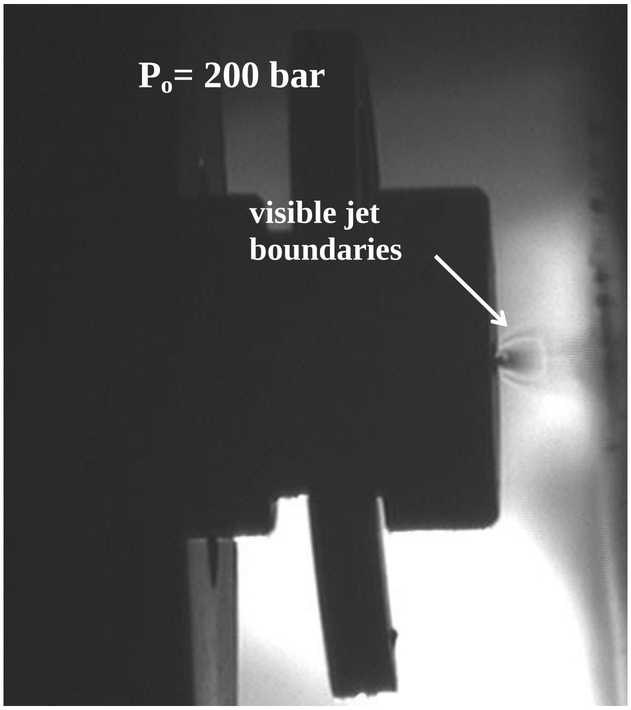


Figure 3.17: Shock structure of the CO_2 free-jet at high pressure

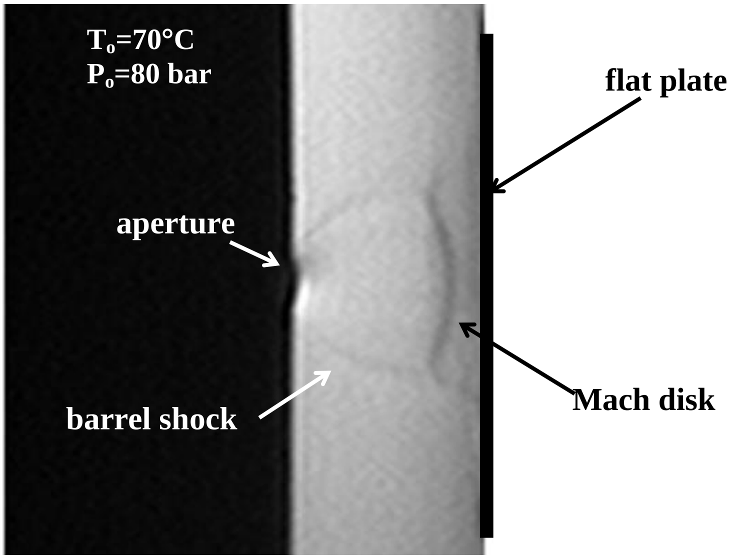


Figure 3.18: Shadowgraph of the CO_2 free-jet impacting a flat plate

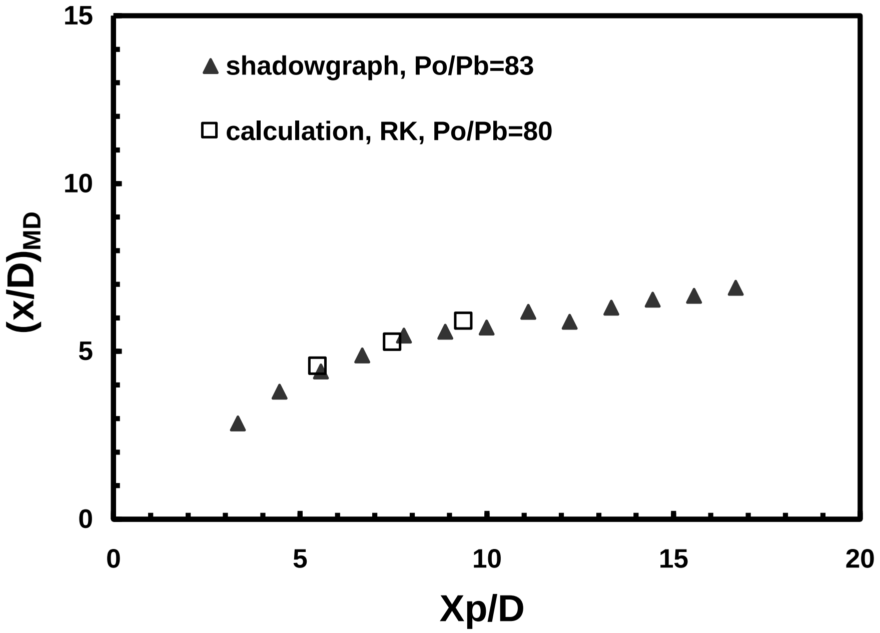


Figure 3.19: Mach disk location as a function of plate distance for CO_2 at P_o= 80 bar, T_o= 70°C

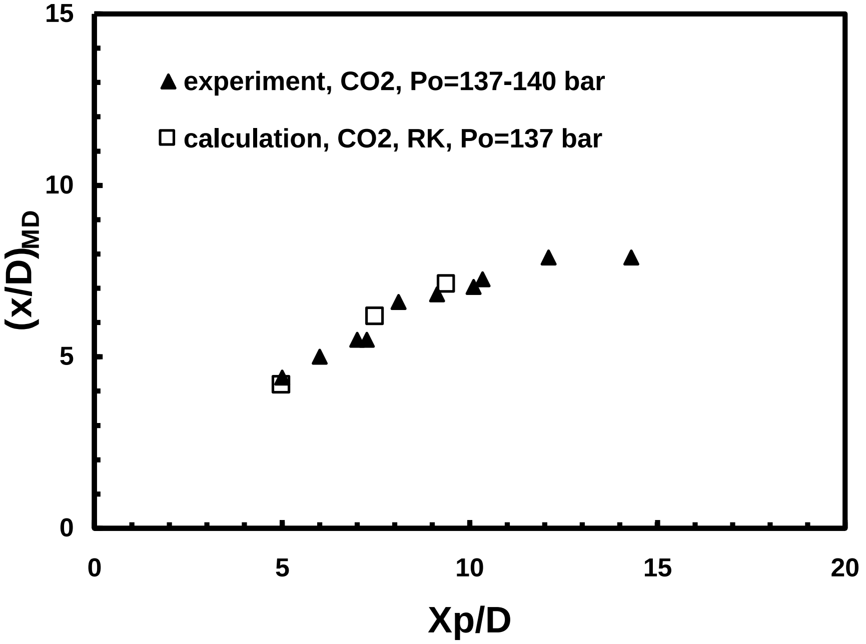


Figure 3.20: Mach disk location as a function of plate distance for CO_2 at P_o= 137 bar, T_o= 70°C

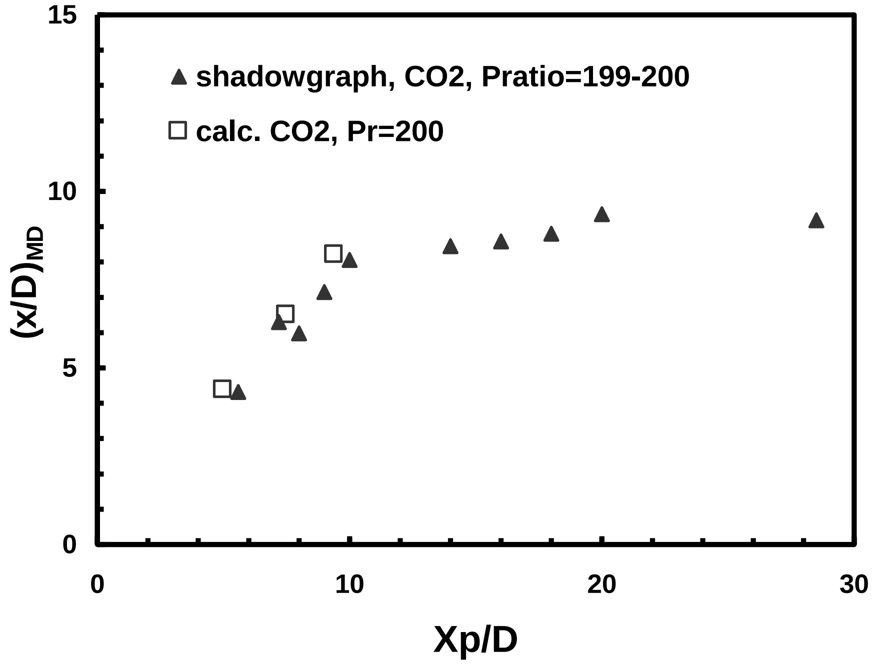


Figure 3.21: Mach disk location as a function of plate distance for CO_2 at P_o= 200 bar, T_o= 70°C

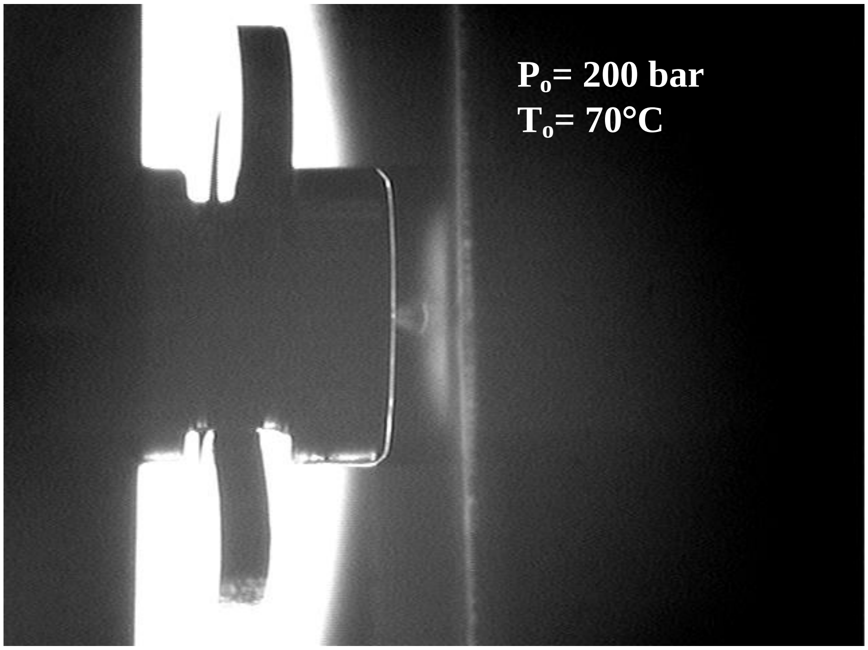


Figure 3.22: Condensation of CO_2 downstream of the Mach disk

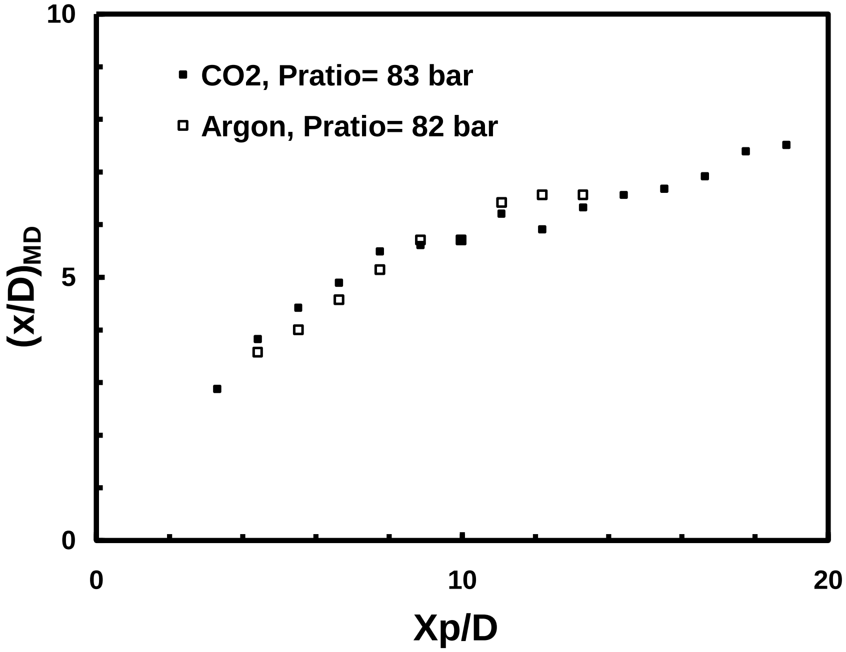


Figure 3.23: Comparison of experimental Mach disk location as a function of plate location for CO_2 and argon

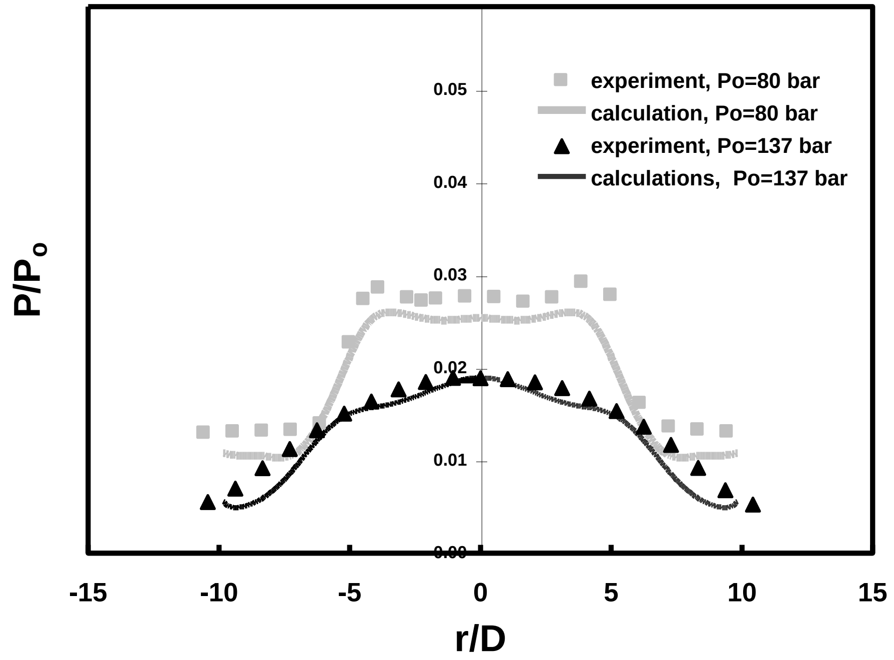


Figure 3.24: Pressure profiles at the plate at P_o= 80 and 137 bar, T_o= 70°C

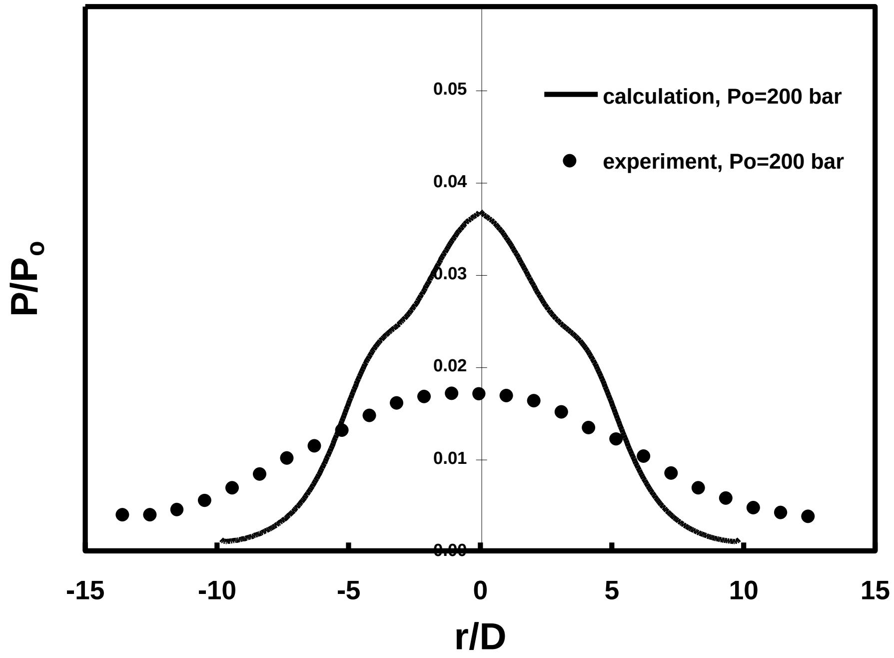


Figure 3.25: Pressure profiles at the plate at P_o= 200 bar, T_o= 70°C

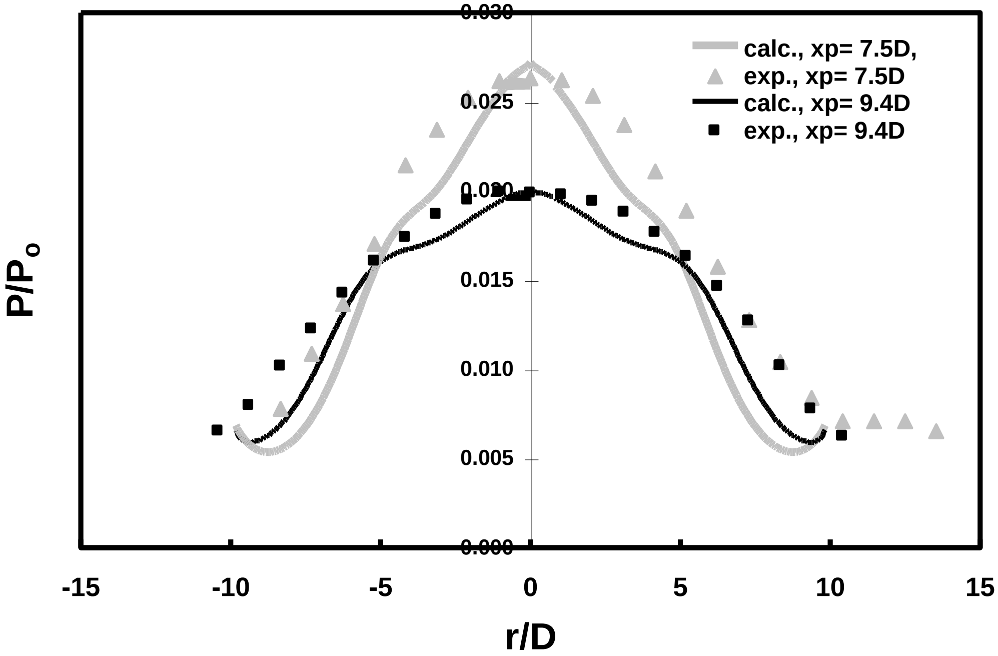


Figure 3.26: Pressure profiles at the plate at P_o= 137 bar for different plate distances, T_o= 70°C

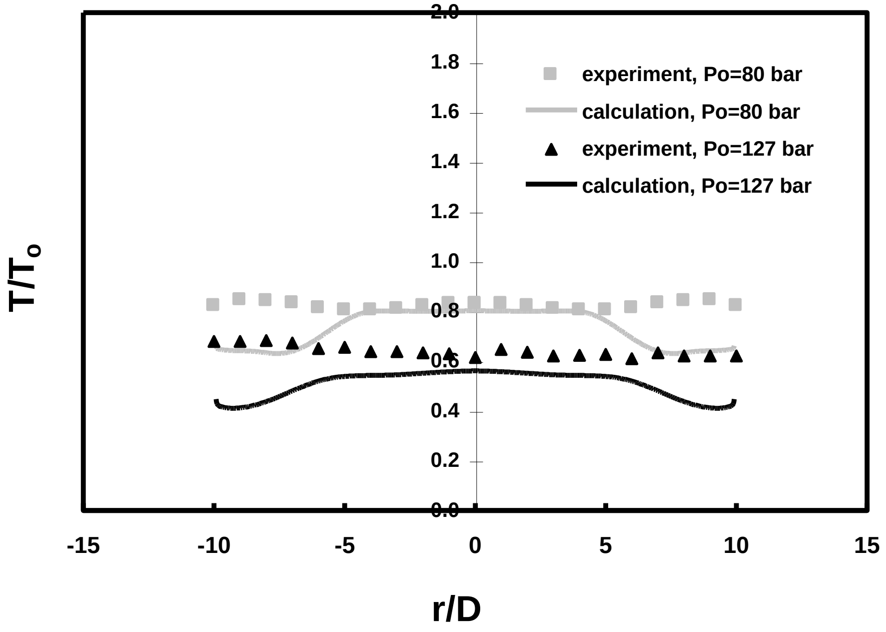


Figure 3.27: Temperature profiles at the plate at P_o= 80 and 137 bar, T_o= 70°C

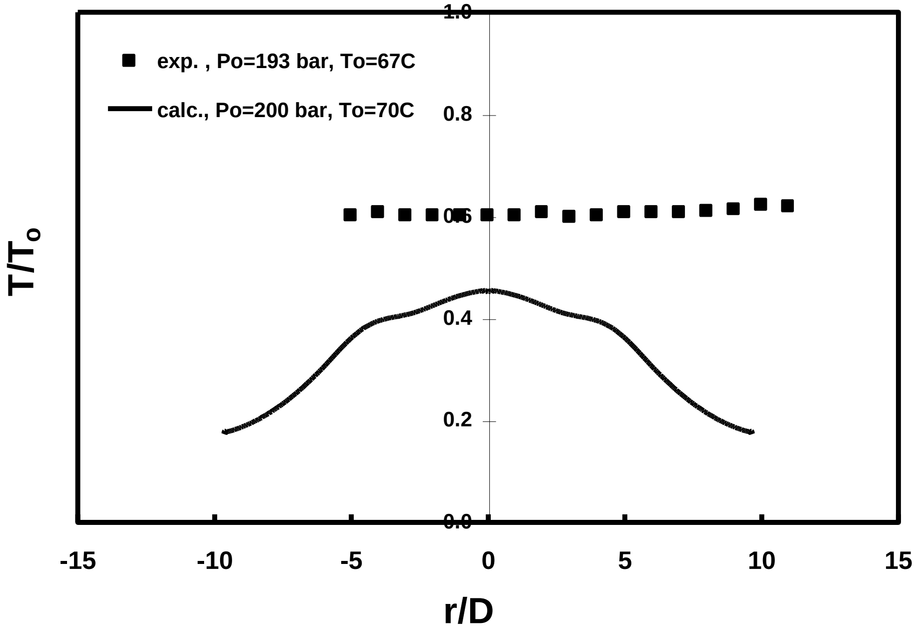


Figure 3.28: Temperature profiles at the plate at P_o= 200 bar

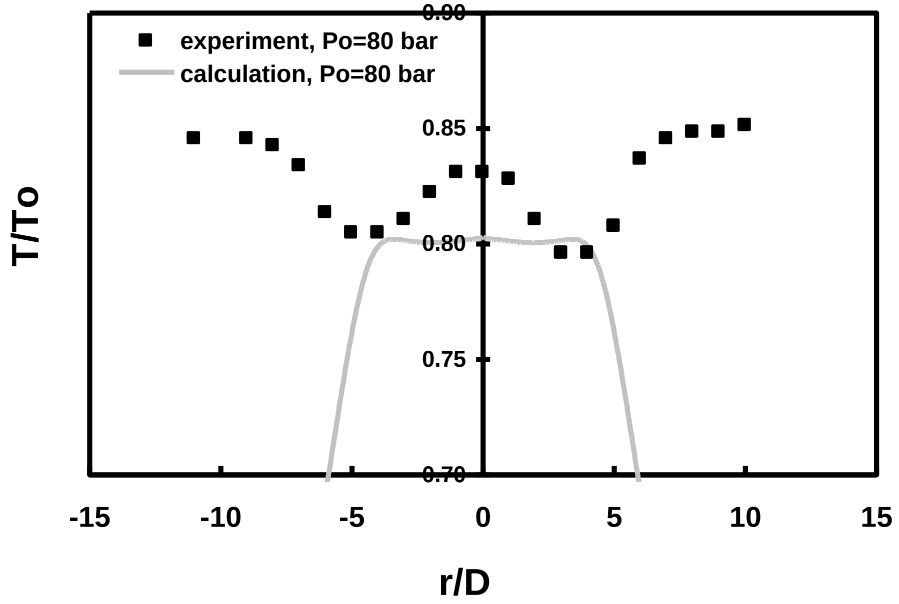


Figure 3.29: Temperature at the plate, P_o= 80 bar, T_o= 70°C

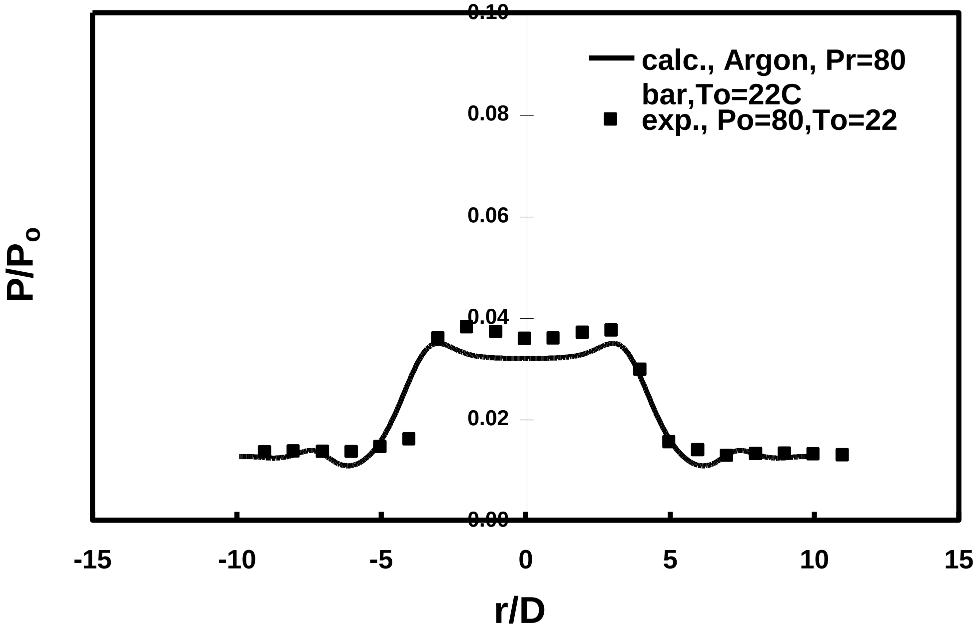


Figure 3.30: Pressure profile at the plate for argon at P_o= 80 bar

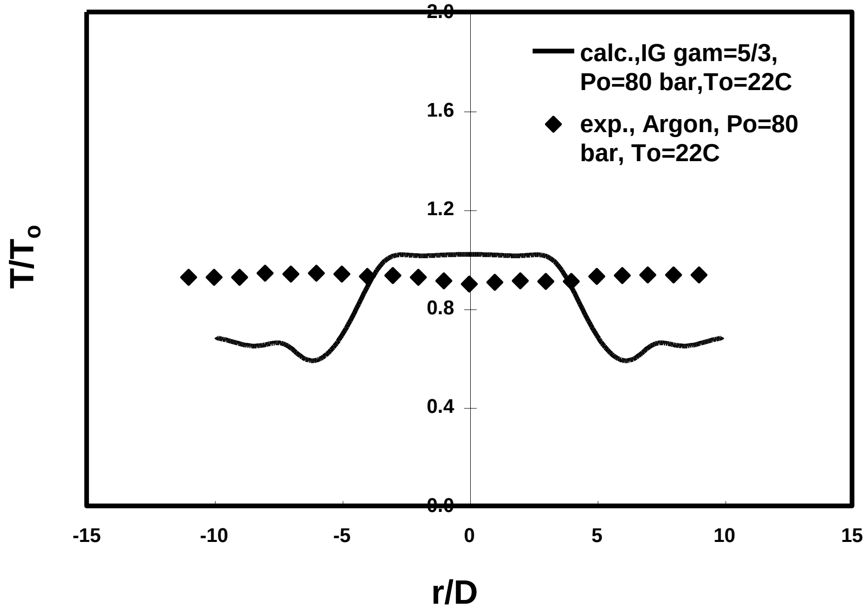


Figure 3.31: Temperature profile at the plate for argon at P_o= 80 bar

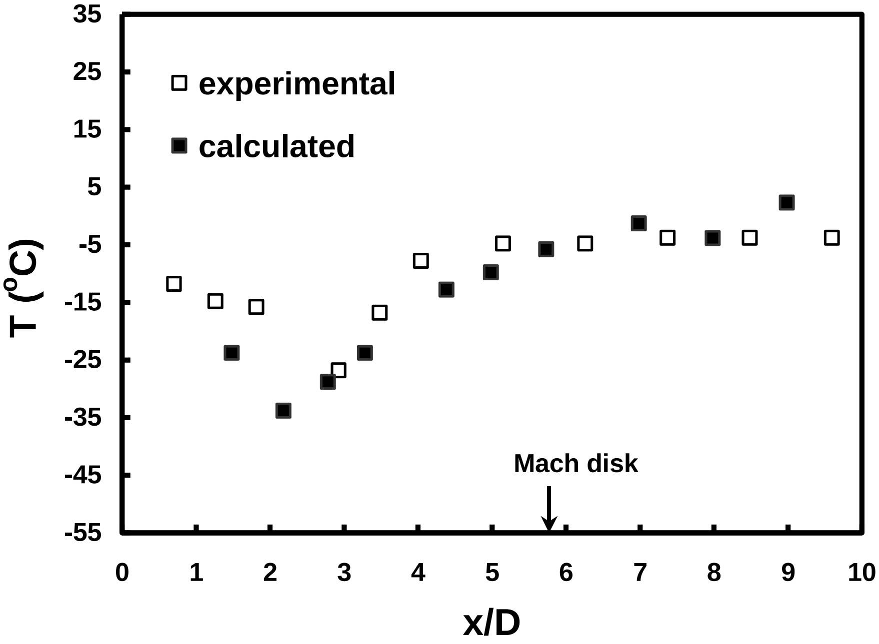


Figure 3.32: Probe temperature of CO_2 free-jet into the atmosphere at P_o= 80 bar, T_o= 70°C

REFERENCES

Liepman, H. W., Puckett, A. E., Introduction to Aerodynamics of a Compressible Fluid, John Wiley, New York (1947)

CHAPTER 4

CAPILLARY

4.1 Introduction

Small tubular capillary sources are often used in the Rapid Expansion of Supercritical Solutions (RESS) experiments because they are easy to fabricate, and in addition to source temperature and pressure, they provide capillary length as one additional parameter to control precipitation of solutes from the supercritical fluid solvent. Although they are not the principal topic of this thesis, we present some preliminary studies of the capillary source because of its wide use by researchers. In order to examine such sources with our free-jet calculation and diagnostics, it is necessary to analyze the flow in the capillary from stagnation conditions to the sonic exit. For our orifice sources above, this was a straightforward isentropic flow calculation, but for the capillary, the typical analysis involves the quasi-one-dimensional (QOD) approximation including viscous frictional effects, as well as heat transfer if the capillary is substantially heated or cooled (Reverchon and Pallado 1996; Maharrey and Miller 2001; Weber and Thies 2002). A strong pressure gradient exists in the constant area tube flow to overcome the viscous wall shear and accelerates the fluid from stagnation conditions to sonic conditions at the exit. Unlike the orifice flow in which the area change dominates the acceleration to sonic conditions, the capillary flow is not isentropic. For the capillaries in our studies, heat transfer is negligible, the flow is adiabatic, and it is referred to as Fanno flow in compressible gas dynamics (Fox and McDonald 1992).

4.2 Theory and Experimental Results

The steady state QOD conservation equations for the capillary flow are:

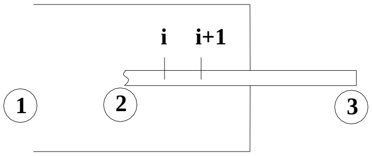
$$\rho u = \dot{m} / A \tag{4.1}$$

$$\frac{dP}{\rho} + udu + 2fu^2 \frac{dx}{D} = 0 \tag{4.2}$$

$$h(\rho, T) + \frac{u^2}{2} = h_o(\rho_o, T_o) \tag{4.3}$$

where friction factors f are available in the literature as a function of Reynolds number, Re. We used standard text relations $f = 0.316\ Re^{-1/4}$ or $f = 64\ Re$, depending on if the flow is turbulent (Re >2300) or laminar (Fox and McDonald 1992). The Reynolds number is calculated as $Re = \rho uD/\mu = 4\dot{m}/(\pi D\mu)$ so that it would be constant down the tube except for the variation of viscosity with temperature and pressure. For our conditions, the range of Re is 76,000 to 245,000 so the flow is turbulent.

In addition, the above implies that, the one-dimensional integration requires the equation of state P(ρ,T) and the viscosity μ(ρ,T) relations. We are able to utilize any of our equations of state above. The following iterative scheme was used to solve the set of equations (4.1) thru (4.3), breaking the calculation into two steps, the isentropic acceleration from stagnation state 1 to the tube inlet at 2, followed by the friction Fanno flow to sonic conditions at the exit at state 3:



isentropic flow to inlet from 1 to 2:

1. guess $\dot{m}$
2. guess ρ_2 $(\rho_2 = \rho_1 - \Delta)$
3. $u_2 = \dot{m}/\rho_2 A_2$
4. $h_2 = h_1 - u_2^2/2$
5. $T_2 = T_2(h_2, \rho_2)$
6. $s_2(\rho_2, T_2) \Rightarrow s_2$, if $s_2 \neq s_1$, go back to #2.

step in x direction from 2 to 3:

7. guess ρ_{i+1} $(\rho_{i+1} = \rho_i - \Delta)$

8. $u_{i+1} = \dfrac{\rho_i u_i}{\rho_{i+1}}$

9. $h_{i+1} = h_o - \dfrac{u_{i+1}{}^2}{2}$

10. $T_{i+1} = T_{i+1}\left(h_{i+1}\ , \rho_{i+1}\right)$

11. $P_{i+1} = P_{i+1}\left(T_{i+1}\ , \rho_{i+1}\right)$

12. calculate x_{i+1} from $\dfrac{P_{i+1} - P_i}{\rho} + u(u_{i+1} - u_i) + 2fu^2\ \dfrac{x_{i+1} - x_i}{D} = 0$

13. $c_{i+1}{}^2 = \left.\dfrac{dP}{d\rho}\right)_{S,i+1} = \dfrac{P_{i+1} - P_{i+1}'}{\rho_{i+1} - \rho_{i+1}'}$;

14. If $x_{i+1} \rangle\ L \pm 0.0000\,L$, then $\mathrm{Mach}_3 = \dfrac{u_3}{c_3}$

15. if $\mathrm{Mach}_3 \neq 1$, go back to #1 to adjust $\dot{m}$

Since the capillaries protrude into the stagnation chamber it is necessary to compute the capillary inlet properties assuming an isentropic acceleration from the stagnation reservoir to the capillary inlet (steps 2-6). The complete iteration involves assuming a mass flow rate, followed by an isentropic calculation to the capillary inlet, and then integrating the Fanno flow equations above to the exit (steps 7-15). At the exit the speed of sound is evaluated to determine if the flow is sonic (step 14), a necessary exit boundary condition for a Fanno flow. The mass flow rate is then iterated until the proper sonic exit condition is satisfied. The secant method was used in steps 5 and 10 to calculate h, given ρ and T. To calculate the speed of sound we numerically evaluated the derivative $(dP/d\rho)_s$, at any ρ, and T by selecting a small Δρ about the local ρ and calculating the associated ΔP for an isentropic process using both P(ρ,T) and s(ρ,T) equations of state. The FORTRAN code for this calculation is given in the Appendix C.

This calculation not only provides the flow rate but also the inlet conditions, P^*, T^* for the subsequent supersonic free-jet expansion. The free-jet is subsequently solved exactly as was done for the orifices above. The viscosity of CO_2, μ, is needed to calculate the friction factor and we used data from (Michels *et al.* 1957; Vukalovich and Altunin 1968; Herreman *et al.* 1971; Poling *et al.* 2000). The data for viscosity was programmed as a look-up table whose values are listed as a function of ρ and T. A subroutine that interpolates the data for viscosity in the table was used to calculate μ given ρ and T. Figures 4.1 and 4.2 show the resulting pressure and temperature profiles along the capillary with inside diameter D = 100 μm and a length versus diameter ratio L/D = 100, when T_o= 70°C and P_o= 100 bar upstream in the stagnation region using the Relich-Kwong equation of state. The very rapid decrease in temperature and pressure in the last ten diameters has serious consequences for precipitation of solutes in RESS experiments, and the possible onset of condensation for the solvent. However, in our experiments in this thesis, we do not include solutes and the conditions were such that the CO_2 did not become super-saturated until exiting the capillary and expanding into the free-jet.

The experimental apparatus is identical to that described in chapter 3 above with the exception of the capillary source. In order to keep the comparison with the orifice source as nearly identical as possible, we used quartz capillary tubing (Scientific Instrument Supply, P/N16026351), with a 0.036 cm outside diameter and a 0.01 cm inside diameter. The diameters were verified directly using a microscope. The capillary length was cut so that the length-to-diameter ratio of the capillary is 100. A 0.04 cm diameter hole was drilled into the same swagelok high pressure fitting used for the orifice sources and the capillaries were then inserted and cemented into the swagelok fitting, with about 0.1 cm projecting externally and the rest protruding back into the stagnation source. Figure 4.3 is a photo of the capillary source. Since a portion of the capillary protruded out from the metal nut, there was concern that the flow might not be adiabatic due to heat transfer to the ambient air. We therefore made several runs with the capillary submerged in a heated bath, so that the entire source was maintained at stagnation conditions. We found excellent agreement with the source in the air, which is consistent with our heat transfer analysis and previous laboratory studies with ideal gases (Murhpy and Miller 1984; Miller *et al.* 1982). We therefore feel that the adiabatic assumption is very reasonable for our experiments. An example of the results for flow rate is shown in figure 4.4. This calculation used the Redlich-Kwong equation of state for CO_2, and is compared with data taken for a capillary made of quartz tubing, 100 μm internal diameter and one cm long (L/D =100). The agreement is very satisfactory, and perhaps somewhat fortuitous given the assumptions of QOD tube flows. We also show the experimental results for the 100 μm orifice, which demonstrates the effect of the loss of stagnation pressure due to friction on mass flow rate. It would of course be much better to extend our two-dimensional time marching calculations into the subsonic flow in order to treat the viscous flow properly, and more importantly, to provide a better inlet condition for the free-jet. The time marching calculation has this capability (Ben Moussa *et al.* 2003) and such an extension would be useful in the future, especially for application to RESS solute experiments. Previous probes of such Fanno flow in tubes for ideal gases have shown that the velocity profile is not one-dimensional at the exit. In connection with our molecular beam studies, we have previously made extensive comparisons between capillary and orifice sources for ideal gases (Murphy and Miller 1984; Miller *et al.* 1982) and found that the two sources provide nearly identical free-jet properties for the same source diameters provided the comparison is made at the same mass flow rate. The reason for this result is that for the adiabatic acceleration of an ideal gas to sonic conditions, the same flow rate yields the same pressure P^* and temperature T^* at the exit, where Mach number = 1. This is due to the fact that for an ideal gas, both the enthalpy and the speed of sound depend only on temperature, and the temperature at the sonic point depends only on the source or stagnation temperature. However, for the supercritical fluid, the enthalpy and the speed of sound depend on both temperature and pressure and the two parameters are strongly coupled in the accelerating flow. As a demonstration of these features, we plot the experimental mass flow rate of CO_2 versus the pressure at the throat P^* for both the capillary and orifice, at the same stagnation temperature T_o, in figure 4.5. The pressure at the throat is calculated using the experimental stagnation pressure. Therefore it is not valid to scale the free-jet for the capillary source to the results for the isentropic orifice for supercritical fluids by assuming that the same mass flow rate will provide the same inlet conditions to the free-jet, as has been found for ideal gases.

Figure 4.6 shows the experimental Mach disk location as a function of plate position for the orifice and the capillary, both 100 μm in diameter, at the same mass flow rate. In both cases, T_o= 70°C while P_o= 80 bar for the orifice and 100 bar for the capillary. The calculated free-jet shock position is shown to be quiet good for either case provided we use the correct initial sonic conditions. The calculated properties at the sonic exit for this case are T^*= 22°C, P^*= 48 bar for the orifice and T^*= -28°C and P^*= 24 bar for the capillary.

Figure 4.7 shows the pressure profile at the plate for the above two cases, orifice and capillary, at the same mass flow rate when the plate is placed at Xp = 9.4D. As expected, the capillary is not properly scaled using the same mass flow rate and the centerline pressure is lower, primarily because of the lower sonic pressure P^*.

Figure 4.8 compares the experimental pressure profile at the plate for the expansion from the capillary at P_o= 100 bar to the calculated data using the calculated throat conditions for a capillary at P_o= 100 bar (P^*= 24 bar). Although the center pressure is satisfactory, clearly the profiles are not as well reproduced as for those found with the isentropic orifices. Although this feature requires further attention, we feel that the principal cause is that the inlet velocity profile for the free-jet is not well specified for the flow exiting the capillary, and specifically, it is not nearly as one-dimensional as for the orifice. The effect of such a streamline curvature at the sonic exit has been previously investigated for ideal gases by the method of characteristics (Murphy and Miller 1984).

In general these preliminary results suggest that by using the Fanno flow calculation to fix the sonic conditions, we expect that our free-jet calculation will provide a reasonable approximation for the RESS expansion.

To explore the properties of the capillary further, some preliminary experimental work with argon was also done, to eliminate the added complication of the non-ideal gas behavior. Figure 4.9 shows the Mach disk location for the free-jet into the atmosphere for argon as a function of stagnation pressure from the capillary and orifice, both 100 μm in diameter. Because the argon is taken from a gas cylinder, we could not directly compare at the same mass flow rate, only at the same stagnation pressure. Again, as expected the pressure drop in the capillary tube causes the shock wave to occur sooner in the expansion.

Figure 4.10 shows the Mach disk location as a function of P_o for the argon free-jet impacting a flat plate when Xp = 9.4D for the 100 μm capillary and compares it to the previous orifice results of chapter 3. Again, in agreement with the results for supercritical CO_2, the capillary pressure drop, and subsequent decrease in P^* at the source exit, causes the Mach disk location for the capillary to be closer to the sources.

Figure 4.11 shows the pressure profile for argon at the plate, Xp/D = 9.4. The qualitative features and comparisons between the orifice and capillary sources are the same as in figure 4.7 for supercritical CO_2. Figure 4.12 shows the related temperature profiles for the argon expansion. Even though we see the same qualitative flat profiles for the plate temperatures as we did for supercritical CO_2, which we feel are due to heat transfer effects, it is satisfying that the temperature for the ideal gas is essentially the same for the two sources. This is expected for the adiabatic expansion of an ideal gas since the stagnation temperature, T_o, is conserved and there is no real gas Joule-Thompson effect cooling the gas during the expansion and shock compression. Although straightforward, we have not extended our calculations to these ideal gas expansions from capillaries.

These preliminary results for capillary sources will be extended to solutes in the near future because these sources are widely used for RESS experiments. However, the results show that the subsequent free-jet can be understood by correctly assessing the Fanno flow down the tube, and that the QOD Fanno calculation may be a useful approximation. The question of the friction induced velocity profile at the tube exit, and its effect on the free-jet expansion, remains to be investigated. Such future work should assume some profile shapes for the initial free-jet conditions. Early studies with ideal gases (Miller *et al.* 1982) will serve as a guide for the assumed profiles.

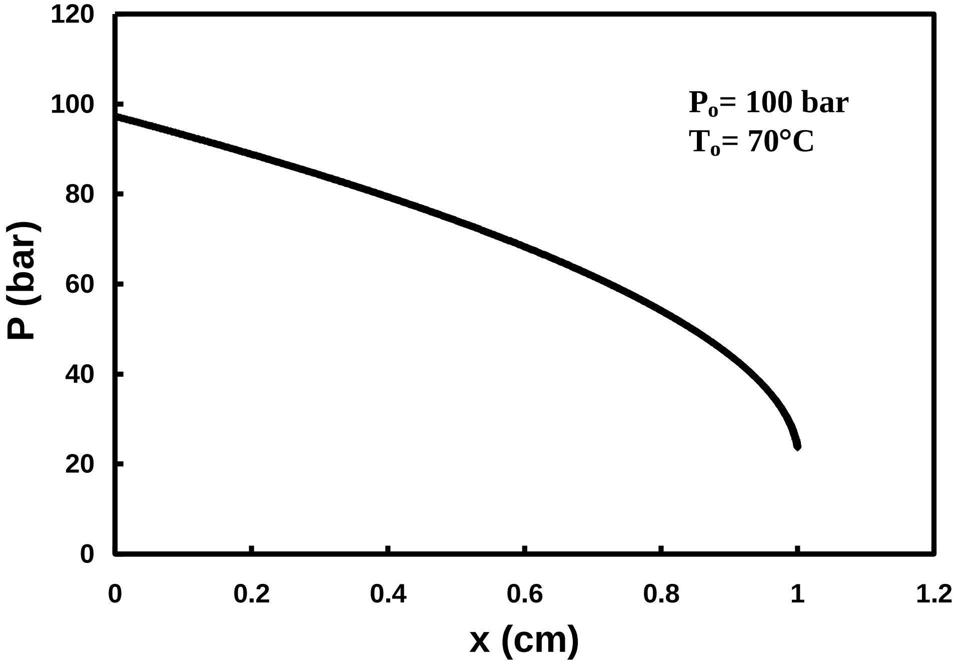


Figure 4.1: Pressure drop along the capillary, D = 100 μm, L/D = 100

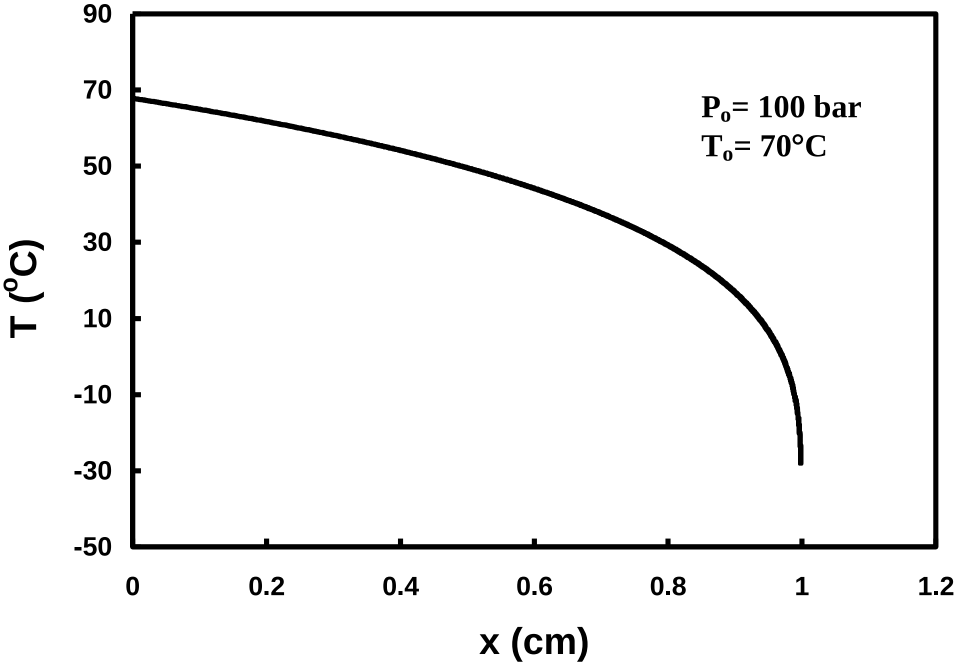


Figure 4.2: Temperature drop along the capillary, D = 100 μm, L/D = 100

Figure 4.3: Capillary source

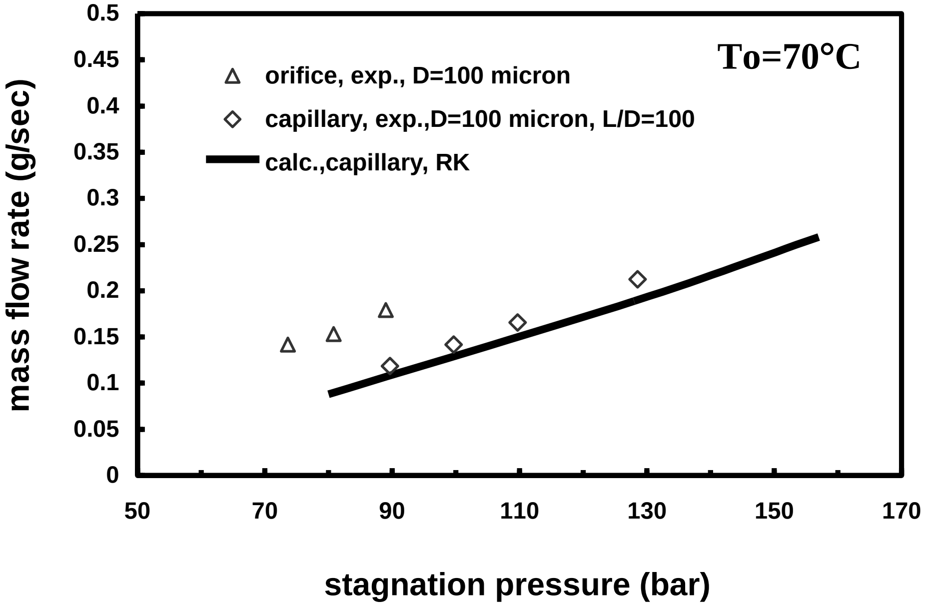


Figure 4.4: Experimental and calculated capillary mass flow rate using Redlich-Kwong equation of state

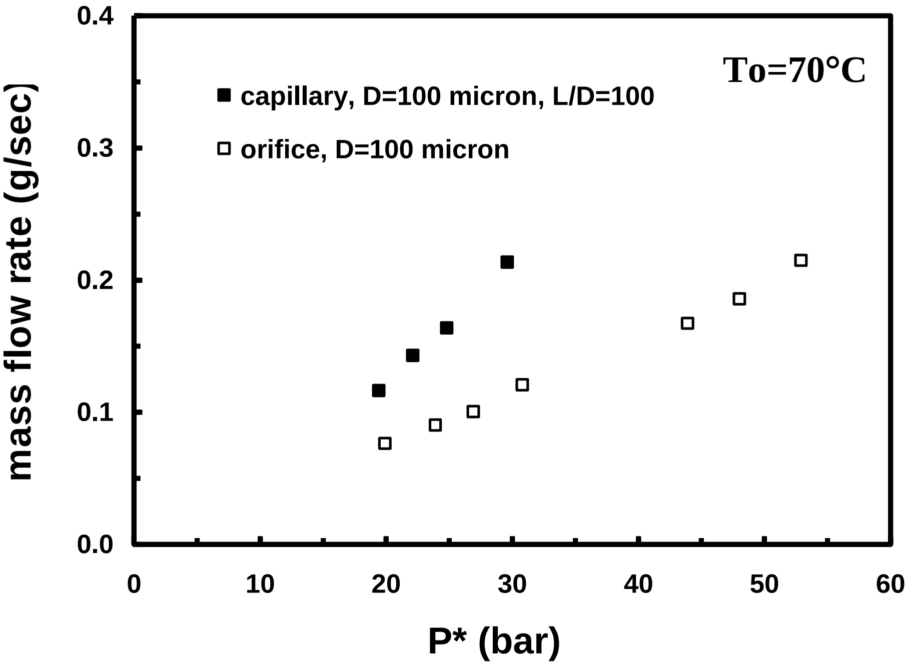


Figure 4.5: CO_2 mass flow rate versus calculated sonic throat pressure for capillary and orifice

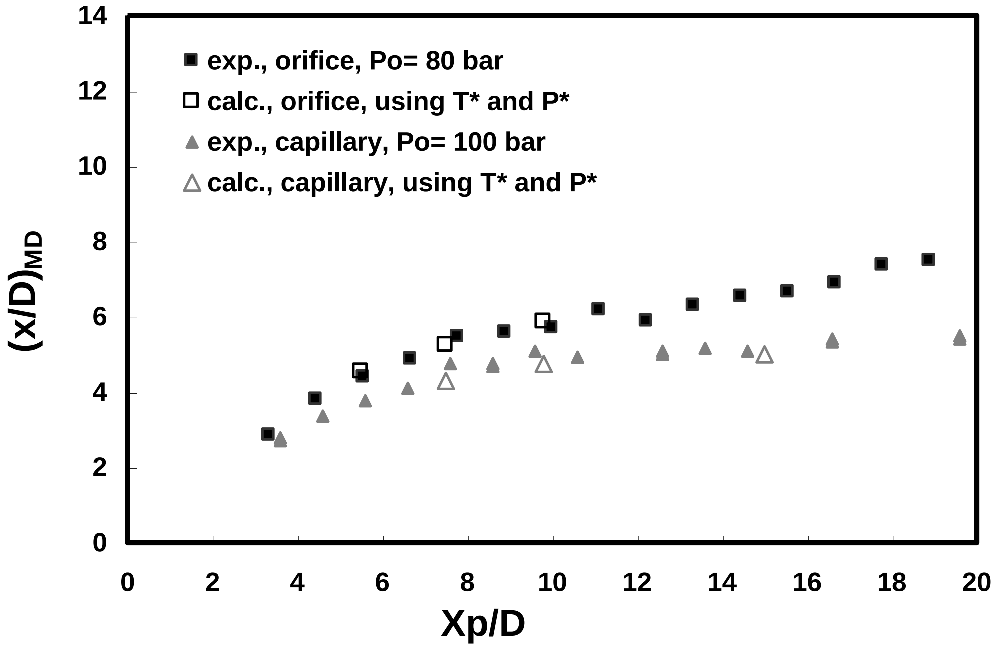


Figure 4.6: Experimental and calculated CO_2 Mach disk location as function of plate distance for orifice and capillary at the same mass flow rate

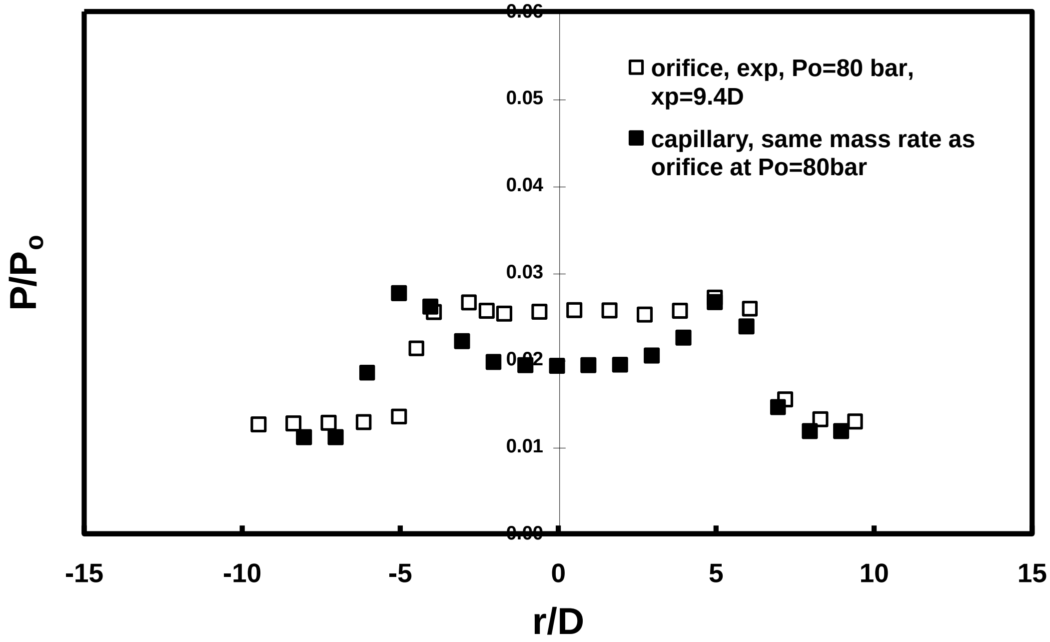


Figure 4.7: Experimental pressure profile at the plate for the orifice and capillary at the same CO_2 mass flow rate; Xp= 9.4D

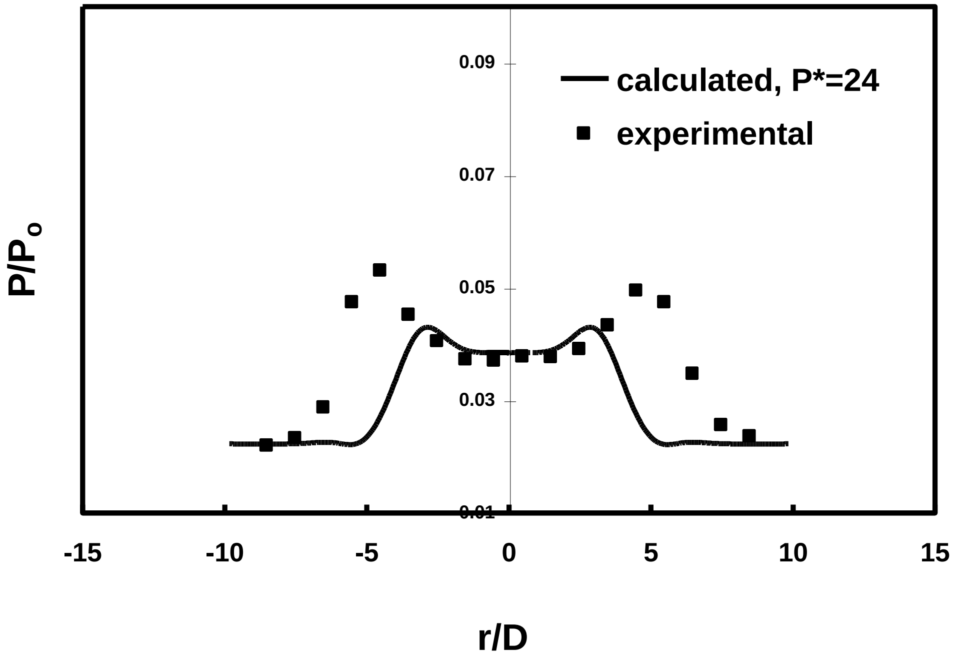


Figure 4.8: Experimental and calculated capillary pressure profile at the plate (data corresponds to figure 4.7)

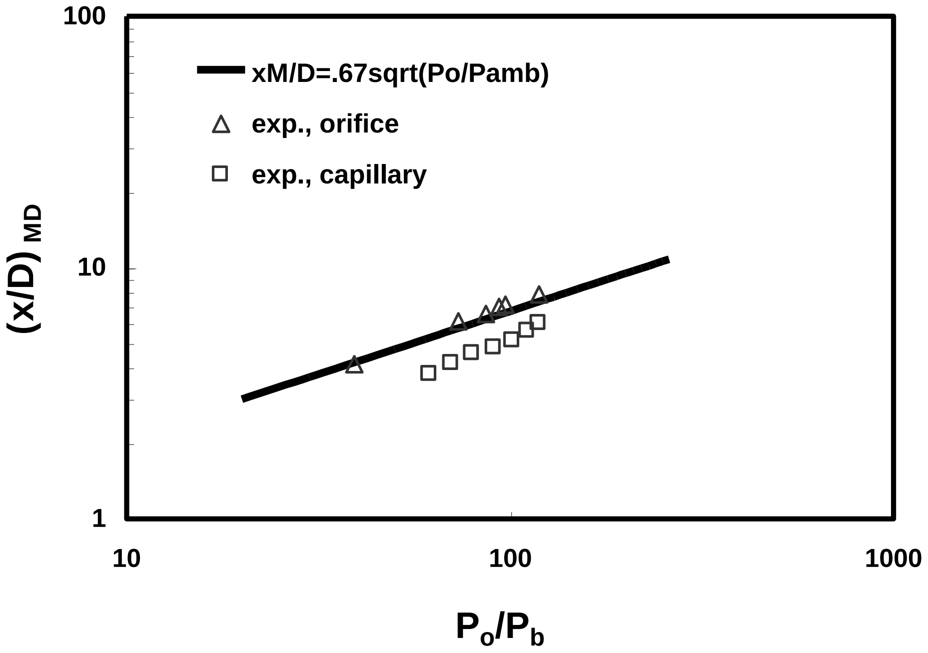


Figure 4.9: Mach disk location as function of stagnation pressure for argon free-jet into atmosphere, T_o= 22°C

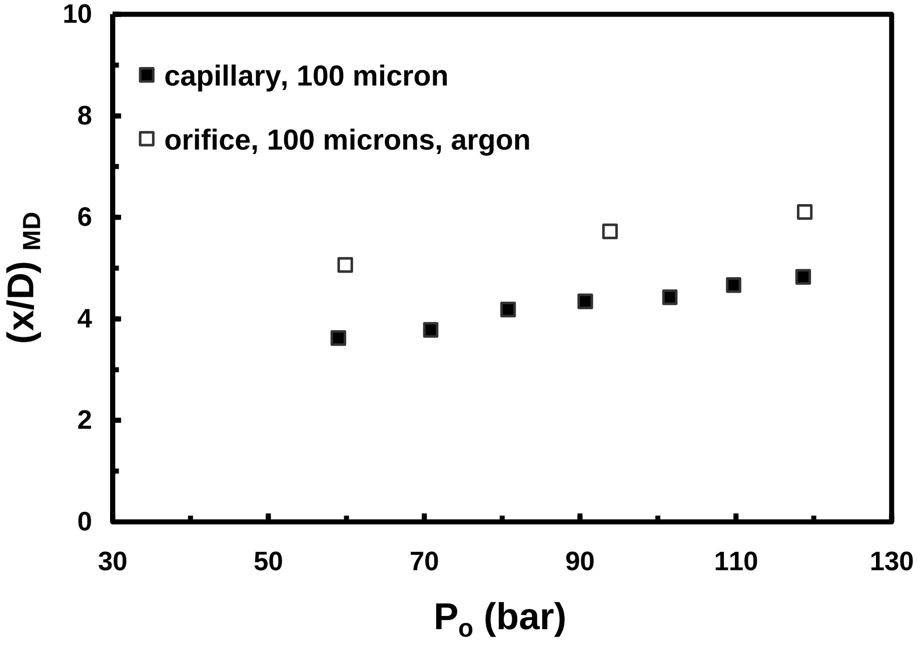


Figure 4.10: Mach disk location as function of stagnation pressure for argon free-jet impacting a plate at Xp= 9.4 D

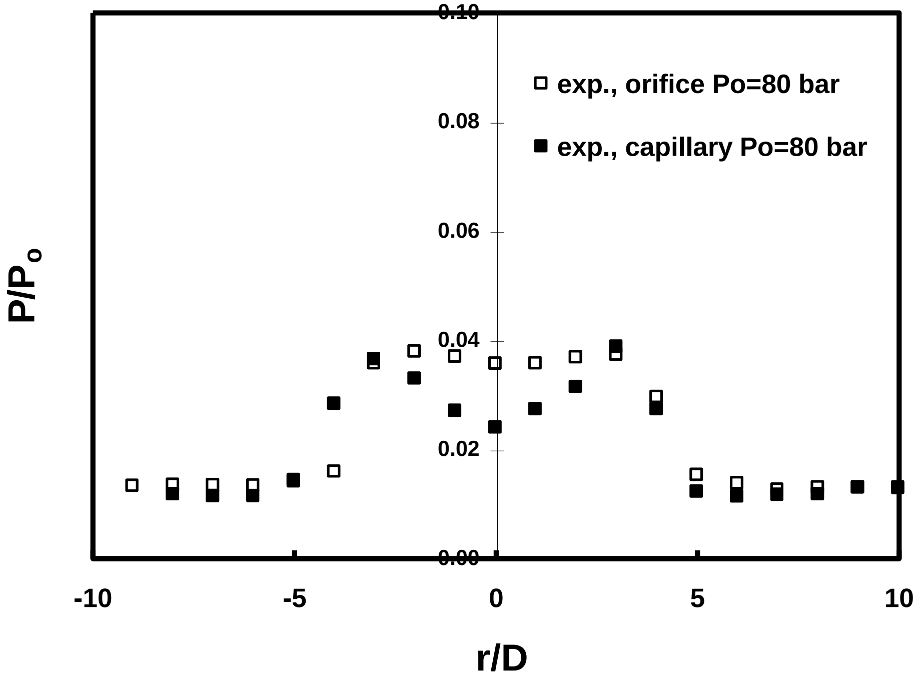


Figure 4.11: Argon experimental pressure profile at the plate for the orifice and capillary at P_o= 80 bar, T_o= 22°C; Xp= 9.4D

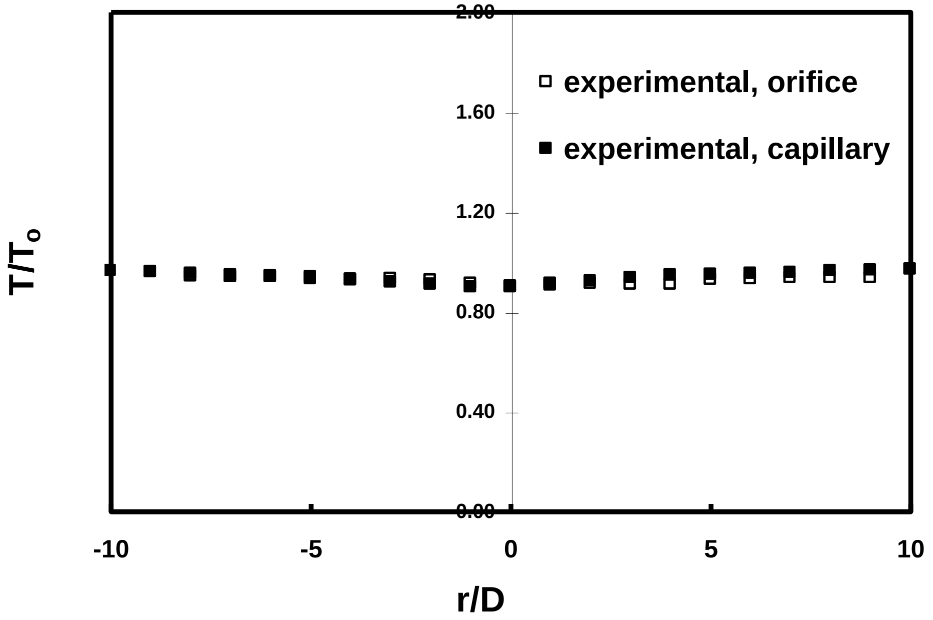


Figure 4.12: Argon experimental temperature profile at the plate for the orifice and capillary at P_o= 80 bar, T_o= 22°C; Xp= 9.4D

REFERENCES

Ben Moussa, A., Ksibi, H., Baccar, M., Capillary Nozzles in the Ress Process: Hydrodynamic Modeling, Proc. 6th International Symposium on Supercritical Fluids, Versailles, France, Vol. 3, pp 1719 (2003)

Fox, R., and McDonald, A., Introduction to Fluid Mechanics, John Wiley and sons (1992)

Herreman, W., Lattenist, A., Grevendonk, W., and De Bock, A., Measurements of the Viscosity of Carbon Dioxide, Physica **52**, pp 489 (1971)

Maharrey, S., and Miller, D. R., A quartz Capillary Micro-Reactor for Studies of Oxidation in Supercritical Water, AIChE J. **47**, pp 1203 (2001)

Michels, A., Botzen, A., and Schuurman, W., The Viscosity of Carbon Dioxide Between 0°C and 75°C and at Pressures up to 2000 Atmospheres, Physica **23**, pp 95 (1957)

Miller, D., Fineman, M., and Murphy, H., The Free-Jet Expansion from a Capillary Source, 13th Int. Symp. On Rarefied Gas Dynamics, Novosibirsk, Russia (1982)

Murphy, H., and Miller, D. R., Off Axis Properties of Free-Jets from Different Subsonic Flows, in 14th Int. Symp. Rarefied Gas Dynamics, edited by H. Oguchi, pp 743, Univ. of Tokyo Press, Tokyo (1984)

Murphy, H., and Miller, D., Effects of Nozzle Geometry on Kinetics in Free-Jet Expansions, J. Physical Chemistry **88**, pp 4474 (1984)

Poling, B. E., Prausnitz, J. M., O’Connell, J. P., The Properties of Gases and Liquids, McGraw Hill (2000)

Reverchon, E., and Pallado, P., Hydrodynamic Modeling of the RESS Process, J. Supercritical Fluids **9**, pp 216 (1996)

Vukalovich, M. P., and Altunin, V. V., Thermophysical Properties of Carbon Dioxide, London and Wellingborough (1968)

Weber, M., and Thies, M., Undestanding the RESS Process, in Supercritical Fluid Technology in Materials Science and Engineering, edited by Y. Sun, pp387, New York (2002)

CHAPTER 5

CONDENSATION

5.1 Introduction

The analysis in the chapters above has assumed a pure homogeneous fluid, CO_2, neglecting any clustering or condensation in the free-jet expansion. As we have shown, the expanding fluid does in fact enter the two phase regime (figure 2.9) where, if thermodynamic equilibrium were achieved, significant condensation would occur. However, the small geometric scale of the experiments and rapid rate of the supersonic expansion means that the time scale for kinetics is of the order of microseconds and equilibrium is not achieved for many processes. For example, in chapter 2, we showed that even the CO_2 vibrational degrees of freedom cannot keep up with the rapid expansion despite the high supercritical fluid source pressures. We therefore do not expect equilibrium to be achieved and the flow is approximated as a supersaturated, metastable fluid. Certainly small clusters of CO_2 molecules are forming and dissociating in the expansion, and at the highest pressures we clearly observe condensation in the jet, shown in figure 3.22 above. For application to Rapid Expansion of Supercritical Solutions (RESS) experiments, which include solutes in the supercritical fluid solvent, there is the added complication of the condensation/precipitation of the solute species, and such solute clusters could additionally serve as heterogeneous nucleation sites for the condensation of the supersaturated CO_2. RESS investigators are primarily interested in the solute and they have decoupled the precipitation of the solute from the solvent, which is reasonable since the solute is naturally supersaturated in the expansion well before the solvent, usually upstream or near the throat area of the expansion where the fluid densities are high. For small concentrations of solutes such calculations are actually more straightforward because the condensation kinetics can be decoupled from the fluid dynamics. However, this thesis is concerned only with pure solvent CO_2 expansions, and in this chapter we present some of our preliminary analysis based on classical nucleation theory. The analysis is certainly incomplete and requires additional experimental data to be able to determine the best approach. To that end, we also describe an experimental time-of-flight mass spectrometer facility being developed to probe these expansions and provide data on the clustering. Debendetti and colleagues have shown how to extend the classical condensation theory to solutes (Kwauk and Debenedetti 1993; Weber and Thies 2002) and if we were to extend our analysis to solutes we would follow their approach. As we will show, the treatment of clustering of pure CO_2 in these high pressure expansions is difficult enough.

There is a vast literature on condensation in expansions such as our free-jets, varying from the classical thermodynamic models to molecular collision kinetic models. We will not review the details of the classical approach, but simply state and use well known equations. Our treatment follows closely that described and used by several investigators who have studied condensation in similar gas dynamic expansions, and the reader is referred to summaries by Stever (1958), Hill (1963), Wegener (1969), Kotake (1978), Abraham *et al.* (1981), and Bayazitoglu (1996). We also make limited use of the molecular kinetic approach utilized by Knuth (1977) to address issues for small clusters. It is clear that the small clusters formed in our free-jets fall into a regime where neither the classical nor the kinetic results are accurate and, as we show, there are too many parameters to adjust to render any such analysis quantitative. Nevertheless, to understand the RESS expansion processes, such an analysis can be used to guide an understanding of the important parameters, and the results can be scaled empirically once clustering data is obtained. Again, this chapter is meant to serve as an introduction for future work on this very important aspect of understanding the expansion of supercritical fluids and their use for molecular beam experiments and film growth.

5.2 Theoretical Results

To investigate the clustering and condensation of CO_2 in the rapid free-jet expansion, we utilize the quasi-one-dimensional (QOD) approach described in chapter 2. The approach can be extended to the two-dimensional axisymmetric free-jet (ASFJ) analysis but the uncertainties in the thermodynamic and kinetic parameters do not warrant that effort at this point in our studies. The quasi-one-dimensional area expansion A(x) used in this chapter has been determined by the methods described above and should be a reasonable approximation to the rate of expansion. As noted above, the QOD approach permits us to directly integrate the fluid equations so that we do not need to use the time marching numerical method. As we show below, the analysis predicts substantial condensation and it is therefore necessary to couple the kinetics and thermodynamics to the fluid dynamic equations of the flow in order to properly account for the loss of vapor phase molecules and the heat release in the condensation.

The QOD steady state equations with condensation are:

$$\rho A v = \dot{m}(1-\mu) \tag{5.1}$$

$$\rho A u du = -A dP \tag{5.2}$$

$$h + u^2/2 = h_o + \mu h_{fg} \tag{5.3}$$

$$h = h(\rho, T) \tag{5.4}$$

$$P = P(\rho, T) \tag{5.5}$$

As before, we will also utilize the entropy relation in place of the momentum equation whenever possible, which with condensation takes the form:

$$T ds = \delta Q = h_{fg}\, d\mu \tag{5.6}$$

In these equations, ρ is the density of vapor, P is the vapor phase pressure and $P(\rho,T)$ is the non-ideal equation of state (e.g. Redlich-Kwong), μ is the liquid phase mass fraction due to both nucleation and growth, μ = mass liquid / mass (gas+liquid), and h_{fg} is the latent heat per unit mass released during condensation. In addition to the new terms in the conservation of mass and total energy equations, it is necessary to properly treat the equations of state for h and s to account for the two phases, and to introduce the condensation equations which determine the mass fraction μ.

Following classical nucleation theory, at each differential volume along the QOD nozzle flow, a meta-stable cluster of critical radius size r^* is formed at rate J (number per m^3 per second). Clusters of size $r > r^*$ may grow rapidly, while clusters $r < r^*$ evaporate rapidly. These critical clusters are then able to grow by monomer attachment through gas phase collisions at a rate dr/dt = u dr/dx along the nozzle; we neglect agglomeration of large clusters. The resulting mass fraction, μ , is then a sum over all of the clusters which have been generated and grown upstream of any particular nozzle position. The appropriate relations for these terms are (Wegener 1969; Kotake 1978):

$$r^* = \frac{2\sigma\mu_v}{\rho_L R_u T \ln\left(P_v / P_{vs}{}^{\infty}\right)} \tag{5.7}$$

$$J = \alpha_v \left(\frac{P_v}{kT}\right)^2 \frac{1}{\rho_L} \left(\frac{2\mu_v \sigma}{\pi N_A}\right)^{1/2} \exp\left(\frac{-4\pi\sigma r^{*2}}{3kT}\right) \tag{5.8}$$

$$\frac{dr}{dx} = \frac{\alpha_v}{\rho_L u} \left(\frac{\mu_v}{2\pi R_u T}\right)^{1/2} \left(P_v - P_{vs}\right) \tag{5.9}$$

$$\mu_{i+1} = \mu_i + \sum_{k=1}^{i-1} \frac{4\pi\rho_L}{3\dot{m}} 3r(k)^2 dr(k)J(k)A(k)dx + \frac{4\pi\rho_L}{3\dot{m}} r(i)^3 J(i)A(i)dx \quad (5.10)$$

The cluster growth relation includes gain and loss terms to account for addition by attachment and evaporation. The relation for the mass fraction at position (i) includes a term for the generation of new clusters J(i) and a sum over all previous clusters generated at position k at rate J(k), and which have grown, dr(k), from position (k) to (i). The above relations account for the effects of small radius on vapor pressure.

In these relations σ is the surface tension, μ_v is the molecular weight of the vapor CO_2, ρ_L is the density of the liquid CO_2, P_v is the pressure of the vapor (written as P in fluid flow equations of motion (5.2) above), P_{vs}^{∞} is the saturation pressure of the vapor at temperature T, P_{vs} is the saturation pressure including Kelvin effects, α_v is a sticking coefficient correction for J, K is Boltzman constant, R_u is the universal gas constant, N_A is Avogadro's number, and u is the centerline velocity in the expansion.

We have used the following values for these variables in our calculations for CO_2 (Zharkova 1978):

$$\sigma = (34.96 - 0.092\ T) \times 10^{-3}\ N/m \quad (5.11)$$

$$\rho_L = (-1.0333 \times 10^{-3}\ T + 1.4387) \times 10^3\ Kg/m^3 \quad (5.12)$$

$$P_{vs}^{\infty} = \left[\exp\left(11.445 - \frac{2126.9}{T}\right)\right] x10^5\ N/m^2;\ \text{if}\ T \geq 216.4K \quad (5.13)$$

$$P_{vs}^{\infty} = 10^{9.81066 - 1347.786/T} \text{ x } \frac{10^5}{760} \text{ N/m}^2 \text{; if } T \langle 216.4\text{K} \qquad (5.14)$$

$$P_{vs} = P_{vs}^{\infty} \exp(2\sigma/\rho_L RTr) \qquad (5.15)$$

We also take for h_{fg} = 40,200 Joules/ Kg for CO_2 (Zharkova 1978) and μ = 44.02 Kg/mol.

In the above relations, additional corrections, with additional parameters, to the surface tension and heat of condensation can be introduced to account for effects of small cluster radius (Hirschfelder 1954; Wegener 1969), and a sticking coefficient correction α_v can be introduced to both J and dr/dx relations. We have made many such parametric studies but, as we shall see below, there are already enough parameters so that further parameterization is not felt to be justified, nor presented in this thesis, at this stage of our investigation. We can also obtain a wide range of results by such adjustments to the thermodynamic properties, especially surface tension, which are poorly understood for the small clusters predicted for our expansions but, again, the uncertainties do not warrant such parameterization until we obtain experimental results with which to compare.

Because of the rapid rate of expansion, and subsequent large cooling rates, there have been several studies and calculations made (Collins 1955; Andres and Boudart 1964) to correct the steady state nucleation rate, given above for $J = J_{ss}$, by introduction of a nucleation lag time τ. This results in the following correction to J to give the time-dependent nucleation rate:

$$J_t = J_{ss} \left(1- \exp(-t/\tau)\right) \tag{5.16}$$

where t is the flow time and

$$\tau = \pi r^{*2} \frac{\sqrt{2\pi mkT}}{P} \left(\frac{\rho_L}{m}\right)^2 \frac{kT}{\sigma} \tag{5.17}$$

We will show the effects of such a lag time below.

This classical analysis indicates that the critical size clusters are often close to the expected size of dimers, only two CO_2 molecules, well below the expected range of validity of the classical rate of formation relation given by J above. We have also therefore investigated the use of a kinetic formulation for J_2 for dimers. This is a well known kinetic model which has been used to analyze free-jet expansions in molecular beam research (Knuth 1977; Murphy and Miller 1984; McDaniels *et al.* 2002). The dimer kinetic model is based on the following two-step mechanism, involving an intermediate dimer complex:

$$A + A \underset{kf*}{\overset{kr*}{\Leftrightarrow}} A_2^* \tag{5.18}$$

$$A_2^* + A \underset{kf}{\overset{kr}{\Leftrightarrow}} A_2 + A \tag{5.19}$$

where K_f and K_r are the forward and reverse rate constant.

The time rate of change of dimer mole fraction is (Knuth 1977; Murphy and Miller 1984):

$$\frac{dn_{A2}}{dt} = k_f \left[k_1 n_A{}^3 - n_{A2} n_A / k_2 \right] \tag{5.20}$$

$$K_1 = \frac{9}{2}\sqrt{2\pi}\sigma^3\left(\frac{\varepsilon}{kT}\right)^{1/2} \tag{5.21}$$

$$K_f = 1.4\pi\sigma^2\Omega^{(2,2)}\left(\frac{12kT}{\pi m_A}\right)^{1/2} 0.4\exp\left(-0.6\frac{\varepsilon}{kT}\right) \tag{5.22}$$

$$K_2 = 0.447\left(\frac{kT}{\varepsilon}\right)\left[\exp\left(\frac{\varepsilon}{kT}\right) - \left(1 + \frac{\varepsilon}{kT}\right)\right] \tag{5.23}$$

K_1 and K_2 are the rate and equilibrium constant for equations (5.18) and (5.19). σ and ε are Lennard-Jones intermolecular potential parameters. For CO_2, $\sigma = 3.996 \times 10^{-8}$ cm, $\Omega^{(2,2)} = 1.314\ (305/T)^{1/2}$, and $\varepsilon/k = 190$ K (Hirschfelder, 1954).

This dimer formation rate can be rearranged to provide an equivalent dimer formation rate in the same units as J in the classical theory, which becomes:

$$J_2 = \frac{N_A}{MW}\frac{\rho u}{D}\frac{dw_2}{dx} \tag{5.24}$$

where D is the diameter at the throat for the nozzle, and w_2 is the mass fraction of dimer.

In the following, we will compare this result with the classical result, and use it to replace J when the classical critical radius r^* is less than 5×10^{-10} m, about the size of a CO_2 dimer.

The parameters in the condensation model are clearly very sensitive to temperature and pressure and hence the expansion characteristics. When the condensed phase mass fraction becomes large, the kinetics are strongly coupled to the fluid flow conservation equations. Given the flow properties at position (i), the following algorithm is used to calculate the flow properties at position (i+1) along the convergent-divergent nozzle:

1. calculate r^*_i, J_i, μ_i , and μ_{i+1}
2. guess ρ_{i+1}
3. $u_{i+1} = \dfrac{\dot{m}(1-\mu_{i+1})}{A_{i+1}\rho_{i+1}}$
4. $h_{i+1} = h_{g_i} + \dfrac{{u_i}^2}{2} - \dfrac{{u_{i+1}}^2}{2} + \Delta\mu_{(i\text{ to }i+1)}\, h_{fg}$
5. $T_{i+1} = T_{i+1}(h_{i+1}, \rho_{i+1})$
6. calculate $T_{av} = \dfrac{1}{2}\left(T_i + T_{i+1}\right)$
7. $s_{i+1} = s_{i+1}(T_{i+1}, \rho_{i+1})$

   if $s_{i+1} \neq s_i + \dfrac{\Delta\mu_{(i\text{ to }i+1)}\, h_{fg}}{T_{av}}$, go back to step 2
8. $P_{i+1} = P_{i+1}(T_{i+1}, \rho_{i+1})$

The calculation begins downstream of the throat, and the condensation does not start until the flow enters into the super-saturation regime where $P_v > P$. Further, following results by Hill *et al.* (1963), the clusters are not assumed to begin to grow until $r(k) > 1.3r^*(i)$. To examine the approximation that the condensation is decoupled from the fluid dynamics, we simply set $\Delta\mu = 0$. The total mass flow rate, $\dot{m}$, is known in this calculation, for a given throat diameter and given stagnation conditions, from the solutions to Lax-Wendroff integration or from the isentropic orifice program above. The location of the shock in the diverging section is guessed and the properties at the exit of the nozzle are calculated using the Rieman shock equations, (see equations 2.33) followed by an isentropic calculation to the exit, neglecting any further condensation downstream of the shock wave, which reheats the flow. The shock position is iterated until $P_{exit} = 1$ atm.

The results indicated below are all for expansions for supercritical CO_2 from 80 bar and 70°C with an orifice diameter of 50 μm. We first give a few results which would be obtained assuming the condensation can be decoupled from the fluid mechanics.

Figure 5.1 shows the critical radius size of nucleated clusters as a function of position in the expansion based on fluid properties decoupled from the condensation. The hard sphere diameter of a single CO_2 molecule is ~ 4×10^{-10} meters. Figure 5.2 shows three corresponding nucleation rates, $J = J_{ss}$, J_t corrected for the nucleation time lag, and J_2 , the kinetic dimer formation rate (note the scale change). We see that while large clusters are thermodynamically stable and can grow very near the throat ($x < D$) the rate of formation for such clusters is very small, and that large rates only occur when the critical radius is of order 2 angstrom. Figure 5.2 also shows that, for this decoupled approximation and this expansion, the effect of nucleation lag time is substantial. We also see that the kinetic rate for dimer formation is substantially less than predicted by the classical theory for the same r^*, so that a reasonable model would be to use J_2 when $r^* < 5 \times 10^{-10}$ m. Finally, for this decoupled case, figure 5.3 shows the accumulated total condensed phase mass fraction along the nozzle for the three different models for J. It is clear that the decoupled calculation is not appropriate for any of these calculations because the total mass fraction rapidly exceeds one, even when suppressed by limiting the formation rate to J_2 or by considering the effects of nucleation lag time.

The coupled equations are expected to reduce the condensation by accounting for the heat release and the loss of vapor density. The resulting effects are not necessarily intuitive because of the sensitivity of the nucleation kinetics to temperature and pressure, affected both by the expansion itself as well as the heat release and vapor mass reduction. For the same conditions as above, figure 5.4 shows the critical radius size along the expansion, figure 5.5 shows the nucleation rates for two of the models, and figure 5.6 shows the resulting mass fraction just upstream of the shock wave. We find that the critical radius size now does not become so small that a dimer kinetic model is not necessary, that the nucleation rates are much smaller, and not significantly effected by the nucleation lag time, and that the mass fractions, while larger than we expected, reach about 25% of the flow (the mole fraction of condensed phase would be much less). Again, we can move these results by altering the surface tension or assigning a sticking coefficient to the condensation and growth rates. To place these results into some perspective, if we had done a textbook equilibrium thermodynamic computation for an isentropic expansion to the same pressure ratio, neglecting the coupling, we would compute a mass fraction of 33%.

Figures 5.7 and 5.8 give the resulting pressure and temperature profiles for the condensing flow. As a reference, the bottom curve on both plots represents the profiles when condensation is suppressed; the results agree exactly with our time-dependent

QOD calculations discussed in chapter 2. We find that, if this condensation model is correct, the shock wave position would move upstream by 0.6 diameter and the temperature upstream of the shock would be warmed by about 100 degrees. Neither of these two results were suggested, even qualitatively, by our experiments discussed in chapter 3. Previous work in this area suggests that it is certainly not surprising that this classical model might overestimate the condensation. If we had more quantitative data, especially better temperature profiles in the expansion or some spectroscopic measurement of cluster size, then we could empirically adjust some of the parameters in the model.

In our calculations, we did not pursue condensation, growth, or evaporation beyond the shock wave, an analysis which needs to done eventually. However, we can comment on what we might expect from the present model. Figure 5.9 is a histogram of cluster size distribution, in brackets of half angstrom (0.5×10^{-10} m), just before the shock. We see a bimodal distribution, often found for these calculations, with growth to about 8.5×10^{-10} m size clusters; such a cluster size would contain about five CO_2 molecules with a mass of about 220 amu. Note that the classical model permits a continuum distribution of r^* while in reality r^* would increase incrementally as each cluster grows by the addition of one CO_2 molecule. If we compute conditions downstream of the shock, we find $T = 383$ K and $P = 0.87$ bar, conditions which are outside the super-saturation regime (i.e. there would be no further nucleation). We would then expect that the existing clusters would begin to evaporate, provided there is sufficient heat transfer rates to the clusters to provide the necessary latent heat of evaporation. Equation (5.9) could be used to compute the resulting negative dr/dx. In any event we would expect that the cluster distribution upstream of the shock we have computed to represent the maximum amount of cluster formation.

Clearly, this important problem is unsolved and requires experimental data to proceed further. The time-of-flight mass spectrometer facility introduced and briefly discussed next can provide such data.

5.3 Mass Spectrometer Time-of-Flight Facility

To study the nature of the clustering in the free-jet expansion, we have interfaced a supercritical fluid expansion source directly to a time-of-flight (TOF) mass spectrometer system, capable of resolving masses up to 20,000 amu. If our condensation calculations above are a reasonable estimate of cluster size, then this facility can resolve the clusters in these CO_2 experiments and provide a valuable complement to our theoretical calculations. The technique is referred to as direct coupled mass spectrometry (DCMS) and, while difficult to apply to such high pressure expansions, it has been achieved previously for supercritical fluid expansions (Maharrey and Miller 2001). The TOF is a home built mass spectrometer that has been used in studies of laser desorption ionization of metal oxides and proteins in the laboratory of Dr. Robert Continetti (Andres *et al.* 2000; Rohrbacher and Continetti 2001). We have designed, fabricated, and tested the adaptation of our supercritical fluid sources to the TOF, greatly aided by Joe McDaniels. Unfortunately, quantitative cluster characterization has not yet been achieved and a considerable effort remains to optimize the ionization source and appropriate differential pumping.

Figure 5.10 is a schematic of the apparatus while figure 5.11 is a picture of the apparatus. The supercritical chamber that was added to the TOF to study the composition of the free-jet expansion is shown in figure 5.12. The beam source delivery system is identical to the one used in this thesis to study the free-jet expansions under ambient atmospheric conditions, and is shown with its accessories in figure 5.13. The design of the source chamber permits the free-jet expansion to be directed to a flat plate at atmospheric pressure, with a 100 μm hole in the plate, to pass a sample of the gases into the supercritical first vacuum chamber. The hat design also allows the plate to be removed and the nozzle block to be pushed inside the supercritical source chamber, as shown in figure 5.14, so that the expansion occurs directly into the first vacuum chamber. This avoids shock waves and passes the molecular beam directly into the mass spectrometer to accomplish the DCMS.

The supercritical source chamber is pumped by a root blower pump and the base pressure is maintained at 10^{-2} torr. The supercritical chamber is separated from the TOF ionization source chamber by a conical skimmer, shown in figure 5.15. The source chamber is pumped by a diffusion pump and maintained at a base pressure of 10^{-6} torr. The source chamber is separated from the detector chamber by a small aperture and a gate valve. The detector chamber is pumped by a turbo pump and is maintained at a base pressure of 10^{-9} torr.

The beam enters the source chamber through the skimmer to be ionized by electron bombardment with an electronic pulsing circuit that acts like a timing gate to release the ions in a very short time. The ions then enter the detection chamber through the small aperture that separates the ion source chamber from the detection chamber and travel a distance L to the detector. The TOF mass spectrometer measures the time it takes ions of different species that were produced at the same time to move from the source to the detector. If all ions are formed at the same energy $E= 1/2\ mv^2$ in the ionization source, and $v = L/t$, the mass of the ions reaching the detector can be calculated by the flight time t according to $m= 2E\ t^2/L^2$. A TOF mass spectrometer can theoretically measure unlimited mass ranges where a complete mass spectrum is acquired at one time. In reality though, it is limited by detection sensitivity, the timing resolution of the ionization source, and the timing dispersion possible by the length of the flight distance between the ionizer and detector. This TOF mass spectrometer has previously demonstrated sensitivity up to 20,000 amu.

Recently, Dr. Kenichi Iwamoto has extended our development of this TOF facility (significantly improving the pumping and ionization characteristics of the machine) to probe free-jets and observe small clusters. This facility will become a crucial component for future studies of supercritical fluid for this laboratory, especially when solutes are added, providing a complete characterization of the expansion composition and an understanding of the RESS process. The calculations we have described in chapters 2-5 above can then be readily extended to more complicated but interesting and useful systems.

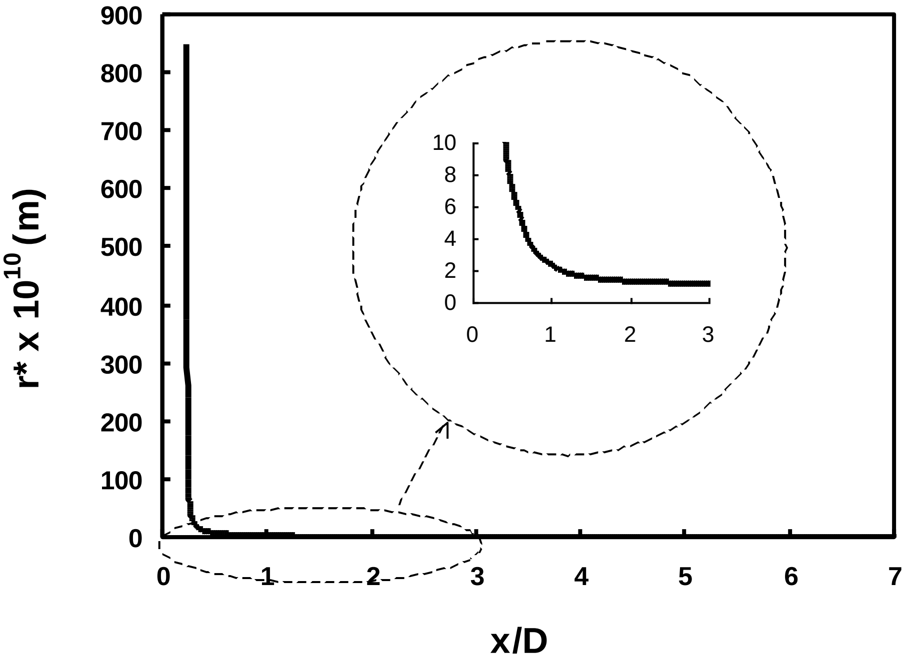


Figure 5.1: Critical size clusters r^* vs. x/D
(CO_2 at P_o= 80 bar, T_o= 70°C, decoupled QOD condensation calculation)

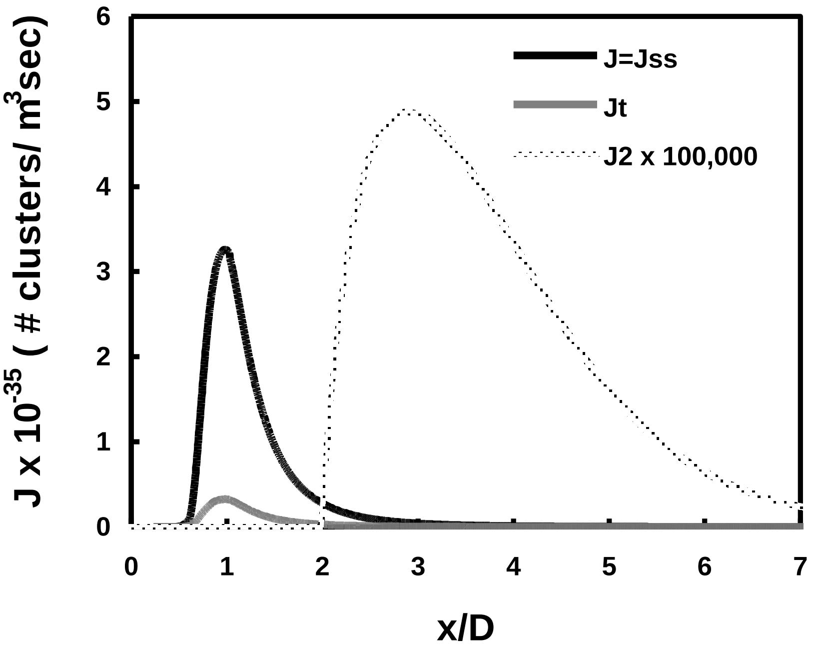


Figure 5.2: Nucleation rate vs. x/D for different models
(CO_2 at P_o= 80 bar, T_o= 70°C, decoupled QOD condensation calculation)

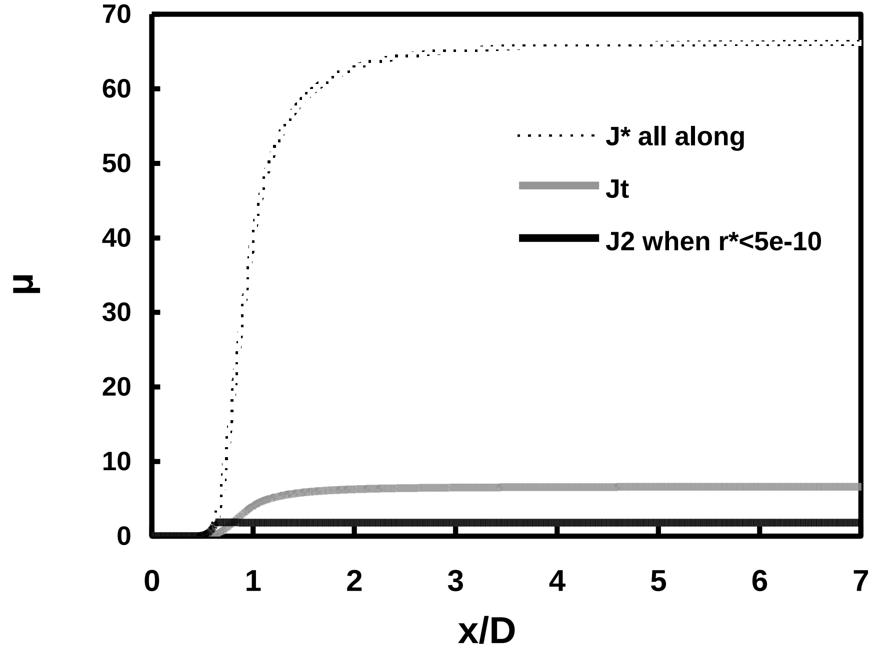


Figure 5.3: Mass fraction of condensed CO_2 vs. x/D
(P_o= 80 bar, T_o= 70°C, decoupled QOD condensation calculation)

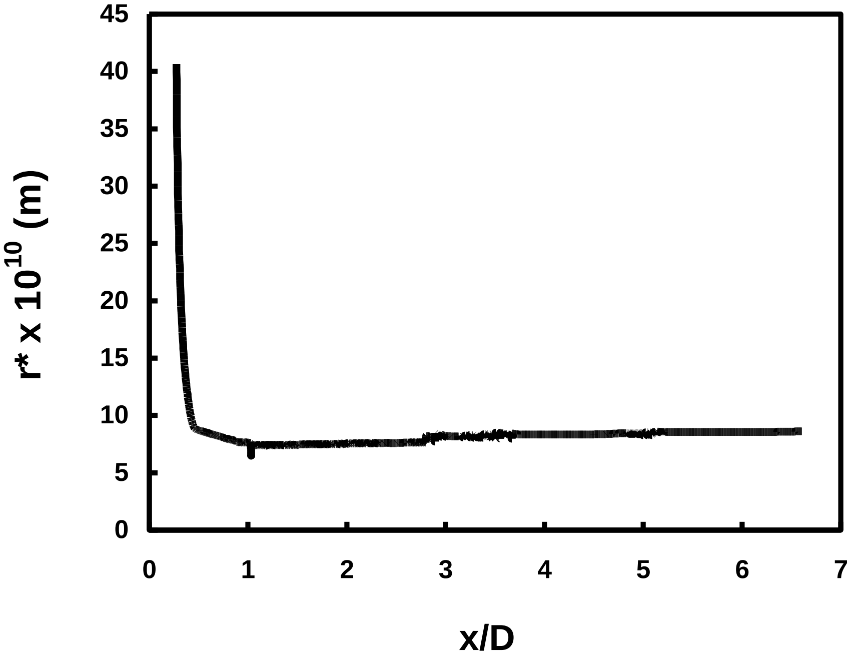


Figure 5.4: Critical size clusters r^* vs. x/D
(CO_2 at P_o= 80 bar, T_o= 70°C, coupled QOD condensation calculation)

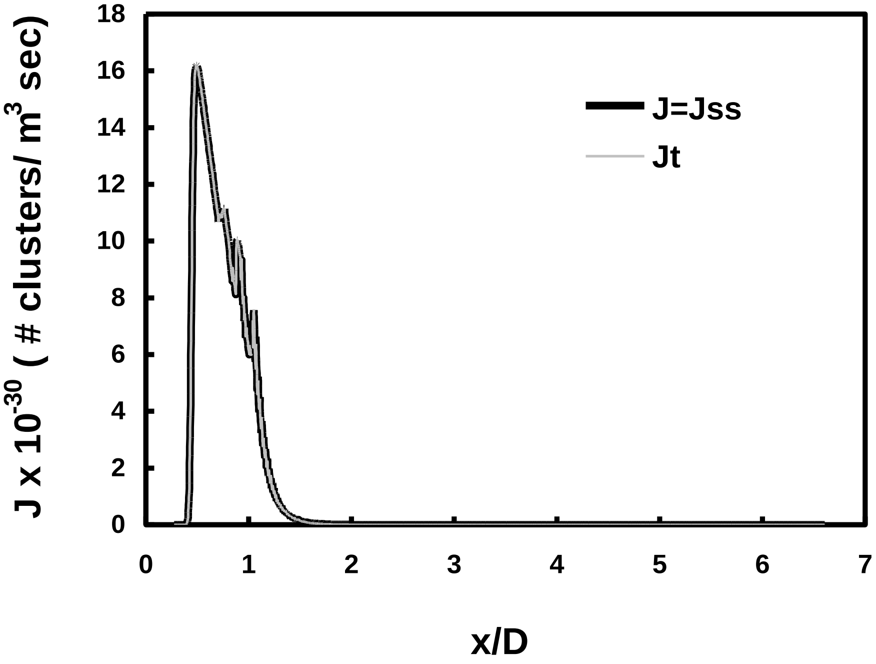


Figure 5.5: Nucleation rate vs. x/D for different models
(CO_2 at P_o= 80 bar, T_o= 70°C, coupled QOD condensation calculation)

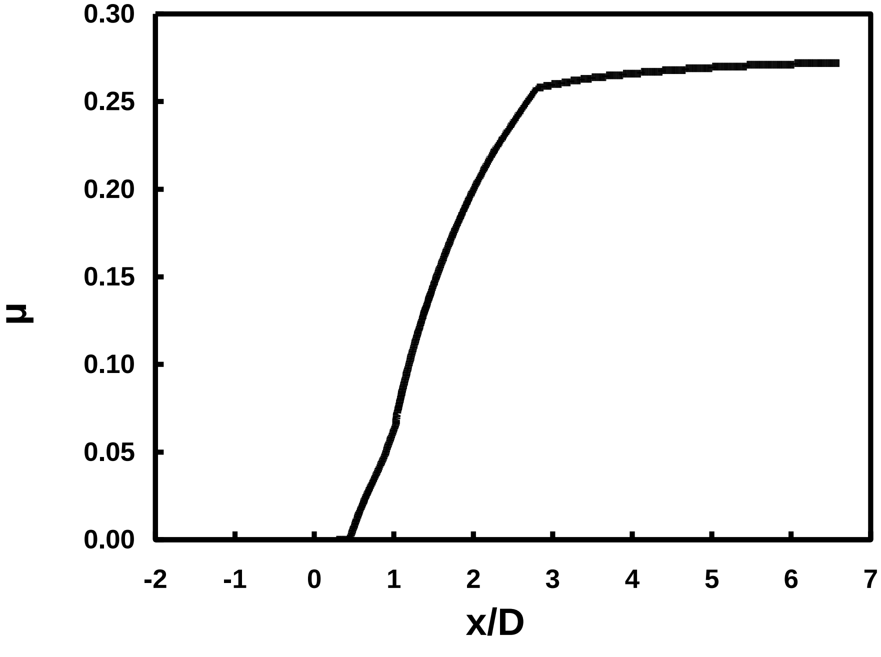


Figure 5.6: Mass fraction of condensed CO_2 vs. x/D
(P_o= 80 bar, T_o= 70°C, coupled QOD condensation calculation)

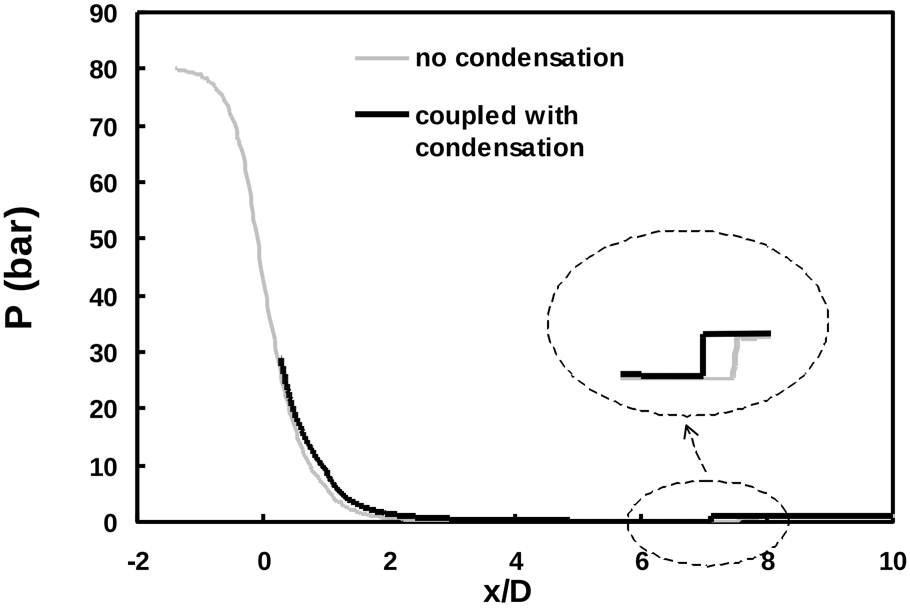


Figure 5.7: Pressure profile for the condensing flow
(CO$_2$ at P$_o$= 80 bar, T$_o$= 70°C, coupled QOD condensation calculation)

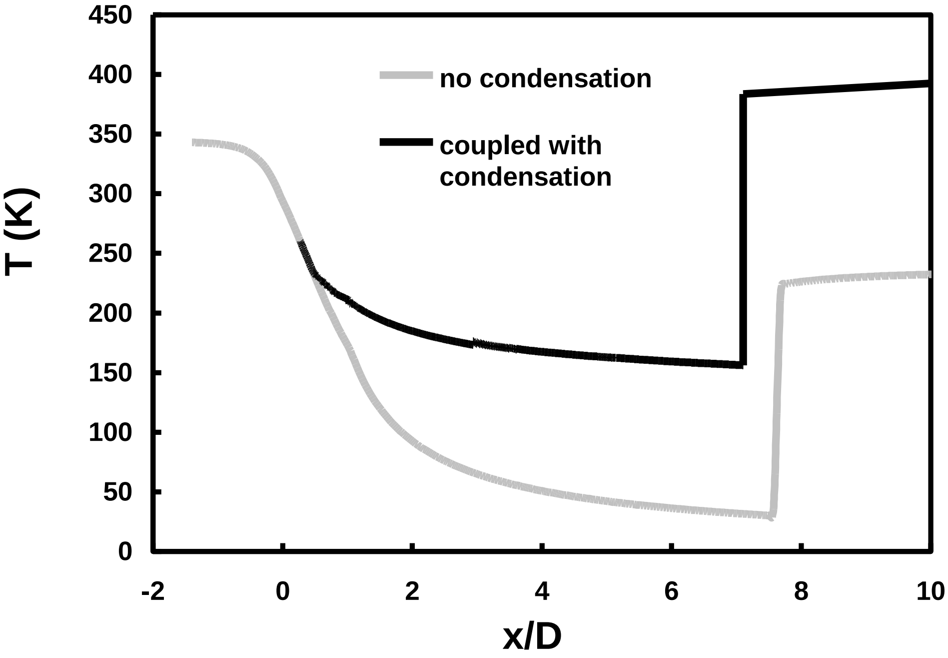


Figure 5.8: Temperature profile for the condensing flow
(CO$_2$ at P$_o$= 80 bar, T$_o$= 70°C, coupled QOD condensation calculation)

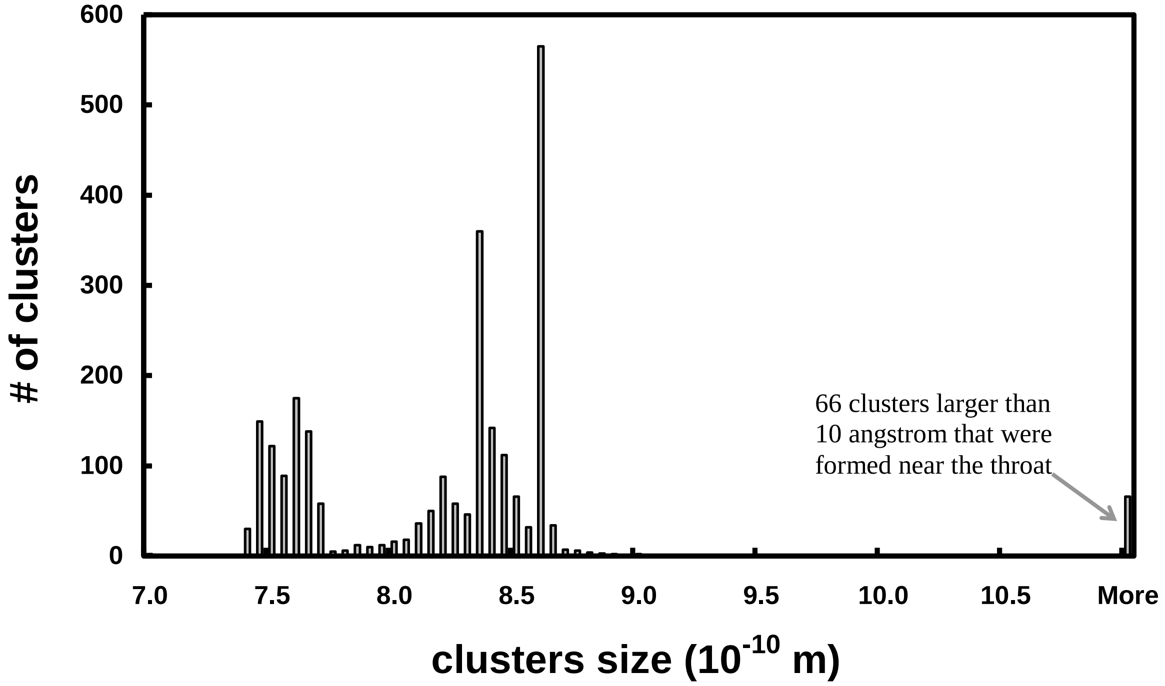


Figure 5.9: Size distribution of clusters formed in the coupled condensation (CO_2 at P_o= 80 bar, T_o= 70°C, coupled QOD condensation calculation)

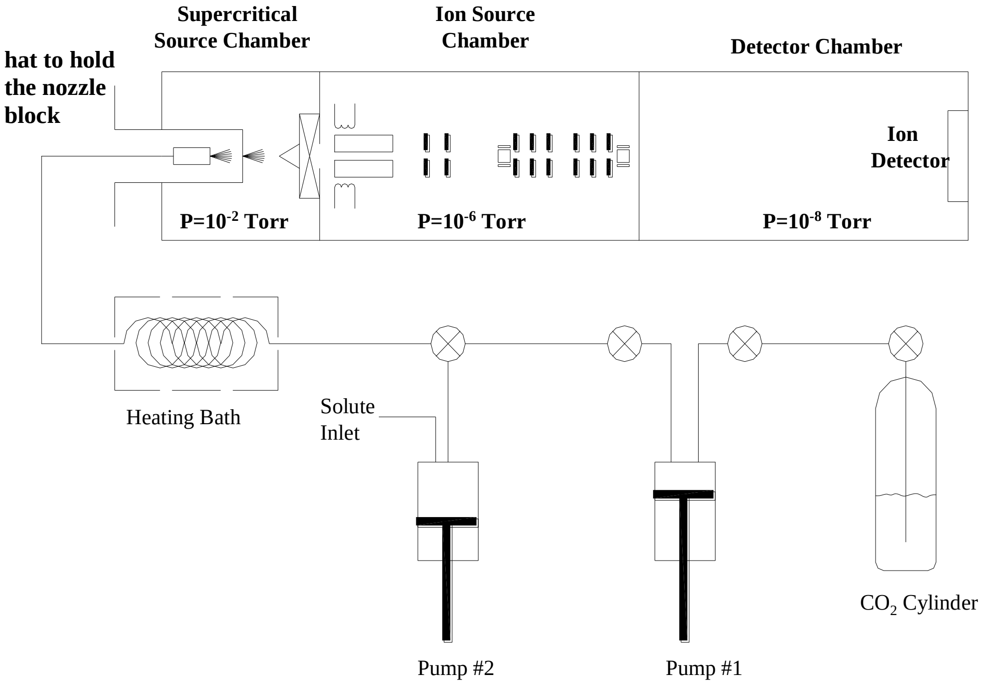


Figure 5.10: Schematic of the time-of-flight machine

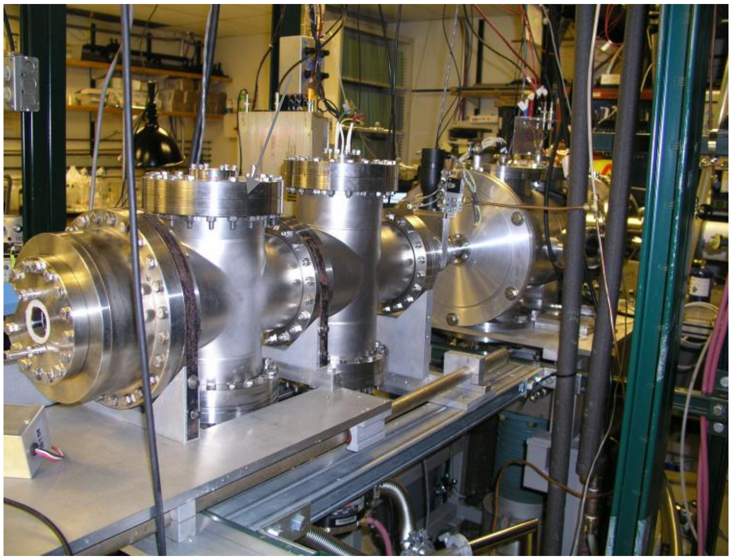


Figure 5.11: Time-of-flight machine

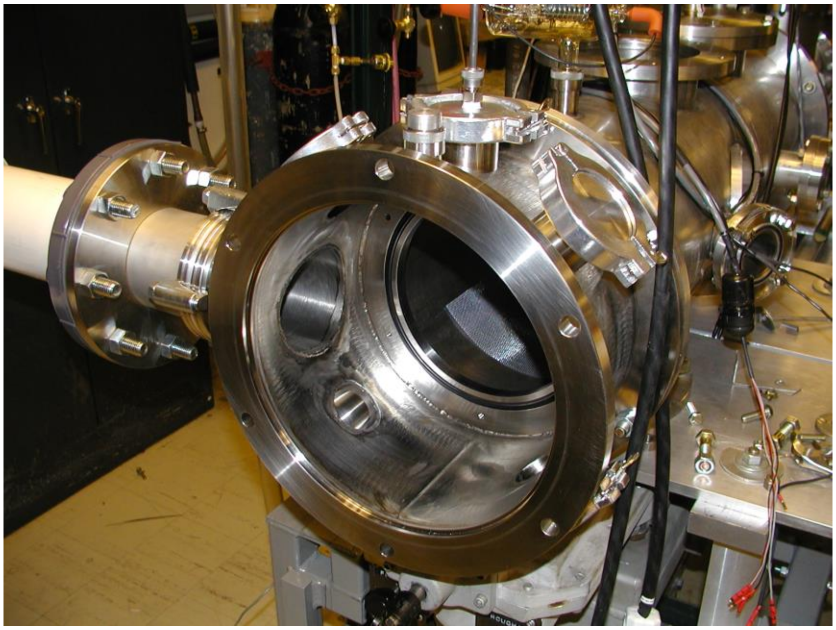


Figure 5.12: Supercritical source chamber added to time-of-flight

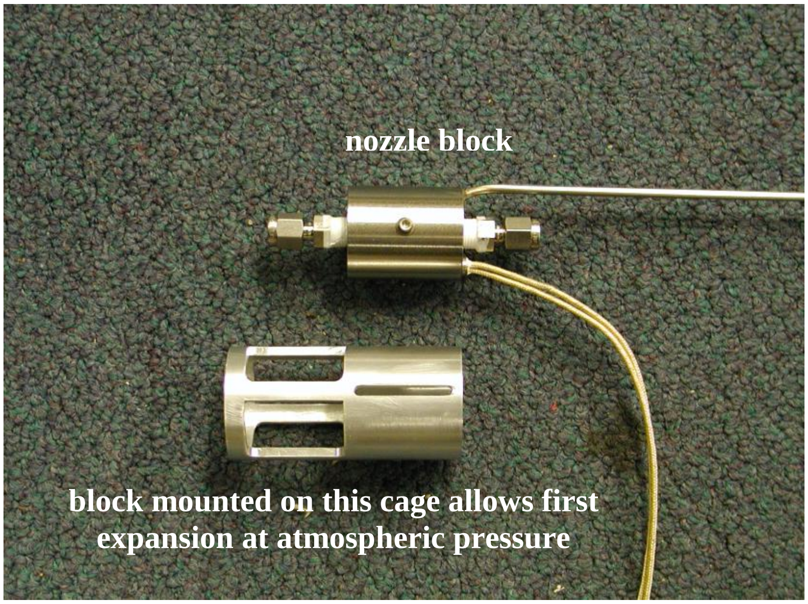


Figure 5.13: Nozzle block with accessories to mount inside the hat

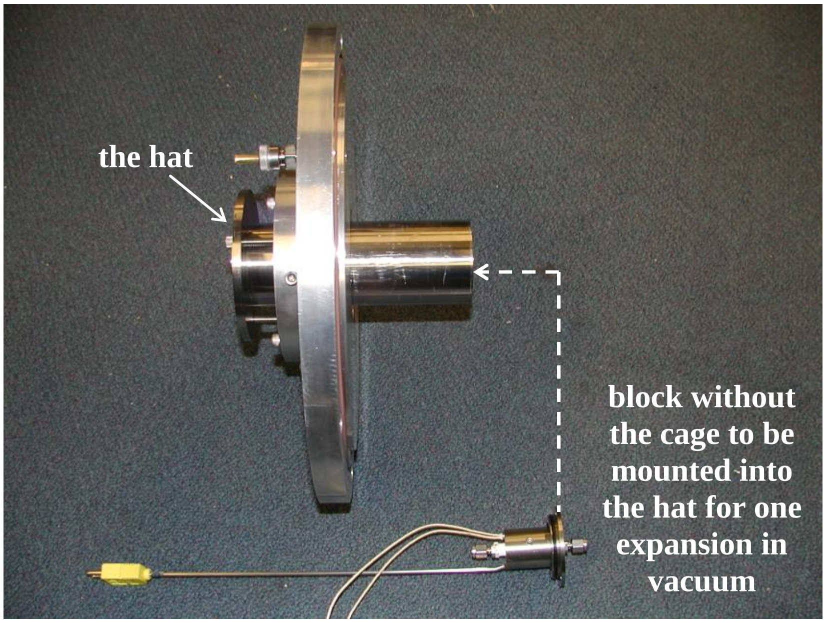


Figure 5.14: The hat designed to permit direct coupled mass spectrometry

Figure 5.15: Skimmer

REFERENCES

Abraham, O., Kim, S., and Stein, G., Homogeneous Nucleation of Sulfur Hexafluoride Clusters in Laval Nozzle Molecular Beams, J. Chem. Phys.**75** (1), pp 402 (1981)

Andres, R., and Boudart, M., Time Lag in Multistate Kinetics: Nucleation, The Journal of Chemical Physics, **42**, Number 6, pp 2057 (1964)

Andrews, L.S., Rohrbacher, A., Laperle, C. M., and Continetti, R. E., Laser Desorption-Ionization of Transition Metal Atoms and Oxides from Solid Argon., J. Phys. Chem. A **104**, pp 8173 (2000)

Bayazitoglu, Y., Brotzen, F. R., and Zhang, Y., Metal Vapor Condensation in a Converging Nozzle, Nanostructured Materials **7**, No 7, pp 789 (1996)

Collins, F., Time Lag in Spontaneous Nucleation Due to Non-Steady State Effects, Zeitschrift für Elektrochemie, Bd. 59, Nr. 5, pp 404 (1955)

Hill, P. G., Witting, H., Demetri, E. P., Condensation of Metal Vapors During Rapid Expansion, J. Heat Transfer, pp 303 (November 1963)

Hirschfelder, J. O., Curtiss, C. F., Bird, R. B., Molecular Theory of Gases and Liquids, John Wiley & Sons, New York (1954)

Knuth, E. L., Dimer Formation Rate Coefficients from Measurements of Terminal Dimer Concentrations in Free-Jet Expansions, Journal of Chemical Physics **66**, No. 8, pp 3515 (1977)

Kotake, S., and Glass, I. I., Univ. of Toronto Institute for Aerospace Studies Review No. 42 (1978)

Kwauk, X., and Debenedetti, P., Mathematical Modeling of Aerosol Formation by Rapid Expansion of Supercritical Solutions in a Converging Nozzle., J. Aerosol Science **24**, pp 445 (1993)

Maharrey, S., and Miller, D. R., A Direct Sampling Mass Spectrometer Investigation of Oxidation Mechanisms for Acetic Acid in Supercritical Water, J. Phys. Chem. A **105**, pp 5860 (2001)

McDaniels, J., Continetti, R. E., and Miller, D. R., in Proceedings 23rd Int. Symp. Rarefied Gas Dynamics, **663**, pp 670, Whistler, Canada (2002)

Murphy, H., and Miller, D., Effects of Nozzle Geometry on Kinetics in Free-Jet Expansions, J. Physical Chemistry **88**, pp 4474 (1984)

Rohrbacher, A., and Continetti, R. E., A Multiple-Ion-Beam Time-of-Flight Mass Spectrometer, Rev. Sci. Instrum. **72**, pp 3386 (2001)

Stever, H. G., Condensation Phenomena in High Speed Flows, in Fundamentals of Gas Dynamics, **3**, edited by Emmons, H., Princeton University Press, pp 526 (1958)

Weber, M., and Thies, M., Undestanding the RESS Process, in Supercritical Fluid Technology in Materials Science and Engineering, edited by Y. Sun, pp387, New York (2002)

Wegener, P., in Nonequilibrium Flows, edited by P. Wegener, pp 163, Marcel Dekker, New York (1969)

Zharkova, N., Prokkoev, V., Rebrov, A., Skovorodko, P., and Yarygin, V., The Effects of Nonequilibrium Condensation and Vibrational Relaxation in Supersonic Expansion of Carbon Dioxide, 11th Int. Symp. On Rarefied Gas Dynamics, Cannes (1978)

Appendix A: Equation of state for carbon dioxide

The 27-parameter equation of state for CO_2 derived by Huang *et al.* (1984) was used and given in chapter 2. This equation is repeated below:

$$
\begin{aligned}
Z = \frac{P}{\rho RT} = \; & 1 + b_2\rho' + b_3\rho'^2 + b_4\rho'^3 + b_5\rho'^4 + b_6\rho'^5 + b_7\rho'^2 \exp[-c_{21}\rho'^2] \\
& + b_8\rho'^4 \exp[-c_{21}\rho'^2] + c_{22}\rho' \exp[-c_{27}(\Delta T)^2] \\
& + c_{23}\frac{\Delta\rho}{\rho'} \exp[-c_{25}(\Delta\rho)^2 - c_{27}(\Delta T)^2] \\
& + c_{24}\frac{\Delta\rho}{\rho'} \exp[-c_{26}(\Delta\rho)^2 - c_{27}(\Delta T)^2]
\end{aligned}
$$

where T denotes temperature in K, ρ is density in Kg/m^3, $T' = T/T_c$, $\rho' = \rho/\rho_c$, $\Delta T = 1-T'$, $\Delta\rho = 1-\rho'$, the parameters ci and bi are given below:

i	ci	i	Ci
1	0.376194	15	-2.79498
2	0.118836	16	5.62393
3	-3.04379	17	-2.93831
4	2.27453	18	0.988759
5	-1.23863	19	-3.04711
6	0.250442	20	2.32316
7	-0.115350	21	1.07379
8	0.675104	22	-0.599724E-4
9	0.198861	23	0.885339E-4
10	0.216124	24	0.316418E-2
11	-0.583148	25	10.
12	0.119747E-1	26	50.
13	0.537278E-1	27	80,000.
14	0.265216E-1		

$$b_2 = \left(c_1 + \frac{c_2}{T'} + \frac{c_3}{T'^2} + \frac{c_4}{T'^3} + \frac{c_5}{T'^4} + \frac{c_6}{T'^5} \right)$$

$$b_3 = \left(c_7 + \frac{c_8}{T'} + \frac{c_9}{T'^2} \right)$$

$$b_4 = \left(c_{10} + \frac{c_{11}}{T'} \right)$$

$$b_6 = \left(\frac{c_{14}}{T'} \right)$$

$$b_7 = \left(\frac{c_{15}}{T'^3} + \frac{c_{16}}{T'^4} + \frac{c_{17}}{T'^5} \right)$$

$$b_8 = \left(\frac{c_{18}}{T'^3} + \frac{c_{19}}{T'^4} + \frac{c_{20}}{T'^5} \right)$$

Appendix B: Subroutine to solve for T(ρ,T)

For many steps in the calculations, we needed to invert an equation of state in order to calculate T. Subroutines that use the secant method provided such a root finding calculation and an example of calculating T, given h(ρ,T) and ρ is given below:

```
        subroutine tfind (t,h,v,tstart)
        implicit none
        double precision t1,p1,h1,s1,v1,a,b,h1r,t,v,p,hr,h
        double precision acp,bcp,ccp,dcp
        double precision MW,R,Ru,tstart,tol
        double precision tref,pref,vref,hadj,teta
        teta=960.1
        tref=273.
        pref=1.e5
        vref=1./2.
        tol=1.0e-5
        MW =44.02
        Ru=8314.
        R=Ru/MW

        a=64.64e5/(MW**2.0)
        b=0.02969/MW
        t=200.
        call rootfind(h,h1,h1r,t,a,b,v,t1,acp,bcp,ccp,dcp,tol)
        end


        subroutine rootfind(h,h1,h1r,t,a,b,v,t1,acp,bcp,ccp,dcp,tol)
        implicit none
        double precision f1,f2,h1,h,h1r,t1,t,tol,e,df,terr,fdum
        double precision acp,bcp,ccp,dcp,a,b,v1,v,p1,ro1,s1,z1
        integer i,n
        e=1.e-6
        n = 50
        do 20 i= 1,50

                call function (f1,h1,h,h1r,t,v,t1,a,b,acp,bcp,ccp,dcp)
                call function (f2,h1,h,h1r,t+e,v,t1,a,b,acp,bcp,ccp,dcp)
                df=(f2-f1)/e
                fdum=t
                t=t-f1/df
                terr=(t-fdum)/t
                if(abs(terr).lt.tol) then
                        goto 90
                end if
20      continue

90      return
        end
```

```
         subroutine function (func,h1,h,h1r,t,v,t1,a,b,acp,bcp,ccp,dcp)
         implicit none
         double precision func,h1,h,h1r,t,b,v,a,acp,bcp,ccp,dcp,t1
         double precision MW,R,Ru
         double precision tref,pref,vref,hadj,teta,hr,p
         teta=960.1
         tref=273.
         pref=1.e5
         vref=1./2.

         MW =44.02
         Ru=8314.
         R=Ru/MW
         p=((R*t)/(v-b))-(a/(t**(1.0/2.0)*v*(v+b)))

         hr = -(((R*t)*(( (3*a)/(2*b*R*t**(3.0/2.0))*(log(1+(b/v))))
&                   -(p*v)/(R*t)+1)) )

         hadj = - ( (7./2.)*R*tref +
&        R*(teta+( 2.*teta/(exp(teta/tref)-1. ) )) )

         func = h-
&        ( hr + (7./2.)*R*t + R*(teta+( 2.*teta/(exp(teta/t)-1. ) ))
&                                  +hadj)
         end
```

Appendix C: FORTRAN quasi-one dimensional codes for flow inside the capillary

```
        program finalRedlich-Kwong
        implicit none
        double precision po,to,vo,ho,so,ro1,area2
        double precision a,b,h1r,t2,s2,v2,p2,h2r,h2,ro2,s2r,s1r,v1,z1
        double precision h3, v3, vel3, t3, ro3,p3,s3
        double precision nu,d,l,re,fre,pi
        double precision dpdv,gamma
        double precision x,x2,x3
        double precision MW,R,Ru
        double precision tprime,pprime,c,mach
        double precision acp,bcp,ccp,dcp
        double precision mdot,vel2,eps,mdot_inc
        integer i,k

        open(18,file='results',status='unknown')
        open(19,file='state2',status='unknown')

c       'This program uses the Redlich-Kwong equations of state.'
        eps=.0001

        MW =44.02
        Ru=8314.
        R=Ru/MW

        a=64.64e5/(MW**2.0)
        b=0.02969/MW

        ro1=100.
        to = 343.

        do 105 i=1,10000
              ro1=ro1+.5
              vo= 1.0/ro1
              po=((R*to)/(vo-b))-(a/(to**(1.0/2.0)*vo*(vo+b)))
              if(po/100000.gt.100.and.po/100000.lt.101)then
              go to 106
              end if
105     continue
106   continue

        d=.000100
        l=100*(100.e-6)

        mdot=.007/1000.
        mdot_inc=.001/1000.

        do 112 k=1,600
```

```
333     if(mach.gt.1.0) then
                mdot = mdot - mdot_inc
                mdot_inc = mdot_inc*.5
        end if

        mdot=mdot+mdot_inc

        call hfind(ho,to,vo)
        call sfind(so,to,vo)

        area2=(pi*d**2.0)/4.0

        do 20 i=1,100
                ro2=ro1-.001*ro1
                v2= 1.0/ro2
                vel2=mdot/(ro2*area2)
                h2=ho-((vel2**2.0)/2.0)

                call tfind (t2,h2,v2,to)
                p2=((R*t2)/(v2-b))-(a/(t2**(1.0/2.0)*v2*(v2+b)))

                call sfind (s2,t2,v2)

                if (abs(s2-so)/so.lt.eps) then
                        go to 60
                end if
                ro1=ro2
20      continue
60      continue

        dpdv=(-R*t2*((v2-b)**-2.0))+ ((a/t2**(1.0/2.0))*(2.0*v2+b)*
   &    ((v2**2.0+b*v2)**2.0))

c       start analysis of flow in the tube with friction

        x2=0.0
        x=x2
        do 30 i=1,30000
                ro3=ro2-(0.02*ro2)
                v3=1.0/ro3
                vel3=(mdot/area2)/ro3
                h3=ho-(vel3**2.0/2.0)

                call tfindsv (tprime,s2,v3,t2)

                pprime=((R*tprime)/(v3-b))-(a/(tprime**(1.0/2.0)*v3*(v3+b)))
                c=sqrt((pprime-p2)/(ro3-ro2))

                mach=vel3/c
                if(mach.gt.1.)then
                        write(6,*)'mach larger than 1'
                        go to 1100
                end if

                call tfind(t3,h3,v3,t2)
                call sfind(s3,t3,v3)
```

```
                    p3=((R*t3)/(v3-b))-(a/(t3**(1.0/2.0)*v3*(v3+b)))


                    if (t3.gt.320.and.t3.lt.390.)then
                              call viscofind (t3,p3,nu)
                    else if (t3.lt.304.) then
                              nu = 10.e-6
c                                       nu =  (exp ( 3.3882 +
c   &                         ( (1.3423*ro3/1000.)/(1.909-(ro3/1000.)) ) ) )/1000000.
                    else
                    nu=60.e-6
                    endif

                    re=mdot*(4.0/(pi*d*nu))
                              if(re.lt.100000.)then
                                        fre=.316*(re**(-1.0/4.0))
c         friction factor below for laminar flow, Re less than 2300
                              else
                              fre=.316*(re**(-1.0/4.0))
                    endif

                    x3=x2+
   &                d*((area2)/(-2.0*fre*mdot))*(2.0/(vel3+vel2))*
   &                (((p3-p2))+((mdot/area2)*(vel3-vel2)))

                    x=x+(x3-x2)
                    write(18,81)100*x,t3,fre, nu,re
81                  format(f9.5,2x,f10.0,2x,f10.5,2x,f20.10,2x,f16.5)

                    if(x.gt.l)then
                    go to 80
                    end if
                    x2=x3
                    p2=p3
                    ro2=ro3
                    vel2=vel3
                    h2=h3
                    t2=t3
                    v2=v3
                    s2=s3
30        continue
1100      continue

80        continue
          write(6,*)mdot,mach

          if(mach.gt..999.and. mach.lt.1.)then
          write(6,*)'DONE'
          go to 1101
          end if

112       continue
1101      continue
          write(6,*)'   rho=          ','To=         ','Po='
          write(6,*)1./vo,to,po/100000.
103       format(f6.5,f7.2,3x,f7.2,1x,f15.2,3x,f6.2,2x,
```

```
     $     f7.4,3x,f9.2,2x,f7.2)
           end
```

Appendix D: FORTRAN codes for the axisymmetric free-jet expansion of CO_2

```
              program plume

              implicit none
              integer nmax
              integer imax,jmax
              integer i,ii,n,j
              double precision small, dx, dt, alp ,dy
              double precision c1,cx,cy,gam,eps,phi
c              i just changed the time step to .028/8, it was .028/4, but
c              the program failed after ~1000 time steps
c             original imax=226, jmax=242
              parameter (small = 1.0e-20, nmax =40000, imax = 226, jmax = 242,
     &                                dx = 1/24., dy = 1/24.,dt = .023/4., alp = 1.2)

              double precision u(imax+1,jmax+1,4), u2(imax+1,jmax+1,4)
              double precision du(imax+1,jmax+1,4)
              double precision du1(imax+1,jmax+1,4)
              double precision du2(imax+1,jmax+1,4)
              double precision jj(imax+1,jmax+1,4)
              double precision jj2(imax+1,jmax+1,4)
              double precision f(imax+1,jmax+1,4)
              double precision f2(imax+1,jmax+1,4)
              double precision g(imax+1,jmax+1,4)
              double precision g2(imax+1,jmax+1,4)

              double precision up1(imax+1,jmax+1,4), u2p1(imax+1,jmax+1,4)
              double precision dup1(imax+1,jmax+1,4)
              double precision du1p1(imax+1,jmax+1,4)
              double precision du2p1(imax+1,jmax+1,4)
              double precision jjp1(imax+1,jmax+1,4)
              double precision jj2p1(imax+1,jmax+1,4)
              double precision fp1(imax+1,jmax+1,4)
              double precision f2p1(imax+1,jmax+1,4)
              double precision gp1(imax+1,jmax+1,4)
              double precision g2p1(imax+1,jmax+1,4)


              double precision  mach(imax+1, jmax+1)
              double precision L,x,y
              double precision ro(imax+1,jmax+1)
              double precision t(imax+1,jmax+1)
              double precision us(imax+1,jmax+1)
              double precision vs(imax+1,jmax+1)
              double precision us2(imax+1,jmax+1)
              double precision vs2(imax+1,jmax+1)
              double precision p(imax+1,jmax+1)
              double precision pp1(imax+1,jmax+1)
              double precision p2p1(imax+1,jmax+1)
```

```
        double precision p2(imax+1,jmax+1)
        double precision e(imax+1,jmax+1)
        double precision ep1(imax+1,jmax+1)

        double precision e2(imax+1,jmax+1)
        double precision ro2(imax+1,jmax+1)
        double precision t2(imax+1,jmax+1)

        double precision rop1(imax+1,jmax+1)
        double precision tp1(imax+1,jmax+1)
        double precision usp1(imax+1,jmax+1)
        double precision vsp1(imax+1,jmax+1)
        double precision us2p1(imax+1,jmax+1)
        double precision vs2p1(imax+1,jmax+1)
        double precision pressp1(imax+1,jmax+1)
        double precision press2p1(imax+1,jmax+1)
c       double precision ep1(imax+1,jmax+1)
        double precision e2p1(imax+1,jmax+1)
        double precision ro2p1(imax+1,jmax+1)
        double precision t2p1(imax+1,jmax+1)

        double precision s(imax+1,jmax+1)
        double precision h(imax+1,jmax+1)


        double precision tt1,ppp1,ss1,to,roo,ao,po,ho,eo,cp,R,cv
        double precision pj,roj,tj,ej,hj,uj,Zo,roamb,tamb,pamb

        double precision  torif,roorif,MW,d,a,b
        double precision acp,bcp,ccp,dcp
        double precision  t1,p1,h1,v1,h1r
        double precision rorc,ttc,vvc,eec,tinitp,tinitc,tol
        double precision rorp,ttp,vvp,eep
        double precision vo,vj

c       the files below are to be used as IC for viscosity
        open(18,file='usrk',status='unknown')
        open(19,file='vsrk',status='unknown')
        open(20,file='temprk',status='unknown')
        open(21,file='densityrk', status='unknown')

        open(22,file='centerline',status='unknown')
        open(40,file='plate',status='unknown')

        open(51,file='usig',status='unknown')
        open(52,file='vsig',status='unknown')
        open(53,file='tempig',status='unknown')
        open(54,file='densityig',status='unknown')

        open (61,file='denscontour',status='unknown')
        open (62,file='velocityu',status='unknown')
        open (63,file='velocityv',status='unknown')
```

```
      tinitp=340.
      tinitc=340.

      tol=1.0e-5


      MW=44.02
      R=8314./MW
      gam=1.4
      a=64.64e5/(MW**2.0)
      b=0.02969/MW

      acp=2.401
      bcp=8.735e-3
      ccp=-6.607e-6
      dcp=2.002e-9

      t1=300.
      p1=20.0e5
      h1=404770
      v1=.02536
      h1r=((R)*t1)*(( (3*a)/(2*b*R*t1**(3.0/2.0))*(log(1+(b/v1))))
    &  -(p1*v1)/(R*t1)+1)



      cp= (gam/(gam-1))*R
      cv=cp-R

c     d is diameter of orifice
c     d=50.e-6

      tamb=300.
      roamb=1.7

      pamb =  ((R*tamb)/( 1.0/ (roamb)-b))-
    &                   (a/(tamb**(1.0/2.0)
    &                   * (1.0/roamb )*( 1.0/roamb +b)))
      write(6,*)'Zamb=',pamb/(roamb*tamb*R)
      write(6,*)'pamb=',pamb,'roamb=',roamb
      write(6,*)'tamb=',tamb
      write(6,*)

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     enter source conditions, fix two
      to=343.
c     roo=445 when P ratio is 137
c     roo is 178 when P ratio is 80....roo is 257 when P ratio is 100.
c     see page 57B7
      roo=178.
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      vo=1./roo
      po =  ((R*to)/( 1.0/ (roo)-b))-
    &                   (a/(to**(1.0/2.0)
    &                   * (1.0/roo )*( 1.0/roo +b)))
```

```
          ho=h1+h1r-
   &                ((R*to)*(( (3*a)/(2*b*R*to**(3.0/2.0))*(log(1+(b/vo))))
   &                -(po*vo)/(R*to)+1))
   &         +(R)*((acp*(to-t1))+
   &                ((bcp/2.0)*((to**2.0)-(t1**2.0)))+
   &                ((ccp/3.0)*((to**3.0)-(t1**3.0)))+
   &                ((dcp/4.0)*((to**4.0)-(t1**4.0))))
          write(6,*)'po=',po,'roo=',roo
          write(6,*)'to=',to
          write(6,*)
c         ho=697639.
c         eo=cv*to
          eo=ho-po/roo
          Zo=po/(roo*R*to)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
          tj=299.
          roj=120.
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
          uj=213.
          vj=1./roj
          pj =  ((R*tj)/( 1.0/ (roj)-b))-
   &                          (a/(tj**(1.0/2.0)
   &                          * (1.0/roj )*( 1.0/roj +b)))

          hj=h1+h1r-
   &                ((R*tj)*(( (3*a)/(2*b*R*tj**(3.0/2.0))*(log(1+(b/vj))))
   &                -(pj*vj)/(R*tj)+1))
   &         +(R)*((acp*(tj-t1))+
   &                ((bcp/2.0)*((tj**2.0)-(t1**2.0)))+
   &                ((ccp/3.0)*((tj**3.0)-(t1**3.0)))+
   &                ((dcp/4.0)*((tj**4.0)-(t1**4.0))))



c         hj=682640.
          ej=hj-(pj/roj)

c         eps is = 1 for axisymmetric and eps = 0 for 2D problems
          eps = 1.0

          write(6,*)'pj=',pj,'roj=',roj
          write(6,*)'tj=',tj
          write(6,*)'uj=',uj
c         write(6,*) 'source',po,roo,to

          do 10 i=1,imax
                    do 20 j=1,jmax
                              read(51,*)us(i,j)
                              read(52,*)vs(i,j)
                              read(53,*)t(i,j)
                              read(54,*)ro(i,j)
                    p(i,j) = ( ((R*t(i,j)*to)/( 1.0/ (ro(i,j)*roo)-b))-
   &                          (a/((t(i,j)*to)**(1.0/2.0)
   &                          * (1.0/(ro(i,j)*roo) )*( 1.0/(ro(i,j)*roo) +b))) )/po
                    e(i,j) =( (h1+h1r-((R*t(i,j)*to)*(( (3*a)/
```

```
   &                    (2*b*R*(t(i,j)*to)**(3.0/2.0))
   &                    *(log(1+(b/(1./(ro(i,j)*roo)))))
   &                    -(p(i,j)*po*(1./(ro(i,j)*roo)))/(R*(t(i,j)*to))+1))
   &               +(R)*
   &               ((acp*((t(i,j)*to)-t1))+
   &                    ((bcp/2.0)*(((t(i,j)*to)**2.0)-(t1**2.0)))+
   &                    ((ccp/3.0)*(((t(i,j)*to)**3.0)-(t1**3.0)))+
   &                    ((dcp/4.0)*(((t(i,j)*to)**4.0)-(t1**4.0))))-
   &                    ( (p(i,j)*po)/(ro(i,j)*roo) ) ) )/(to*R)
c       write(61,*)ro(i,j)
20      continue
10      continue



        y = dy/2.

        do 50 j=1,jmax

                    do 60 i=1,imax
                          u(i,j,1) =  ro(i,j)
                          u(i,j,2) =  ro(i,j) * us(i,j)
                          u(i,j,3) =  ro(i,j) * vs(i,j)
                          u(i,j,4) = ( ( (1./2.)*(us(i,j)**2.+ vs(i,j)**2.) )
   &                                         + e(i,j) )*ro(i,j)

                          f(i,j,1) = ro(i,j) * us(i,j)
                          f(i,j,2) = ro(i,j) * us(i,j) * us(i,j)
   &                                                   + p(i,j)*Zo
                          f(i,j,3) = ro(i,j) * us(i,j) * vs(i,j)
                          f(i,j,4) = ((( ( (1./2.)*(us(i,j)**2.+ vs(i,j)**2.) )
   &                                         + e(i,j) )*ro(i,j))+p(i,j)*Zo)*us(i,j)

                          g(i,j,1) = ro(i,j) * vs(i,j)
                          g(i,j,2) = ro(i,j) * us(i,j) * vs(i,j)
                          g(i,j,3) = ro(i,j) * vs(i,j) * vs(i,j)
   &                                                  +p(i,j)*Zo
                          g(i,j,4) = ((( ( (1./2.)*(us(i,j)**2.+ vs(i,j)**2.) )
   &                                         + e(i,j) )*ro(i,j))+p(i,j)*Zo)*vs(i,j)

                          jj(i,j,1) = eps * (  (1.0/y)*ro(i,j)*vs(i,j)  )
                          jj(i,j,2) = eps * ( (1.0/y)*ro(i,j)*us(i,j)*vs(i,j) )
                          jj(i,j,3) = eps * ( (1.0/y)*ro(i,j)*vs(i,j)*vs(i,j) )
                          jj(i,j,4) = eps*( (1./y)*((( ( (1./2.)*(us(i,j)**2.+
   &                                         vs(i,j)**2.) )
   &                                         + e(i,j) )*ro(i,j))+p(i,j)*Zo)*vs(i,j)  )

60                  continue

        y=y+dy

50      continue

c       starting the loop in time for the Mac Cormack method

        do 100 n = 1,nmax
```

```
cc          The loop below is the first Macormic step ( predictor)

                      y = 0.0+dy+dy/2.

                      do 200 j = 2,jmax-1

c           time derivative, du1 arrays is calculated based on the initial values at time t

                                          do 210 i = 2,imax-1
                                                    du1(i,j,1) = -( f(i+1,j,1)-f(i,j,1) )/(dx)
   &                                                                  - ( g(i,j+1,1)-g(i,j,1) )/(dy)
   &                                                                  -jj(i,j,1)
                                                    du1(i,j,2) = -( f(i+1,j,2)-f(i,j,2) )/(dx)
   &                                                                  - ( g(i,j+1,2)-g(i,j,2) )/(dy)
   &                                                                  -jj(i,j,2)
                                                    du1(i,j,3) = -( f(i+1,j,3)-f(i,j,3) )/(dx)
   &                                                                  - ( g(i,j+1,3)-g(i,j,3) )/(dy)
   &                                                                  -jj(i,j,3)
                                                    du1(i,j,4) = -( f(i+1,j,4)-f(i,j,4) )/(dx)
   &                                                                  - ( g(i,j+1,4)-g(i,j,4) )/(dy)
   &                                                                  -jj(i,j,4)

c           based on the du1 array, a new predicted value for u is calculated
c           this is the predicted value of u at t+1

                                                    up1(i,j,1) = (1.0/4.)*(u(i+1,j,1)+ u(i-1,j,1)+
   &                                                                  u(i,j+1,1) + u(i,j-1,1))
   &                                                                  + du1(i,j,1)*(dt)
                                                    up1(i,j,2) = (1.0/4.)*(u(i+1,j,2)+ u(i-1,j,2)+
   &                                                                  u(i,j+1,2) + u(i,j-1,2))
   &                                                                  + du1(i,j,2)*(dt)

                                                    up1(i,j,3) = (1.0/4.)*(u(i+1,j,3)+ u(i-1,j,3)+
   &                                                                  u(i,j+1,3) + u(i,j-1,3))
   &                                                                  + du1(i,j,3)*(dt)
                                                    up1(i,j,4) = (1.0/4.)*(u(i+1,j,4)+ u(i-1,j,4)+
   &                                                                  u(i,j+1,4) + u(i,j-1,4))
   &                                                                  + du1(i,j,4)*(dt)

c           a predicted values for ro, us, vs, t are calculted at t+1 based on u at t+1

                                                    rop1(i,j) =  up1(i,j,1)
                                                    usp1(i,j) =  up1(i,j,2) / up1(i,j,1)
                                                    vsp1(i,j) =  up1(i,j,3) / up1(i,j,1)
                                                    ep1(i,j)  =( -(1./2.)*(usp1(i,j)**2.+ vsp1(i,j)**2.)
   &                                                                  ) + (up1(i,j,4)/rop1(i,j))
c            need subroutine

            rorp = rop1(i,j)*roo
            vvp = 1.0/rorp
            eep = ep1(i,j) * (to*R)

            ttp=tinitp
            call rootfind(eep,h1,h1r,ttp,a,b,vvp,t1,acp,bcp,ccp,dcp,tol)
```

```
          tp1(i,j) = ttp/to
          tinitp=ttp

c                                       tp1(i,j) = ep1(i,j)*(R/cv)


c                                       pp1(i,j) = rop1(i,j)*tp1(i,j)
          pp1(i,j) =( ((R*tp1(i,j)*to)/( 1.0/(rop1(i,j)*roo)-b))-
   &                (a/((tp1(i,j)*to)**(1.0/2.0)
   &                * (1.0/(rop1(i,j)*roo))*( 1.0/(rop1(i,j)*roo)+b))) )/po


c         predicted f, g, and jj arrays based on predicted u at t+1 are calculated

                                        fp1(i,j,1) = rop1(i,j) * usp1(i,j)
                                        fp1(i,j,2) = rop1(i,j) * usp1(i,j) * usp1(i,j)
   &                                                   + pp1(i,j)*Zo
                                        fp1(i,j,3) = rop1(i,j) * usp1(i,j) * vsp1(i,j)
                                        fp1(i,j,4) = ((( ( (1./2.)*(usp1(i,j)**2.+
   &                                              vsp1(i,j)**2.) )
   &                                              + ep1(i,j) )*rop1(i,j))+pp1(i,j)*Zo)*
   &                                              usp1(i,j)

                                        gp1(i,j,1) = rop1(i,j) * vsp1(i,j)
                                        gp1(i,j,2) = rop1(i,j) * usp1(i,j) * vsp1(i,j)
                                        gp1(i,j,3) = rop1(i,j) * vsp1(i,j) * vsp1(i,j)
   &                                               +pp1(i,j)*Zo
                                        gp1(i,j,4) =  ((( ( (1./2.)*(usp1(i,j)**2.+
   &                                              vsp1(i,j)**2.) )
   &                                              + ep1(i,j) )*rop1(i,j))+pp1(i,j)*Zo)*
   &                                              vsp1(i,j)

                                        jjp1(i,j,1)=eps*( (1.0/y)*rop1(i,j)*vsp1(i,j) )
                                        jjp1(i,j,2)=eps*( (1.0/y)*rop1(i,j)*usp1(i,j)*vsp1(i,j) )
                                        jjp1(i,j,3)=eps*( (1.0/y)*rop1(i,j)*vsp1(i,j)*vsp1(i,j) )
                                        jjp1(i,j,4)= eps*( (1./y)*((( ( (1./2.)*(usp1(i,j)**2.+
   &                                              vsp1(i,j)**2.) )
   &                                              + ep1(i,j) )*rop1(i,j))+pp1(i,j)*Zo)*
   &                                              vsp1(i,j))

c              write(6,*) 'test du', i,j,fp1(i+1,j,1),fp1(i-1,j,1)

210                                 continue

                    y =  y + dy

200            continue

c         between the two loops, predictor and corrector, impose the BC on the boundaries and calculate
c         the fp and gp at the boundaries.

c
c         left side
c
          y=0.0+dy/2.
```

```
         do 4100 j=1,13
                 ro(1,j) = roj/roo
                 t(1,j) = tj/to
                 us(1,j) =  uj/ ((to*R)**(1./2.))
                 vs(1,j) = 0.0
c         need subroutine
         p(1,j) =( ((R*t(1,j)*to)/( 1.0/(ro(1,j)*roo)-b))-
   &                     (a/( (t(1,j)*to)**(1.0/2.0)
   &                     * (1.0/(ro(1,j)*roo))*( 1.0/(ro(1,j)*roo)+b))) )/po

c        p(1,j) = ro(1,j)*t(1,j)
c        e(1,j) = t(1,j)*( cv/R  )
                 e(1,j) =( (h1+h1r-((R*t(1,j)*to)*(( (3*a)/
   &                     (2*b*R*(t(1,j)*to)**(3.0/2.0))
   &                     *(log(1+(b/(1./(ro(1,j)*roo))))))
   &                     -(p(1,j)*po*(1./(ro(1,j)*roo)))/(R*(t(1,j)*to))+1))
   &                  +(R)*
   &                  ((acp*((t(1,j)*to)-t1))+
   &                     ((bcp/2.0)*(((t(1,j)*to)**2.0)-(t1**2.0)))+
   &                     ((ccp/3.0)*(((t(1,j)*to)**3.0)-(t1**3.0)))+
   &                     ((dcp/4.0)*(((t(1,j)*to)**4.0)-(t1**4.0))))-
   &                     ( (p(1,j)*po)/(ro(1,j)*roo) ) ) )/ (to*R)

                 up1(1,j,1) =  ro(1,j)
                 up1(1,j,2) =  ro(1,j) * us(1,j)
                 up1(1,j,3) =  ro(1,j) * vs(1,j)
                 up1(1,j,4) =  ( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                 + e(1,j) )*ro(1,j)


                 fp1(1,j,1) = ro(1,j) * us(1,j)
                 fp1(1,j,2) = ro(1,j) * us(1,j) * us(1,j)
   &                                                      + p(1,j)*Zo
                 fp1(1,j,3) = ro(1,j) * us(1,j) * vs(1,j)
                 fp1(1,j,4) =  ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                 + e(1,j) )*ro(1,j))+p(1,j)*Zo)*us(1,j)

                 gp1(1,j,1) = ro(1,j) * vs(1,j)
                 gp1(1,j,2) = ro(1,j) * us(1,j) * vs(1,j)
                 gp1(1,j,3) = ro(1,j) * vs(1,j) * vs(1,j)
   &                                                  +p(1,j)*Zo
                 gp1(1,j,4) =  ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                 + e(1,j) )*ro(1,j))+p(1,j)*Zo)*vs(1,j)

                 jjp1(1,j,1) = eps * ( (1.0/y)*ro(1,j)*vs(1,j) )
                 jjp1(1,j,2) = eps * ( (1.0/y)*ro(1,j)*us(1,j)*vs(1,j) )
                 jjp1(1,j,3) = eps * ( (1.0/y)*ro(1,j)*vs(1,j)*vs(1,j) )
                 jjp1(1,j,4) = eps*( (1./y)*((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                 + e(1,j) )*ro(1,j))+p(1,j)*Zo)*vs(1,j) )

         y=y+dy

4100                             continue

         do 4200 j=14,jmax
```

```
c                   us(1,j) = -usp1(3,j)
c                   vs(1,j) = vsp1(3,j)
c                   ro(1,j) = rop1(3,j)
c                   t(1,j) = tp1(3,j)
                    us(1,j) =0.0
                    vs(1,j) = 0.0
                    ro(1,j) = roamb/roo
                    t(1,j) = tamb/to
          p(1,j) =( ((R*t(1,j)*to)/( 1.0/(ro(1,j)*roo)-b))-
   &                                    (a/((t(1,j)*to)**(1.0/2.0)
   &                          * (1.0/(ro(1,j)*roo))*( 1.0/(ro(1,j)*roo)+b))) )/po

c         p(1,j) = ro(1,j)*t(1,j)
c         e(1,j) = t(1,j)*( cv/R  )
                    e(1,j) =( (h1+h1r-((R*t(1,j)*to)*(( (3*a)/
   &                          (2*b*R*(t(1,j)*to)**(3.0/2.0))
   &                          *(log(1+(b/(1./(ro(1,j)*roo))))))
   &                          -(p(1,j)*po*(1./(ro(1,j)*roo)))/(R*(t(1,j)*to))+1))
   &                   +(R)*
   &                   ((acp*((t(1,j)*to)-t1))+
   &                          ((bcp/2.0)*(((t(1,j)*to)**2.0)-(t1**2.0)))+
   &                          ((ccp/3.0)*(((t(1,j)*to)**3.0)-(t1**3.0)))+
   &                          ((dcp/4.0)*(((t(1,j)*to)**4.0)-(t1**4.0))))-
   &                          ( (p(1,j)*po)/(ro(1,j)*roo) ) ) )/(to*R)

                    up1(1,j,1) =  ro(1,j)
                    up1(1,j,2) =  ro(1,j) * us(1,j)
                    up1(1,j,3) =  ro(1,j) * vs(1,j)
                    up1(1,j,4) =  ( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                        + e(1,j) )*ro(1,j)


                    fp1(1,j,1) = ro(1,j) * us(1,j)
                    fp1(1,j,2) = ro(1,j) * us(1,j) * us(1,j)
   &                                                                  + p(1,j)*Zo
                    fp1(1,j,3) = ro(1,j) * us(1,j) * vs(1,j)
                    fp1(1,j,4) =  ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                        + e(1,j) )*ro(1,j))+p(1,j)*Zo)*us(1,j)

                    gp1(1,j,1) = ro(1,j) * vs(1,j)
                    gp1(1,j,2) = ro(1,j) * us(1,j) * vs(1,j)
                    gp1(1,j,3) = ro(1,j) * vs(1,j) * vs(1,j)
   &                                                         +p(1,j)*Zo
                    gp1(1,j,4) =  ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                        + e(1,j) )*ro(1,j))+p(1,j)*Zo)*vs(1,j)

                    jjp1(1,j,1) = eps * ( (1.0/y)*ro(1,j)*vs(1,j) )
                    jjp1(1,j,2) = eps * ( (1.0/y)*ro(1,j)*us(1,j)*vs(1,j) )
                    jjp1(1,j,3) = eps * ( (1.0/y)*ro(1,j)*vs(1,j)*vs(1,j) )
                    jjp1(1,j,4) = eps*( (1./y)*((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
   &                                                        + e(1,j) )*ro(1,j))+p(1,j)*Zo)*vs(1,j) )

          y=y+dy

4200                                    continue
```

```
c
c         right side
c

          y=0.+dy+dy/2.

          do 2600 j=2,jmax

                    ro(imax,j) = rop1(imax-2,j)
                    us(imax,j) = -usp1(imax-2,j)

c                   us(imax-1,j) = 0.

                    vs(imax,j) = vsp1(imax-2,j)
                    t(imax,j) = tp1(imax-2,j)
          p(imax,j) =  ( ((R*t(imax,j)*to)/( 1.0/(ro(imax,j)*roo)-b))
   &                          -(a/( (t(imax,j)*to)**(1.0/2.0)
   &                * (1.0/(ro(imax,j)*roo))*( 1.0/(ro(imax,j)*roo)+b))) )/po

c         p(imax,j) = ro(imax,j)*t(imax,j)
c         e(imax,j) = t(imax,j)*(cv/R)

                    e(imax,j) =( (h1+h1r-((R*t(imax,j)*to)*(( (3*a)/
   &                          (2*b*R*(t(imax,j)*to)**(3.0/2.0))
   &                          *(log(1+(b/(1./(ro(imax,j)*roo))))))
   &                          -(p(imax,j)*po*(1./(ro(imax,j)*roo)))/
   &                   (R*(t(imax,j)*to))+1))
   &                   +(R)*
   &                   ((acp*((t(imax,j)*to)-t1))+
   &                          ((bcp/2.0)*(((t(imax,j)*to)**2.0)-(t1**2.0)))+
   &                          ((ccp/3.0)*(((t(imax,j)*to)**3.0)-(t1**3.0)))+
   &                          ((dcp/4.0)*(((t(imax,j)*to)**4.0)-(t1**4.0))))-
   &                          ( (p(imax,j)*po)/(ro(imax,j)*roo) ) ) )/(to*R)



c                   have changed the BC at the right side to reflective BC

c                   ro(imax,j) = 2*rop1(imax-1,j) - rop1(imax-2,j)
c                   us(imax,j) = 2*usp1(imax-1,j) - usp1(imax-2,j)
c                   vs(imax,j) = 2*vsp1(imax-1,j) - vsp1(imax-2,j)
c                   t(imax,j) = 2*tp1(imax-1,j) - tp1(imax-2,j)



                    up1(imax,j,1) =  ro(imax,j)
                    up1(imax,j,2) =  ro(imax,j) * us(imax,j)
                    up1(imax,j,3) =  ro(imax,j) * vs(imax,j)
                    up1(imax,j,4) =   ( ( (1./2.)*(us(imax,j)**2.+ vs(imax,j)**2.) )
   &                                              + e(imax,j) )*ro(imax,j)


                    fp1(imax,j,1) = ro(imax,j) * us(imax,j)
                    fp1(imax,j,2) = ro(imax,j) * us(imax,j) * us(imax,j)
   &                                             + p(imax,j)*Zo
                    fp1(imax,j,3) = ro(imax,j) * us(imax,j) * vs(imax,j)
```

```
                fp1(imax,j,4) =  ((( ( (1./2.)*(us(imax,j)**2.+ vs(imax,j)**2.) )
   &                                    + e(imax,j) )*ro(imax,j))
   &                               +p(imax,j)*Zo)*us(imax,j)


                gp1(imax,j,1) = ro(imax,j) * vs(imax,j)
                gp1(imax,j,2) = ro(imax,j) * us(imax,j) * vs(imax,j)
                gp1(imax,j,3) = ro(imax,j) * vs(imax,j) * vs(imax,j)
   &                                                +p(imax,j)*Zo
                gp1(imax,j,4) =   ((( ( (1./2.)*(us(imax,j)**2.+ vs(imax,j)**2.) )
   &                                    + e(imax,j) )*ro(imax,j))
   &                               +p(imax,j)*Zo)*vs(imax,j)


                jjp1(imax,j,1) = eps * ( (1.0/y)*ro(imax,j)*vs(imax,j) )
                jjp1(imax,j,2) = eps * ( (1.0/y)*ro(imax,j)*us(imax,j)*vs(imax,j) )
                jjp1(imax,j,3) = eps * ( (1.0/y)*ro(imax,j)*vs(imax,j)*vs(imax,j) )
                jjp1(imax,j,4) = eps*( (1./y)*  ((( ( (1./2.)*(us(imax,j)**2.+
   &                                   vs(imax,j)**2.) )
   &                                    + e(imax,j) )*ro(imax,j))
   &                               +p(imax,j)*Zo)*vs(imax,j)   )



        y=y+dy

2600    continue

c
c       top
c       fix top BC , use interpolation

        y=dy*float(jmax) +dy/2.0

        do 4300 i=1,imax

                us(i,jmax) =0.0
c       vs(i,jmax) =  (1/.02366)*(1./1600.)
c       vs(i,jmax) =  ( roj/roamb )*(1./1600.)
c               vs(i,jmax) = 0.0
                vs(i,jmax) = 2*vs(i,jmax-1) - vs(i,jmax-2)

                ro(i,jmax) = roamb/roo
                t(i,jmax) = tamb/to
        p(i,jmax) = ( ((R*t(i,jmax)*to)/( 1.0/(ro(i,jmax)*roo)-b))-
   &                        (a/( (t(i,jmax)*to)**(1.0/2.0)
   &                        * (1.0/(ro(i,jmax)*roo))*( 1.0/(ro(i,jmax)*roo)+b))) )/po

c       p(i,jmax) = ro(i,jmax)*t(i,jmax)
c       e(i,jmax) = t(i,jmax)*(cv/R  )
                e(i,jmax) =( (h1+h1r-((R*t(i,jmax)*to)*(( (3*a)/
   &                        (2*b*R*(t(i,jmax)*to)**(3.0/2.0))
   &                        *(log(1+(b/(1./(ro(i,jmax)*roo))))))
   &                        -(p(i,jmax)*po*(1./(ro(i,jmax)*roo)))/
   &                        (R*(t(i,jmax)*to))+1))
   &                  +(R)*
```

```
     &                     ((acp*((t(i,jmax)*to)-t1))+
     &                              ((bcp/2.0)*(((t(i,jmax)*to)**2.0)-(t1**2.0)))+
     &                              ((ccp/3.0)*(((t(i,jmax)*to)**3.0)-(t1**3.0)))+
     &                              ((dcp/4.0)*(((t(i,jmax)*to)**4.0)-(t1**4.0))))-
     &                              ( (p(i,jmax)*po)/(ro(i,jmax)*roo) ) ) )/(to*R)

                  up1(i,jmax,1) =  ro(i,jmax)
                  up1(i,jmax,2) =  ro(i,jmax) * us(i,jmax)
                  up1(i,jmax,3) =  ro(i,jmax) * vs(i,jmax)
                  up1(i,jmax,4) =  ( ( (1./2.)*(us(i,jmax)**2.+ vs(i,jmax)**2.) )
     &                                                   + e(i,jmax) )*ro(i,jmax)

                  fp1(i,jmax,1) = ro(i,jmax) * us(i,jmax)
                  fp1(i,jmax,2) = ro(i,jmax) * us(i,jmax) * us(i,jmax)
     &                                                                       + p(i,jmax)*Zo
                  fp1(i,jmax,3) = ro(i,jmax) * us(i,jmax) * vs(i,jmax)
                  fp1(i,jmax,4) = ((( ( (1./2.)*(us(i,jmax)**2.+ vs(i,jmax)**2.) )
     &                                                   + e(i,jmax) )*ro(i,jmax))
     &                                          +p(i,jmax)*Zo)*us(i,jmax)


                  gp1(i,jmax,1) = ro(i,jmax) * vs(i,jmax)
                  gp1(i,jmax,2) = ro(i,jmax) * us(i,jmax) * vs(i,jmax)
                  gp1(i,jmax,3) = ro(i,jmax) * vs(i,jmax) * vs(i,jmax)
     &                                                  +p(i,jmax)*Zo
                  gp1(i,jmax,4) = ((( ( (1./2.)*(us(i,jmax)**2.+ vs(i,jmax)**2.) )
     &                                                   + e(i,jmax) )*ro(i,jmax))
     &                                          +p(i,jmax)*Zo)*vs(i,jmax)


                  jjp1(i,jmax,1) = eps * ( (1.0/y)*ro(i,jmax)*vs(i,jmax) )
                  jjp1(i,jmax,2) = eps * ( (1.0/y)*ro(i,jmax)*us(i,jmax)*vs(i,jmax) )
                  jjp1(i,jmax,3) = eps * ( (1.0/y)*ro(i,jmax)*vs(i,jmax)*vs(i,jmax) )
                  jjp1(i,jmax,4) = eps*( (1./y)*((( ( (1./2.)*(us(i,jmax)**2.+
     &                                         vs(i,jmax)**2.) )
     &                                                   + e(i,jmax) )*ro(i,jmax))
     &                                          +p(i,jmax)*Zo)*vs(i,jmax)   )

4300      continue

c
c         axis of symmetry
c

          y=0.0+dy/2.

          do 4400 i=1,imax

                  us(i,1) = usp1(i,3)
c                 write(6,*) 'vs at i=2',vs(i,2)
c         12/14: I will remove the v=0 condition on axis, since i applied
c         v at ghost row ( row 1) = v at row above axis( row 3)

c                 vs(i,2) = 0.0
c                 write(6,*) 'vs at i=2',vs(i,2)
                  vs(i,1) = - vsp1(i,3)
```

```
                ro(i,1) = rop1(i,3)
                t(i,1) = tp1(i,3)
          p(i,1) = ( ((R*t(i,1)*to)/( 1.0/(ro(i,1)*roo)-b))-
     &                    (a/((t(i,1)*to)**(1.0/2.0)
     &                    * (1.0/(ro(i,1)*roo))*( 1.0/(ro(i,1)*roo)+b))) )/po

c         p(i,1) = ro(i,1)*t(i,1)
c         e(i,1) = t(i,1)*( cv/R )

                e(i,1) =( (h1+h1r-((R*t(i,1)*to)*(( (3*a)/
     &                    (2*b*R*(t(i,1)*to)**(3.0/2.0))
     &                    *(log(1+(b/(1./(ro(i,1)*roo))))))
     &                    -(p(i,1)*po*(1./(ro(i,1)*roo)))/(R*(t(i,1)*to))+1))
     &                 +(R)*
     &                 ((acp*((t(i,1)*to)-t1))+
     &                    ((bcp/2.0)*(((t(i,1)*to)**2.0)-(t1**2.0)))+
     &                    ((ccp/3.0)*(((t(i,1)*to)**3.0)-(t1**3.0)))+
     &                    ((dcp/4.0)*(((t(i,1)*to)**4.0)-(t1**4.0))))-
     &                    ( (p(i,1)*po)/(ro(i,1)*roo) ) ) )/(to*R)

                up1(i,1,1) =  ro(i,1)
                up1(i,1,2) =  ro(i,1) * us(i,1)
                up1(i,1,3) =  ro(i,1) * vs(i,1)
                up1(i,1,4) =   ( ( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
     &                                         + e(i,1) )*ro(i,1)
c         write(6,*)'test',up1(i,j,1)
                fp1(i,1,1) = ro(i,1) * us(i,1)
                fp1(i,1,2) = ro(i,1) * us(i,1) * us(i,1)
     &                                                    + p(i,1)*Zo
                fp1(i,1,3) = ro(i,1) * us(i,1) * vs(i,1)
                fp1(i,1,4) = ((( ( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
     &                                         + e(i,1) )*ro(i,1))
     &                                    +p(i,1)*Zo)*us(i,1)


                gp1(i,1,1) = ro(i,1) * vs(i,1)
                gp1(i,1,2) = ro(i,1) * us(i,1) * vs(i,1)
                gp1(i,1,3) = ro(i,1) * vs(i,1) * vs(i,1)
     &                                                    +p(i,1)*Zo
                gp1(i,1,4) = ((( ( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
     &                                         + e(i,1) )*ro(i,1))
     &                                    +p(i,1)*Zo)*vs(i,1)


                jjp1(i,1,1) = eps * ( (1.0/y)*ro(i,1)*vs(i,1) )
                jjp1(i,1,2) = eps * ( (1.0/y)*ro(i,1)*us(i,1)*vs(i,1) )
                jjp1(i,1,3) = eps * ( (1.0/y)*ro(i,1)*vs(i,1)*vs(i,1) )
                jjp1(i,1,4) = eps*( ((( ( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
     &                                         + e(i,1) )*ro(i,1))
     &                                    +p(i,1)*Zo)*vs(i,1)  )

4400      continue


cc        The loop below is the second Macormic step ( corrector)
```

```
          y=0.+dy+dy/2.

                do 220 j=2,jmax-1

                      do 230 i = 2, imax-1

c         we calculate a predicted value of the time derivative at time t+1
c         of arrau U by substituting the predicted values of f and g

                            du2p1(i,j,1) = -( fp1(i,j,1)-fp1(i-1,j,1) )/(dx)
     &                                  - ( gp1(i,j,1)-gp1(i,j-1,1) )/(dy)
     &                                  -jjp1(i,j,1)
                            du2p1(i,j,2) = -( fp1(i,j,2)-fp1(i-1,j,2) )/(dx)
     &                                  - ( gp1(i,j,2)-gp1(i,j-1,2) )/(dy)
     &                                  -jjp1(i,j,2)
                            du2p1(i,j,3) = -( fp1(i,j,3)-fp1(i-1,j,3) )/(dx)
     &                                  - ( gp1(i,j,3)-gp1(i,j-1,3) )/(dy)
     &                                  -jjp1(i,j,3)
                            du2p1(i,j,4) = -( fp1(i,j,4)-fp1(i-1,j,4) )/(dx)
     &                                  - ( gp1(i,j,4)-gp1(i,j-1,4) )/(dy)
     &                                  -jjp1(i,j,4)

c         an average value of the time derivative of U is calculated from the
c         mean of du at t and du at t+1

                            du(i,j,1) = du2p1(i,j,1)
                            du(i,j,2) = du2p1(i,j,2)
                            du(i,j,3) = du2p1(i,j,3)
                            du(i,j,4) = du2p1(i,j,4)
c         write(6,81) i,j,fp1(i+1,j,1),fp1(i-1,j,1),gp1(i,j+1,1),du(i,j-1,1)

c         calculate a corrected array U at t+1 based on the corrected time derivative du

                            u2p1(i,j,1) = (1./2.)*( u(i,j,1) + up1(i,j,1) + dt*du(i,j,1) )
c                           u(i,j,1) + du(i,j,1)*dt
c    &                                  + (phi/3.)*( (1.0/4.)*(u(i+1,j,1)+ u(i-1,j,1)+
c    &                                  u(i,j+1,1) + u(i,j-1,1)) - u(i,j,1) )
c     &                     + cx* abs( rop1(i+1,j)*tp1(i+1,j)*gam-
c     &                     2.0*rop1(i,j)*tp1(i,j)*gam +
c     &                     rop1(i-1,j)*tp1(i-1,j)*gam )
c     &                     * ( u2(i+1,j,1) - 2. * u2(i,j,1) + u2(i-1,j,1) )
c     &                     / ( rop1(i+1,j)*tp1(i+1,j)*gam +
c     &                     2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i-1,j)*tp1(i-1,j)*gam  )
c     &                     / ( ro(i+1,j)*t(i+1,j)*gam +
c     &                     2.0*ro(i,j)*t(i,j)*gam + ro(i-1,j)*t(i-1,j)*gam  )
c     &                     + cy* abs( rop1(i,j+1)*tp1(i,j+1)*gam-
c     &                     2.0*rop1(i,j)*tp1(i,j)*gam +
c     &                     rop1(i,j-1)*tp1(i,j-1)*gam )
c     &                     * ( u2(i,j+1,1) - 2. * u2(i,j,1) + u2(i,j-1,1) )
c     &                     / ( rop1(i,j+1)*tp1(i,j+1)*gam +
c     &                     2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i,j-1)*tp1(i,j-1)*gam  )
c     &                     / ( ro(i,j+1)*t(i,j+1)*gam +
c     &                     2.0*ro(i,j)*t(i,j)*gam + ro(i,j-1)*t(i,j-1)*gam  )

                            u2p1(i,j,2) = (1./2.)*( u(i,j,2) + up1(i,j,2) + dt*du(i,j,2) )
```

```
c                              u(i,j,2) + du(i,j,2)*dt
c    &                                        + (phi/3.)*( (1.0/4.)*(u(i+1,j,2)+ u(i-1,j,2)+
c    &                                        u(i,j+1,2) + u(i,j-1,2)) - u(i,j,2) )


c    &                         + cx* abs( rop1(i+1,j)*tp1(i+1,j)*gam-
c    &                         2.0*rop1(i,j)*tp1(i,j)*gam  + rop1(i-1,j)*tp1(i-1,j)*gam )
c    &                         * ( u2(i+1,j,2) - 2. * u2(i,j,2) + u2(i-1,j,2) )
c    &                         / ( rop1(i+1,j)*tp1(i+1,j)*gam +2.0*rop1(i,j)*tp1(i,j)*gam
c    &                         + rop1(i-1,j)*tp1(i-1,j)*gam)
cc    &                        / ( ro(i+1,j)*t(i+1,j)*gam +2.0*ro(i,j)*t(i,j)*gam
cc    &                        + ro(i-1,j)*t(i-1,j)*gam)
c    &                         + cy* abs( rop1(i,j+1)*tp1(i,j+1)*gam-
c    &                         2.0*rop1(i,j)*tp1(i,j)*gam +
c    &                         rop1(i,j-1)*tp1(i,j-1)*gam )
c    &                         * ( u2(i,j+1,2) - 2. * u2(i,j,2) + u2(i,j-1,2) )
c    &                         / ( rop1(i,j+1)*tp1(i,j+1)*gam +
c    &                         2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i,j-1)*tp1(i,j-1)*gam  )
c    &                         / ( ro(i,j+1)*t(i,j+1)*gam +
c    &                         2.0*ro(i,j)*t(i,j)*gam + ro(i,j-1)*t(i,j-1)*gam  )

                               u2p1(i,j,3) =  (1./2.)*( u(i,j,3) + up1(i,j,3) + dt*du(i,j,3))
c                              u(i,j,3) + du(i,j,3)*dt
c    &                                        + (phi/3.)*( (1.0/4.)*(u(i+1,j,3)+ u(i-1,j,3)+
c    &                                        u(i,j+1,3) + u(i,j-1,3)) - u(i,j,3) )

c    &                         + cx* abs( rop1(i+1,j)*tp1(i+1,j)*gam-
c    &                         2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i-1,j)*tp1(i-1,j)*gam )
c    &                         * ( u2(i+1,j,3) - 2. * u2(i,j,3) + u2(i-1,j,3) )
c    &                         / ( rop1(i+1,j)*tp1(i+1,j)*gam
cc    &                        +2.0*rop1(i,j)*tp1(i,j)*gam
cc    &                        + rop1(i-1,j)*tp1(i-1,j)*gam )
c    &                         / ( ro(i+1,j)*t(i+1,j)*gam
c    &                         +2.0*ro(i,j)*t(i,j)*gam
c    &                         + ro(i-1,j)*t(i-1,j)*gam )
c    &                    + cy* abs( rop1(i,j+1)*tp1(i,j+1)*gam-
c    &                         2.0*rop1(i,j)*tp1(i,j)*gam +
c    &                         rop1(i,j-1)*tp1(i,j-1)*gam )
c    &                         * ( u2(i,j+1,3) - 2. * u2(i,j,3) + u2(i,j-1,3) )
cc    &                        / ( rop1(i,j+1)*tp1(i,j+1)*gam +
cc    &                        2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i,j-1)*tp1(i,j-1)*gam  )
c    &                         / ( ro(i,j+1)*t(i,j+1)*gam +
c    &                         2.0*ro(i,j)*t(i,j)*gam + ro(i,j-1)*t(i,j-1)*gam  )

                               u2p1(i,j,4) =(1./2.)*( u(i,j,4) + up1(i,j,4) + dt*du(i,j,4) )
c                               u(i,j,4) + du(i,j,4)*dt
c    &                                        + (phi/3.)*( (1.0/4.)*(u(i+1,j,4)+ u(i-1,j,4)+
c    &                                        u(i,j+1,4) + u(i,j-1,4)) - u(i,j,4) )

c    &                         + cx* abs( rop1(i+1,j)*tp1(i+1,j)*gam-
c    &                         2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i-1,j)*tp1(i-1,j)*gam )
c    &                         * ( u2(i+1,j,4) - 2. * u2(i,j,4) + u2(i-1,j,4) )
cc    &                        / ( rop1(i+1,j)*tp1(i+1,j)*gam +
cc    &                        2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i-1,j)*tp1(i-1,j)*gam )
c    &                         / ( ro(i+1,j)*t(i+1,j)*gam +
c    &                         2.0*ro(i,j)*t(i,j)*gam + ro(i-1,j)*t(i-1,j)*gam )
```

```
c    &                                  + cy* abs( rop1(i,j+1)*tp1(i,j+1)*gam-
c    &                                  2.0*rop1(i,j)*tp1(i,j)*gam +
c    &                                  rop1(i,j-1)*tp1(i,j-1)*gam )
c    &                                  * ( u2(i,j+1,4) - 2. * u2(i,j,4) + u2(i,j-1,4) )
cc    &                                 / ( rop1(i,j+1)*tp1(i,j+1)*gam +
cc    &                                 2.0*rop1(i,j)*tp1(i,j)*gam + rop1(i,j-1)*tp1(i,j-1)*gam  )
c    &                                  / ( ro(i,j+1)*t(i,j+1)*gam +
c    &                                  2.0*ro(i,j)*t(i,j)*gam + ro(i,j-1)*t(i,j-1)*gam  )

c          calculate corrected ro, us, vs, and t at time t+1
c          calcualte corrected f,g and jj arrays at t+1

                                                 ro2p1(i,j) =  u2p1(i,j,1)
                                                 us2p1(i,j) =  u2p1(i,j,2) / u2p1(i,j,1)
                                                 vs2p1(i,j) =  u2p1(i,j,3) / u2p1(i,j,1)
                                                 e2p1(i,j)  =            ( -(1./2.)*(us2p1(i,j)**2.+
   &                                                                           vs2p1(i,j)**2.)
   &                                                                     ) + (u2p1(i,j,4)/ro2p1(i,j))
c          write(6,*)n,i,j,ro2p1(i,j)
           rorc = ro2p1(i,j)*roo
           vvc = 1.0/rorc
           eec = e2p1(i,j) * (to*R)

           ttc=tinitc
c          write(6,*)'before sub',          n,i,j,eec
           call rootfind(eec,h1,h1r,ttc,a,b,vvc,t1,acp,bcp,ccp,dcp,tol)
c          write(6,*)'before sub',          ee,vv,ror,h1r

           t2p1(i,j) = ttc/to
           tinitc=ttc
c          write(6,*) n,i,ttc
c          end



c                                                t2p1(i,j) = e2p1(i,j)*( R/cv  )

           p2p1(i,j) =(  ((R*t2p1(i,j)*to)/( 1.0/(ro2p1(i,j)*roo)-b))-
   &                                  (a/((t2p1(i,j)*to)**(1.0/2.0)
   &                                  * (1.0/(ro2p1(i,j)*roo))*
   &                             ( 1.0/(ro2p1(i,j)*roo)+b))) )/po

c                                                p2p1(i,j) = ro2p1(i,j)*t2p1(i,j)

c                                       ro2p1(i,j) = u2p1(i,j,1)
c                                       us2p1(i,j) = u2p1(i,j,2) / u2p1(i,j,1)
c                                       vs2p1(i,j) = u2p1(i,j,3) / u2p1(i,j,1)
c                                       t2p1(i,j) = (gam-1.) * ( (u2p1(i,j,4)/ro2p1(i,j)) -
c    &                                           ((1./2.)*(us2p1(i,j)**2.+ vs2p1(i,j)**2.)) )

                                                 f2p1(i,j,1) = ro2p1(i,j) * us2p1(i,j)
                                                 f2p1(i,j,2) = ro2p1(i,j) * us2p1(i,j) * us2p1(i,j)
   &                                                                             + p2p1(i,j)*Zo
                                                 f2p1(i,j,3) = ro2p1(i,j) * us2p1(i,j) * vs2p1(i,j)
                                                 f2p1(i,j,4) = ((( ( (1./2.)*(us2p1(i,j)**2.+
   &                                                                     vs2p1(i,j)**2.) )
```

```
   &                                                  + e2p1(i,j) )*ro2p1(i,j))
   &                                                  +p2p1(i,j)*Zo)*us2p1(i,j)

                                            g2p1(i,j,1) = ro2p1(i,j) * vs2p1(i,j)
                                            g2p1(i,j,2) = ro2p1(i,j) * us2p1(i,j) * vs2p1(i,j)
                                            g2p1(i,j,3) = ro2p1(i,j) * vs2p1(i,j) * vs2p1(i,j)
   &                                                       +p2p1(i,j)*Zo
                                            g2p1(i,j,4) =((( ( (1./2.)*(us2p1(i,j)**2.+
   &                                                  vs2p1(i,j)**2.) )
   &                                                  + e2p1(i,j) )*ro2p1(i,j))
   &                                                  +p2p1(i,j)*Zo)*vs2p1(i,j)

                                            jj2p1(i,j,1)=eps*( (1.0/y)*ro2p1(i,j)*vs2p1(i,j) )
                                            jj2p1(i,j,2)=eps*( (1.0/y)*ro2p1(i,j)*us2p1(i,j)*
   &                                        vs2p1(i,j) )
                                            jj2p1(i,j,3)=eps*( (1.0/y)*ro2p1(i,j)*vs2p1(i,j)*
   &                                        vs2p1(i,j) )
                                            jj2p1(i,j,4)= eps*( (1./y)*((( ( (1./2.)*(us2p1(i,j)**2.+
   &                                                  vs2p1(i,j)**2.) )
   &                                                  + e2p1(i,j) )*ro2p1(i,j))
   &                                                  +p2p1(i,j)*Zo)*vs2p1(i,j))

230                            continue

          y=y+dy

220                   continue

c         this is the end of the loop, should reset all values calculated at t+1 to
c         values at t and go back to the top and repeat another time step

            do 240 i = 2, imax-1

                              do 250 j = 2,jmax-1

                                        u(i,j,1) = u2p1(i,j,1)
                                        u(i,j,2) = u2p1(i,j,2)
                                        u(i,j,3) = u2p1(i,j,3)
                                        u(i,j,4) = u2p1(i,j,4)
c                                       write(6,*) i,j,u(i,j,1),u(i,j,2),u(i,j,3),u(i,j,4)

                                        f(i,j,1) = f2p1(i,j,1)
                                        f(i,j,2) = f2p1(i,j,2)
                                        f(i,j,3) = f2p1(i,j,3)
                                        f(i,j,4) = f2p1(i,j,4)

                                        g(i,j,1) = g2p1(i,j,1)
                                        g(i,j,2) = g2p1(i,j,2)
                                        g(i,j,3) = g2p1(i,j,3)
                                        g(i,j,4) = g2p1(i,j,4)

                                        jj(i,j,1) = jj2p1(i,j,1)
                                        jj(i,j,2) = jj2p1(i,j,2)
                                        jj(i,j,3) = jj2p1(i,j,3)
                                        jj(i,j,4) = jj2p1(i,j,4)
```

```
                                ro(i,j) = ro2p1(i,j)
                                us(i,j) = us2p1(i,j)
                                vs(i,j) = vs2p1(i,j)
                                t(i,j) = t2p1(i,j)
        p(i,j)= p2p1(i,j)
        e(i,j)=e2p1(i,j)
250                     continue

240       continue


c       apply BC to the right, at i=imax for all j
c       a line fit using the 2 last two points is used.
c        need to be modified to include the last five point in a quadratic curve fit
c       BC are applied to ro, us, vs, and t, then new U,f,g and jj arrays are calculated


        y = 0.+ dy + dy/2.

        do 260 j=2,jmax

                ro(imax,j) = ro(imax-2,j)
                us(imax,j) = -us(imax-2,j)

c               us(imax-1,j) = 0.

                vs(imax,j) = vs(imax-2,j)
                t(imax,j) = t(imax-2,j)

c               ro(imax,j) = 2*ro(imax-1,j) - ro(imax-2,j)
c               us(imax,j) = 2*us(imax-1,j) - us(imax-2,j)
c               vs(imax,j) = 2*vs(imax-1,j) - vs(imax-2,j)
c               t(imax,j) = 2*t(imax-1,j) - t(imax-2,j)

        p(imax,j) = ( ((R*t(imax,j)*to)/( 1.0/(ro(imax,j)*roo)-b))-
   &                    (a/( (t(imax,j)*to)**(1.0/2.0)
   &            * (1.0/(ro(imax,j)*roo))*( 1.0/(ro(imax,j)*roo)+b))) )/po

c       p(imax,j) = ro(imax,j)*t(imax,j)
c       e(imax,j) = t(imax,j)*( cv/R )
                e(imax,j) =( (h1+h1r-((R*t(imax,j)*to)*(( (3*a)/
   &                    (2*b*R*(t(imax,j)*to)**(3.0/2.0))
   &                    *(log(1+(b/(1./(ro(imax,j)*roo))))))
   &                    -(p(imax,j)*po*(1./(ro(imax,j)*roo)))/
   &                (R*(t(imax,j)*to))+1))
   &                +(R)*
   &                ((acp*((t(imax,j)*to)-t1))+
   &                    ((bcp/2.0)*(((t(imax,j)*to)**2.0)-(t1**2.0)))+
   &                    ((ccp/3.0)*(((t(imax,j)*to)**3.0)-(t1**3.0)))+
   &                    ((dcp/4.0)*(((t(imax,j)*to)**4.0)-(t1**4.0))))-
   &                    ( (p(imax,j)*po)/(ro(imax,j)*roo) ) ) )/(to*R)



                u(imax,j,1) =  ro(imax,j)
                u(imax,j,2) =  ro(imax,j) * us(imax,j)
```

```
                  u(imax,j,3) =  ro(imax,j) * vs(imax,j)
                  u(imax,j,4) =  ( (( (1./2.)*(us(imax,j)**2.+ vs(imax,j)**2.) )
     &                                      ) + e(imax,j) )*ro(imax,j)


                  f(imax,j,1) = ro(imax,j) * us(imax,j)
                  f(imax,j,2) = ro(imax,j) * us(imax,j) * us(imax,j)
     &                                      + p(imax,j)*Zo
                  f(imax,j,3) = ro(imax,j) * us(imax,j) * vs(imax,j)
                  f(imax,j,4) =((( ( (1./2.)*(us(imax,j)**2.+
     &                                           vs(imax,j)**2.) )
     &                                           + e(imax,j) )*ro(imax,j))
     &                                           +p(imax,j)*Zo)*us(imax,j)


                  g(imax,j,1) = ro(imax,j) * vs(imax,j)
                  g(imax,j,2) = ro(imax,j) * us(imax,j) * vs(imax,j)
                  g(imax,j,3) = ro(imax,j) * vs(imax,j) * vs(imax,j)
     &                                  +p(imax,j)*Zo
                  g(imax,j,4) =  ((( ( (1./2.)*(us(imax,j)**2.+
     &                                           vs(imax,j)**2.) )
     &                                           + e(imax,j) )*ro(imax,j))
     &                                           +p(imax,j)*Zo)*vs(imax,j)

                  jj(imax,j,1) = eps * ( (1.0/y)*ro(imax,j)*vs(imax,j) )
                  jj(imax,j,2) = eps * ( (1.0/y)*ro(imax,j)*us(imax,j)*vs(imax,j) )
                  jj(imax,j,3) = eps * ( (1.0/y)*ro(imax,j)*vs(imax,j)*vs(imax,j) )
                  jj(imax,j,4) = eps*( (1./y)*((( ( (1./2.)*(us(imax,j)**2.+
     &                                           vs(imax,j)**2.) )
     &                                           + e(imax,j) )*ro(imax,j))
     &                                           +p(imax,j)*Zo)*vs(imax,j) )

            y=y+dy
260         continue

c           apply the BC to the left side of the mesh
c           from j=1 to 5 , apply sonic conditions, also p=po and t=to
c           from j=6 to 60, apply reflective BC, that is us = 0, only the velocity component
c           in the x direction

            y=0.+dy+dy/2.
            do 410 j=2,13
                  ro(1,j) = roj/roo
                  t(1,j) = tj/to
                  us(1,j) = uj/ ((R*to)**(1./2.))
                  vs(1,j) = 0.0
            p(1,j) = ( ((R*(t(1,j)*to))/( 1.0/(ro(1,j)*roo)-b))-
     &                        (a/((t(1,j)*to)**(1.0/2.0)
     &                        * (1.0/(ro(1,j)*roo))*( 1.0/(ro(1,j)*roo)+b))) )/po

c           p(1,j) = ro(1,j)*t(1,j)
c           e(1,j) = t(1,j)*( cv/R )
                  e(1,j) =( (h1+h1r-((R*t(1,j)*to)*(( (3*a)/
     &                        (2*b*R*(t(1,j)*to)**(3.0/2.0))
     &                        *(log(1+(b/(1./(ro(1,j)*roo))))))
     &                        -(p(1,j)*po*(1./(ro(1,j)*roo)))/(R*(t(1,j)*to))+1))
```

```
     &                     +(R)*
     &                     ((acp*((t(1,j)*to)-t1))+
     &                            ((bcp/2.0)*(((t(1,j)*to)**2.0)-(t1**2.0)))+
     &                            ((ccp/3.0)*(((t(1,j)*to)**3.0)-(t1**3.0)))+
     &                            ((dcp/4.0)*(((t(1,j)*to)**4.0)-(t1**4.0))))-
     &                            ( (p(1,j)*po)/(ro(1,j)*roo) ) ) )/(to*R)

                      u(1,j,1) =  ro(1,j)
                      u(1,j,2) =  ro(1,j) * us(1,j)
                      u(1,j,3) =  ro(1,j) * vs(1,j)
                      u(1,j,4) =  ( (( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
     &                                                        ) + e(1,j) )*ro(1,j)


                      f(1,j,1) = ro(1,j) * us(1,j)
                      f(1,j,2) = ro(1,j) * us(1,j) * us(1,j)
     &                                                        + p(1,j)*Zo
                      f(1,j,3) = ro(1,j) * us(1,j) * vs(1,j)
                      f(1,j,4) =   ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
     &                                      + e(1,j) )*ro(1,j))
     &                              +p(1,j)*Zo)*us(1,j)

                      g(1,j,1) = ro(1,j) * vs(1,j)
                      g(1,j,2) = ro(1,j) * us(1,j) * vs(1,j)
                      g(1,j,3) = ro(1,j) * vs(1,j) * vs(1,j)
     &                                                 +p(1,j)*Zo
                      g(1,j,4) =   ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
     &                                      + e(1,j) )*ro(1,j))
     &                              +p(1,j)*Zo)*vs(1,j)

                      jj(1,j,1) = eps * ( (1.0/y)*ro(1,j)*vs(1,j) )
                      jj(1,j,2) = eps * ( (1.0/y)*ro(1,j)*us(1,j)*vs(1,j) )
                      jj(1,j,3) = eps * ( (1.0/y)*ro(1,j)*vs(1,j)*vs(1,j) )
                      jj(1,j,4) = eps*( (1./y)*((( ( (1./2.)*(us(1,j)**2.+
     &                                      vs(1,j)**2.) )
     &                                      + e(1,j) )*ro(1,j))
     &                              +p(1,j)*Zo)*vs(1,j)  )


            y=y+dy

410         continue

            do 420 j=14,jmax

c                     us(1,j) = -us(3,j)
c                     vs(1,j) = vs(3,j)
c                     ro(1,j) = ro(3,j)
c                     t(1,j) = t(3,j)
                      us(1,j) =0.0
                      vs(1,j) = 0.0
                      ro(1,j) = roamb/roo
                      t(1,j) = tamb/to

            p(1,j) = ( ((R*(t(1,j)*to))/( 1.0/(ro(1,j)*roo)-b))-
     &                            (a/((t(1,j)*to)**(1.0/2.0)
```

```
     &                  * (1.0/(ro(1,j)*roo))*( 1.0/(ro(1,j)*roo+b))) )/po

c     p(1,j) = ro(1,j)*t(1,j)
c     e(1,j) = t(1,j)*( cv/R)
            e(1,j) =( (h1+h1r-((R*t(1,j)*to)*(( (3*a)/
     &                  (2*b*R*(t(1,j)*to)**(3.0/2.0))
     &                  *(log(1+(b/(1./(ro(1,j)*roo))))))
     &                  -(p(1,j)*po*(1./(ro(1,j)*roo)))/(R*(t(1,j)*to))+1))
     &              +(R)*
     &              ((acp*((t(1,j)*to)-t1))+
     &                  ((bcp/2.0)*(((t(1,j)*to)**2.0)-(t1**2.0)))+
     &                  ((ccp/3.0)*(((t(1,j)*to)**3.0)-(t1**3.0)))+
     &                  ((dcp/4.0)*(((t(1,j)*to)**4.0)-(t1**4.0))))-
     &                  ( (p(1,j)*po)/(ro(1,j)*roo) ) ) )/(to*R)

            u(1,j,1) =  ro(1,j)
            u(1,j,2) =  ro(1,j) * us(1,j)
            u(1,j,3) =  ro(1,j) * vs(1,j)
            u(1,j,4) =  ( (( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
     &                                        ) + e(1,j) )*ro(1,j)


            f(1,j,1) = ro(1,j) * us(1,j)
            f(1,j,2) = ro(1,j) * us(1,j) * us(1,j)
     &                                        + p(1,j)*Zo
            f(1,j,3) = ro(1,j) * us(1,j) * vs(1,j)
            f(1,j,4) = ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
     &                        + e(1,j) )*ro(1,j))
     &                   +p(1,j)*Zo)*us(1,j)

            g(1,j,1) = ro(1,j) * vs(1,j)
            g(1,j,2) = ro(1,j) * us(1,j) * vs(1,j)
            g(1,j,3) = ro(1,j) * vs(1,j) * vs(1,j)
     &                               +p(1,j)*Zo
            g(1,j,4) = ((( ( (1./2.)*(us(1,j)**2.+ vs(1,j)**2.) )
     &                        + e(1,j) )*ro(1,j))
     &                   +p(1,j)*Zo)*vs(1,j)

            jj(1,j,1) = eps * ( (1.0/y)*ro(1,j)*vs(1,j) )
            jj(1,j,2) = eps * ( (1.0/y)*ro(1,j)*us(1,j)*vs(1,j) )
            jj(1,j,3) = eps * ( (1.0/y)*ro(1,j)*vs(1,j)*vs(1,j) )
            jj(1,j,4) = eps*( (1./y)*((( ( (1./2.)*(us(1,j)**2.+
     &                        vs(1,j)**2.) )
     &                        + e(1,j) )*ro(1,j))
     &                   +p(1,j)*Zo)*vs(1,j) )
      y=y+dy
420   continue
c     apply BC at the top
c     p=po/50, t=to
c     non dimensional p=1/50, non dimensional t=1


      y=float(jmax)*dy+dy/2.

      do 430 i=1,imax
```

```
        us(i,jmax) =0.0
c       vs(i,jmax) =  (1/.02366)*(1./1600.)
c       vs(i,jmax) =  ( roj/roamb )*(1./1600.)
                vs(i,jmax) = 2*vs(i,jmax-1) - vs(i,jmax-2)
c               vs(i,jmax) = 0.0
                ro(i,jmax) = roamb/roo
                t(i,jmax) = tamb/to

        p(i,jmax) =( ((R*t(i,jmax)*to)/( 1.0/(ro(i,jmax)*roo)-b))-
   &                    (a/( (t(i,jmax)*to)**(1.0/2.0)
   &                    * (1.0/(ro(i,jmax)*roo))*( 1.0/(ro(i,jmax)*roo)+b))) )/po

c       p(i,jmax) = ro(i,jmax)*t(i,jmax)
c       e(i,jmax) = t(i,jmax)*( cv/R)
                e(i,jmax) =( (h1+h1r-((R*t(i,jmax)*to)*(( (3*a)/
   &                    (2*b*R*(t(i,jmax)*to)**(3.0/2.0))
   &                    *(log(1+(b/(1./(ro(i,jmax)*roo))))))
   &                    -(p(i,jmax)*po*(1./(ro(i,jmax)*roo)))/
   &               (R*(t(i,jmax)*to))+1))
   &               +(R)*
   &               ((acp*((t(i,jmax)*to)-t1))+
   &                    ((bcp/2.0)*(((t(i,jmax)*to)**2.0)-(t1**2.0)))+
   &                    ((ccp/3.0)*(((t(i,jmax)*to)**3.0)-(t1**3.0)))+
   &                    ((dcp/4.0)*(((t(i,jmax)*to)**4.0)-(t1**4.0))))-
   &                    ( (p(i,jmax)*po)/(ro(i,jmax)*roo) ) ) )/(to*R)

                u(i,jmax,1) =  ro(i,jmax)
                u(i,jmax,2) =  ro(i,jmax) * us(i,jmax)
                u(i,jmax,3) =  ro(i,jmax) * vs(i,jmax)
                u(i,jmax,4) =  ( (( (1./2.)*(us(i,jmax)**2.+ vs(i,jmax)**2.) )
   &                                            ) + e(i,jmax) )*ro(i,jmax)


                f(i,jmax,1) = ro(i,jmax) * us(i,jmax)
                f(i,jmax,2) = ro(i,jmax) * us(i,jmax) * us(i,jmax)
   &                                            + p(i,jmax)*Zo
                f(i,jmax,3) = ro(i,jmax) * us(i,jmax) * vs(i,jmax)
                f(i,jmax,4) =  ((( ( (1./2.)*(us(i,jmax)**2.+ vs(i,jmax)**2.) )
   &                            + e(i,jmax) )*ro(i,jmax))
   &                      +p(i,jmax)*Zo)*us(i,jmax)

                g(i,jmax,1) = ro(i,jmax) * vs(i,jmax)
                g(i,jmax,2) = ro(i,jmax) * us(i,jmax) * vs(i,jmax)
                g(i,jmax,3) = ro(i,jmax) * vs(i,jmax) * vs(i,jmax)
   &                                    +p(i,jmax)*Zo
                g(i,jmax,4) =    ((( ( (1./2.)*(us(i,jmax)**2.+ vs(i,jmax)**2.) )
   &                            + e(i,jmax) )*ro(i,jmax))
   &                      +p(i,jmax)*Zo)*vs(i,jmax)


                jj(i,jmax,1) = eps * ( (1.0/y)*ro(i,jmax)*vs(i,jmax) )
                jj(i,jmax,2) = eps * ( (1.0/y)*ro(i,jmax)*us(i,jmax)*vs(i,jmax) )
                jj(i,jmax,3) = eps * ( (1.0/y)*ro(i,jmax)*vs(i,jmax)*vs(i,jmax) )
                jj(i,jmax,4) = eps*( (1./y)*  ((( ( (1./2.)*(us(i,jmax)**2.+
   &                            vs(i,jmax)**2.) )
   &                            + e(i,jmax) )*ro(i,jmax))
```

```
     &                    +p(i,jmax)*Zo)*vs(i,jmax))
430        continue
c          apply BC on the axis of symmetry
cc         fluxes of mass, momentum, energy are put to zero along the axis
c          apply reflection technique
           y=0.+dy/2.

           do 440 i=1,imax

                  us(i,1) = us(i,3)
c                 write(6,*) 'vs at i=2',vs(i,2)
c          12/14: I will remove the v=0 condition on axis, since i applied
c          v at ghost row ( row 1) = v at row above axis( row 3)

c                 vs(i,2) = 0.0
c                 write(6,*) 'vs at i=2',vs(i,2)
                  vs(i,1) = -vs(i,3)
                  ro(i,1) = ro(i,3)
                  t(i,1) = t(i,3)

           p(i,1) = ( ((R*t(i,1)*to)/( 1.0/(ro(i,1)*roo)-b))-
     &                   (a/((t(i,1)*to)**(1.0/2.0)
     &                   * (1.0/(ro(i,1)*roo))*( 1.0/(ro(i,1)*roo)+b))) )/po

c          p(i,1) = ro(i,1)*t(i,1)
c          e(i,1) = t(i,1)*(cv/R )
                  e(i,1) =( (h1+h1r-((R*t(i,1)*to)*(( (3*a)/
     &                   (2*b*R*(t(i,1)*to)**(3.0/2.0))
     &                   *(log(1+(b/(1./(ro(i,1)*roo))))))
     &                   -(p(i,1)*po*(1./(ro(i,1)*roo)))/(R*(t(i,1)*to))+1))
     &              +(R)*
     &              ((acp*((t(i,1)*to)-t1))+
     &                   ((bcp/2.0)*(((t(i,1)*to)**2.0)-(t1**2.0)))+
     &                   ((ccp/3.0)*(((t(i,1)*to)**3.0)-(t1**3.0)))+
     &                   ((dcp/4.0)*(((t(i,1)*to)**4.0)-(t1**4.0))))-
     &                   ( (p(i,1)*po)/(ro(i,1)*roo) ) ) )/(to*R)


                  u(i,1,1) =  ro(i,1)
                  u(i,1,2) =  ro(i,1) * us(i,1)
                  u(i,1,3) =  ro(i,1) * vs(i,1)
                  u(i,1,4) =  ( (( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
     &                                               ) + e(i,1) )*ro(i,1)


                  f(i,1,1) = ro(i,1) * us(i,1)
                  f(i,1,2) = ro(i,1) * us(i,1) * us(i,1)
     &                                               + p(i,1)*Zo
                  f(i,1,3) = ro(i,1) * us(i,1) * vs(i,1)
                  f(i,1,4) =   ((( ( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
     &                              + e(i,1) )*ro(i,1))
     &                    +p(i,1)*Zo)*us(i,1)


                  g(i,1,1) = ro(i,1) * vs(i,1)
                  g(i,1,2) = ro(i,1) * us(i,1) * vs(i,1)
```

```
                  g(i,1,3) = ro(i,1) * vs(i,1) * vs(i,1)
    &                                  +p(i,1)*Zo
                  g(i,1,4) = ((( ( (1./2.)*(us(i,1)**2.+ vs(i,1)**2.) )
    &                          + e(i,1) )*ro(i,1))
    &                      +p(i,1)*Zo)*vs(i,1)


                  jj(i,1,1) = eps * ( (1.0/y)*ro(i,1)*vs(i,1) )
                  jj(i,1,2) = eps * ( (1.0/y)*ro(i,1)*us(i,1)*vs(i,1) )
                  jj(i,1,3) = eps * ( (1.0/y)*ro(i,1)*vs(i,1)*vs(i,1) )
                  jj(i,1,4) = eps*( (1./y)*((( ( (1./2.)*(us(i,1)**2.+
    &                      vs(i,1)**2.) )
    &                          + e(i,1) )*ro(i,1))
    &                      +p(i,1)*Zo)*vs(i,1) )



440         continue

                  do 510 i=imax-1, imax-1
                        do 510 j = 2,jmax
                              if(p(i,j).lt.0.)then
            write(6,*)i,j,p(i,j)
            end if
c                             mach(i) = v(n+1,i) / sqrt( t(n+1,i) )

c           write(6,*)ro(i,j)*t(i,j)*gam
c           write(18,*)ro(i,j)*t(i,j)*gam

c                             write(18,91) n,us(i,j),ro(i,j),t(i,j),ro(i,j)*t(i,j)*gam
c                             write(6,91) n,us(i,j),ro(i,j),t(i,j),ro(i,j)*t(i,j)*gam
c    &                        ( (us(i,j)**2.0
c    &                        + vs(i,j)**2.0)/(gam*t(i,j)) ) **(1./2.)
c                             write(6,*) n

c                             write(28,*)p(i,j)
c                             write(30,*)t(i,j)
c                             write(18,61) n,i,j,press(i,j),ro(i,j),e(i,j)
510                     continue



            write(6,*)n

100         continue


c           write plate profiles T and P
            do 500 i=imax-1, imax-1
                  do 500 j=2,jmax
                              write(40,*)p(i,j), t(i,j)
500         continue


c           write IC  for viscosity codes
            do 555 i = 1,imax
```

```
                        do 555 j = 1,jmax
                                write(18,*)             us(i,j)
                                write(19,*)             vs(i,j)
                                write(20,*) t(i,j)
                                write(21,*) ro(i,j)
555             continue

c       write centerline properties
        do 5003 j = 2,2
                        do 5003 i = 1,imax
                        write(22,*)ro(i,j), t(i,j),p(i,j)
5003    continue

c       write for matlab density and velocity files
        do 5551 i = 1,imax
                        do 5551 j = 2,jmax
                                write(61,*)ro(i,j)
5551            continue

        do 5550 i = 1,imax,8
                        do 5550 j = 2,jmax,8
                                write(62,*)
   &                                                    us(i,j)
                                write(63,*)
   &                                                    vs(i,j)
5550    continue


11      format(i4,5(f7.3,2x))
21      format ( 'u1=',f7.4,2x,'u2=',f7.4,2x,'u3=',f7.4)
31      format(i4,f10.4,2x,f7.4,2x,f10.4)
41      format(9(f7.3,2x))
51      format(i4,i4,7(f7.3,2x))
61      format(i4,i4,i4,3(f15.6,1x))
71      format(12x,3(f14.10,4x))
81      format(i4,i4,4(f7.3,2x))
91      format(i4,4(f15.4,1x))
101     format(i4,i4,4(f7.3,2x))
111     format(4(f12.4,2x))

        end
```