A Study of Gas Spring Heat Transfer in Reciprocating Cryogenic Machinery

by

James Nathaniel Chafe

B.S., Tulane University
(1982)

S.M., Massachusetts Institute of Technology
(1984)

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

July 1988

Signature of Author_____
Department of Mechanical Engineering
July 18, 1988

Certified by_____
Professor Joseph L. Smith, Jr.
Thesis Supervisor

Accepted by_____
Professor Ain A. Sonin
Chairman, Departmental Graduate Committee

A Study of Gas Spring Heat Transfer in Reciprocating Cryogenic Machinery

by

James Nathaniel Chafe

Submitted to the Department of Mechanical Engineering
on July 18, 1988 in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy
in Mechanical Engineering

Abstract

This thesis is a study of the gas spring heat transfer which occurs in reciprocating machinery and the way it affects the performance of these machines particularly cryogenic reciprocators. In this study, a closed volume piston-cylinder apparatus is used to measure gas spring heat transfer under a variety of conditions at room and cryogenic temperatures. Analytic models are developed to represent the gas spring heat transfer and the interaction between the heat transfer and the thermodynamic behavior of the working gas. Pressure and volume waves for a complete cycle from the experiments and analytic models are represented in the form of Fourier series. This way the data are represented by a relatively small number of parameters and without extensive reduction and can be manipulated mathematically with relative ease.

We have found that gas spring heat transfer is primarily affected by the piston cycle frequency and the thermal diffusivity of the working gas and that its effects are potentially detrimental for room temperature reciprocators for reciprocators which operate near 77 K, high cycle frequencies would help reduce gas spring heat transfer losses. For temperatures near 10 K, the effects of gas spring heat transfer are small; this is mostly attributed to the extremely large gas densities which occur at these temperatures.

Thesis Supervisor: Professor Joseph L. Smith, JR.
Title: Professor of Mechanical Engineering

Acknowledgments

I thank Professor J.L. Smith, Jr., Dr. Yukikazu Iwasa, and Professor Borivoje Mikic for all their guidance and technical assistance. Recognition should go to Sumitomo Heavy Industries for providing the financial support to carry out this work.

A special thanks to Al Kornhauser for his suggestions, encouragement and friendship. I especially would like to thank my wife Josefina for all the help and support she has given me. She performed above and beyond the call of duty with the help she gave me in preparing this transcript.

Table of Contents

Table of Figures

Table of Tables

List of Symbols

A	heat transfer area
A_{sur}	surface area of cylinder at mid-stroke
b	coefficient of heat penetration (eqn. 5.17)
C	helium thermal capacitance for one space model
C_p	helium specific heat at constant pressure
C_v	helium specific heat at constant volume
C_1, C_2	thermal capacitance at indicated node for two-space model
D	cylinder diameter
d	depth of thermal boundary layer
h_f	film coefficient
h	specific enthalpy
k	thermal conductivity
L	length of the appendix gap
$Loss$	nondimensional thermodynamic loss (eqn, 5.8)
M	total mass of gas
m	mass
m_1, m_2	mass of gas at indicated node for two space model
M_T	total mass in cylinder working space for gap-flow model
M_g	mass in the appendix gap for gap-flow model
P	gas pressure
P_∞	pressure if all gas is in appendix gap for gap-flow model (eqn. 4.62)
Pe	Peclet number (eqn. 3.1)
P_0, P_1, P_2	pressure amplitude for indicated harmonic
Pad_n	n^{th} harmonic adiabatic pressure amplitude
$Pisot_n$	n^{th} harmonic isothermal pressure amplitude

P_{poly}	polytropic pressure (eqn. 3.6)
P_{ref}	reference pressure
q	work input to gas for one space model
q_A	work input to gas in region 1 for two-space model
q_B	work input to gas in region 2 for two-space model
q_1	heat transfer to cylinder wall for one space model
q_2	rate of energy storage in the gas for one space model
q_a, q_b, q_c, q_d	heat fluxes defined by equations 4.24 through 4.27
$\dot{Q}$	heat transfer rate to the gas
r	thermal resistance for one-space model
r_1, r_2	thermal resistance at indicated node for two-space model
R	specific gas constant
t	time
t_1, t_2, t_3	time constants defined in equations 4.42 to 4.44
T	temperature
T_g	gas temperature in gap for gap-flow model
T_l	average temperature of cylinder wall
T_1, T_2	temperatures at indicated nodes for two-space model
T_{mm}	mixed mean temperature
TR	temperature ratio parameter (eqn 4.64)
U_p	piston speed (eqn. 3.2)
U	total internal energy of gas
u	specific internal energy of gas
V	volume of main working space for gap-flow model
V_t	time-average volume of main working space for gap-flow model
V_m	amplitude of volume wave of main working space for gap-flow model
V_{eq}	equivalent volume (eqn 4.54)

V_g	volume of appendix gap for gap-flow model
VR	volume ratio (eqn. 4.13)
VRG	gap volume ratio parameter (4.66)
$\dot{W}$	rate of work done by the gas
x	location along appendix gap in gap-flow model
z	polytropic exponent

Greek Symbols

α	helium thermal diffusivity
Δ	difference operator
γ	ratio of specific heats (eqn. 4.9)
θ_1,θ_2	pressure phase angle of indicated harmonic
λ	thermal wavelength (eqn. 4.19)
μ	parameter representing distribution of helium mass for two-space model (eqn. 4.35)
ρ	helium density
τ	nondimensional speed for one space model (eqn. 4.14)
τ_1, τ_2	nondimensional speeds for two-space model (eqns. 4.36, 4.37)
τ_h	nondimensional speed for hybrid model (eqn. 4.73)
υ	total gas volume
υ_0	average gas volume
υ_m	amplitude of volume wave
ϕ	phase angle of the volume wave
ω	angular velocity

Subscripts

ad	adiabatic
isot	isothermal
g	appendix gap
he	helium
n	n^{th} harmonic
ss	stainless steel

Superscripts

*	nondimensional

Chapter 1
INTRODUCTION

This report documents the experimental and analytical work, conducted at the Cryogenic Engineering Laboratory at M.I.T., on one of the loss mechanisms affecting reciprocating machinery in general and cryogenic reciprocating machinery in particular. We term the phenomenon that causes this loss "gas spring heat transfer" which is the heat transfer associated with the temperature changes of a gas caused by gas expansion and compression in a reciprocator.

This study consists of carrying out gas spring heat transfer experiments at room temperature (300 K), liquid nitrogen temperature (77 K), and near liquid helium temperature (approximately 10 K). Four analytic models are also developed for gas spring heat transfer. Each model emphasizes different physical characteristics. For both the experiments and the models, parameters representing the operating conditions are correlated to those representing the gas spring heat transfer and to the thermodynamic loss associated with the heat transfer. The correlations for the models are compared to the correlations for the experiments to verify the validity of the models.

The remainder of this thesis is divided into four chapters. Chapter 2 provides the necessary background for understanding this work on gas spring heat transfer. Chapter 3 describes the experiments, the methods used for data reduction, and the results of the experiments. Chapter 4 presents the four analytical models and their results relative to the experiments' results. Finally, Chapter 5 presents the general results and conclusions of this study of gas spring heat transfer.

Chapter 2
BACKGROUND THEORY

In this thesis a "gas spring" refers to a piston in a cylinder which has no intake or exhaust ports. That is, there is no influx or efflux of the working gas in the cylinder. This gas can be considered a thermodynamically closed system. The piston face applies work to the system while heat flows between the gas and the cylinder wall. This type of device is common; for example, an hydraulic accumulator may be thought of a gas spring. The behavior of many open cycle machines during parts of their cycles also closely resembles a gas spring. A reciprocating expander or compressor acts much like a gas spring in the portions of their cycles when the valves are closed, i.e. during the compression and expansion strokes.

We focus our attention on cryogenic machinery in particular because comprehensive heat transfer studies on these machines are not available. Reciprocators are used quite often for cryogenic operation. Collins[1] used a pair of expansion engines (open cycle reciprocators) to produce the first commercially successful helium liquefiers and this basic design is still in widespread use. More recently, a third expansion engine has been added to the design of some helium liquefiers. This engine runs in place of the Joule-Thomson valve where the gaseous helium is liquefied therefore the engine operates on two-phase helium. Experimental cycles[2] which use a reciprocating cryogenic compressor are being developed at the Cryogenic Engineering Laboratory of M.I.T. Other devices involving cryogenic reciprocators are also under development at this laboratory. These include miniature cryocoolers[3] as well as solid refrigerant magnetic cryocoolers.[4] Stirling type machines are also in wide spread use for achieving cryogenic temperatures.[5] Thus not only are cryogenic reciprocators in widespread use but there is also considerable effort to use them for more innovative

---

1 Barron, R., *Cryogenic Systems*, McGraw-Hill Book Company, New York, 1966, pg. 122.

2 Minta, M., *Analytical and Experimental Studies of an Optimal Helium Liquefaction Cycle*, Sc.D. thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1984.

3 Crunkleton, J.A., *A New Configuration for a Liquid Helium Temperature Cryocooler*, Ph.D. thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1987.

4 Taussig, C.P., *Magnetically Active Regeneration*, Ph.D. thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1986.

5 Barron, R., *Cryogenic Systems*, McGraw-Hill Book Company, New York, 1966, pg. 318.

cryogenic refrigeration. A good understanding of the heat transfer within these reciprocating machines during cycles is very important in achieving optimum operating conditions in these machines; it is perhaps even more important in the design of new machines and the development of new refrigeration systems.

In addition to cryogenic machinery, the gas spring heat transfer studied in this thesis may also play an important role for machines which operate at room temperatures or even elevated temperatures. Certainly any insight into the heat transfer processes which occur at cryogenic temperatures will be helpful in understanding the processes which take place at room temperature and vice-versa. This thesis is part of ongoing gas spring heat transfer work at the Cryogenic Engineering Laboratory of M.I.T. Faulkner[6] provided the initial investigation of this heat transfer by performing experiments with a modified gas compressor operated at room temperature. Kornhauser[7] is continuing this room temperature work with a special emphasis on the ways this heat transfer affects Stirling type machines. He has presented a comparison of some of the more important research published on this subject.[8]

2.1 Gas Spring Heat Transfer

In a gas spring when the piston moves from bottom dead center to top dead center the gas is compressed and most of the work of compression is stored in the gas. The work of compression not only elevates the gas's pressure but also raises its temperature. When the piston is allowed to return to bottom dead center, the work of compression is released. As this happens, energy is recovered from the gas in the form of work and both the pressure and temperature of the gas are lowered. Therefore as the pressure of the gas changes cyclically, so does its temperature. For most operating conditions, the heat capacity of the cylinder walls is much larger than that of the gas. This means that over a cycle the cylinder wall temperature remains relatively constant while the gas experiences a cyclic temperature variation, causing heat to flow between the two. We refer to this as gas spring heat transfer which can occur not only in a gas spring but also in any machine

---

6 Faulkner, H.B., *An Investigation of Instantaneous Heat Transfer During Compression and Expansion In Reciprocating Gas Handling Equipment*, Ph.D. Thesis, Mechanical Engineering Department, Massachusetts Institute of Technology, May 1983.

7 Kornhauser, A.A., *Gas-Wall Heat Transfer During Compression and Expansion*, Ph.D. thesis, Department of Mechanical Engineering, M.I.T., (expected publication date: February, 1989).

8 Kornhauser, A.A., Smith, J.L., "A Comparison of Cylinder Heat Transfer Expressions Based on Prediction of Gas Spring Hysteresis Loss", *Fluid Flow and Heat Transfer in Reciprocating Machinery*, American Society of Mechanical Engineers, 1987, pp.89-96.

where there are cyclic temperature variations induced by cyclic pressure variations. This heat transfer causes a thermodynamic loss which means that not all of the work of compression is recovered upon re-expansion.

2.2 Gap-Flow Heat Transfer

The design of cryogenic reciprocators is strongly influenced by the cryogenic operating temperatures. These machines are usually designed with long pistons which have relatively short strokes to provide a large amount of thermal isolation between the cold working end and the room temperature end. Typically, the seal between the piston and cylinder is located at the room temperature end of the piston creating a long annular gap between the piston and cylinder. As the density of the gas varies over a cycle, the amount of gas within the gap changes, causing gas to flow in and out of the gap. The temperature of the gas which flows in and out of the gap can be different from the gap wall temperature. This may cause a significant amount of heat transfer between the two which we refer to as the gap-flow heat transfer. As is the case for gas-spring heat transfer, the heat transfer due to gap-flow has a thermodynamic loss associated with it.

2.3 Other Losses for Reciprocators

There are other processes that produce thermodynamic loss within a reciprocator besides those due to gas spring heat transfer and gap-flow. The piston motion induces what is sometimes referred to as solid convection or shuttle heat transfer[9]. This refers to energy which is transported in a stepwise fashion down the length of the cylinder wall. Other losses include steady state conduction down the cylinder wall and the piston, thermal radiation to the cylinder wall, and friction losses. The friction losses are attributable to the viscosity of the fluid, the seal between the piston and cylinder and any rubbing between the piston and cylinder wall. For open cycle reciprocators, there are also losses associated with the gas flowing in and out of the valves.

This thesis focuses on formulating analytical models for the gas-spring losses and to a lesser extent gap-flow. The analytical models are based on a series of experimental results derived from an apparatus specifically designed for this purpose, with losses other than gas spring losses either eliminated or minimized.

---

9 Zimmerman, F.J., Longsworth, R.C., *Shuttle Heat Transfer*, Advances in Cryogenic Engineering, Volume 16, New York, NY, 1971.

2.4 Relation between P-V Diagrams and Heat Transfer

In studying gas spring heat transfer, the thermodynamic analysis for a closed system is much simpler than that for an open system. Faulkner presents a thorough discussion of the technique we use to determine the instantaneous spatially averaged heat transfer.[10] First we assume that all operating conditions have low enough Mach numbers so the pressure is always uniform in the cylinder. We also assume that the spatial temperature variations of the gas have a negligible effect on the gas specific heat. With these assumptions, the first law of thermodynamics, and Gibbs phase rule indicate that the instantaneous spatially averaged heat transfer during a cycle in a closed system can be calculated from the time histories of any two independent thermodynamic properties. By history we mean not only the value of these properties at each point in time but also how they change with time. From an experimental viewpoint the most practical choices for the two thermodynamic properties are pressure and specific volume. Since the pressure and specific volume were measured in the experiments, we chose to correlate these measurements to the operating conditions rather than to correlate the heat transfer to the operating conditions. This provides several advantages. First, the correlations are not subject to extensive manipulation required to calculate the heat transfer, i.e. numerical truncation, potentially faulty assumptions, and thermodynamic property relations. Also, careful correlations of the pressure and volume preserve nearly all the information recorded during an experiment allowing the data to be used in future analysis without the need of back-stepping through the data reduction process or the need for completely reducing the raw data again. Lastly, the method used to correlate the pressure and volume reduces them to analytic functions which are easily manipulated in a variety of different ways. In this thesis we develop a method of quantifying the pressure and volume histories so that they can be correlated to the operating conditions.

2.5 Discretized Pressure and Volume Waves

In the experiments, a microcomputer-controlled digital data acquisition system measures and records the pressure and volume of a gas within a gas spring as the piston compresses and expands the gas. The piston oscillates with a sinusoidal motion which causes the gas volume and pressure to vary as periodic waves. As measured by the data

---

10 Faulkner, H.B., *An Investigation of Instantaneous Heat Transfer during Compression and Expansion in Reciprocating Gas Handling Equipment*, Ph.D. thesis, Department of Mechanical Engineering, M.I.T., 1983, pp. 26-31.

acquisition system, each wave is represented as a series of pressure or volume values spaced at equal intervals of time. In addition to the experiments, models are developed which generate pressure and volume waves also represented by series of values equally spaced in time.

Discrete Fourier transforms are specifically applicable to data stored as a series of equally spaced points. These transforms are used to create analytic representations of the pressure and volume waves. These analytic representations are series of sinusoidal functions, i.e. Fourier series. With these Fourier series the data can be directly correlated to the operating conditions. The Fourier series are also used to calculate the heat transfer and the thermodynamic loss.

2.6 Adiabatic and Isothermal Baseline Cases

For a given cycle, reversible adiabatic and reversible isothermal pressure waves are calculated and used as baselines from which the performance of the apparatus or a model is measured. For an adiabatic cycle no heat transfer occurs between the gas and the cylinder wall. For an isothermal cycle the gas remains at a constant temperature equal to the cylinder wall temperature, in order for this to happen there is a varying amount of heat transfer that occurs over a cycle. When choosing these cycles to be reversible, we have implicitly made the assumption that the ideal thermodynamic system is constantly passing through equilibrium states which means that both the temperature and pressure distributions are uniform. We have chosen these two reversible cycles as baselines because they have no entropy generation or thermodynamic loss associated with them and the adiabatic case can be thought of as a lower bound for heat transfer while the isothermal case can be thought of as an upper bound for the heat transfer.

If we examine the pressure waves created when a gas spring operates reversibly adiabatically and reversibly isothermally, we find that adiabatic case has large pressure swing over the cycle while the isothermal case has a relatively small pressure swing. We use the magnitudes of the adiabatic and isothermal pressure waves to quantitatively compare various pressure waves of the experiments as well as the models.

Chapter 3
Experiments

This chapter presents a general description of the apparatus as well the experiment procedure and results. A detailed description of the instrumentation used in the experiments is presented in Appendix A, the computer programs used to for data acquisition and also data reduction are presented in Appendix D, and the tabulated results of the experiments is presented in Appendix B.

3.1 Description of Apparatus

The experiment apparatus was designed to operate at cryogenic temperatures. This design constraint forced the geometry of the apparatus to be very similar to many commercially available cryogenic expanders. These machines usually have a very long piston with a relatively short stroke. The dynamic piston-to-cylinder seal is located at the end of the piston nearest the drive mechanism (the room temperature end); creating a long thin gap between the piston and cylinder which opens to the main working space. Figure 3.1 is a schematic of the apparatus.

3.1.1 Piston and Cylinder

The piston is a Micarta® (phenolic impregnated linen) rod approximately 1.5 in. (38 mm) in diameter and 26 in. (660 mm) long. The cylinder is a stainless steel tube. The radial clearance between the two is approximately 0.010 in. (0.25 mm). The cylinder is comprised of two major sections. The upper section has a wall thickness of 0.5 in. (13 mm) and is approximately 10 in. (250 mm) long. The lower section has a wall thickness of only 0.020 in. (0.5 mm) and is 25 in. (630 mm) long. The thick upper section is used as a rigid mount for the drive mechanism, and also provides a thermally and mechanically stable surface for the piston-to-cylinder seal. The long thin lower section of the cylinder creates a large thermal resistance between the warm and cold ends of the cylinder.

The piston-to-cylinder seal is a buna O-ring with leather back-up rings on both its upper and lower sides. An oil-soaked felt washer lubricates the seal.

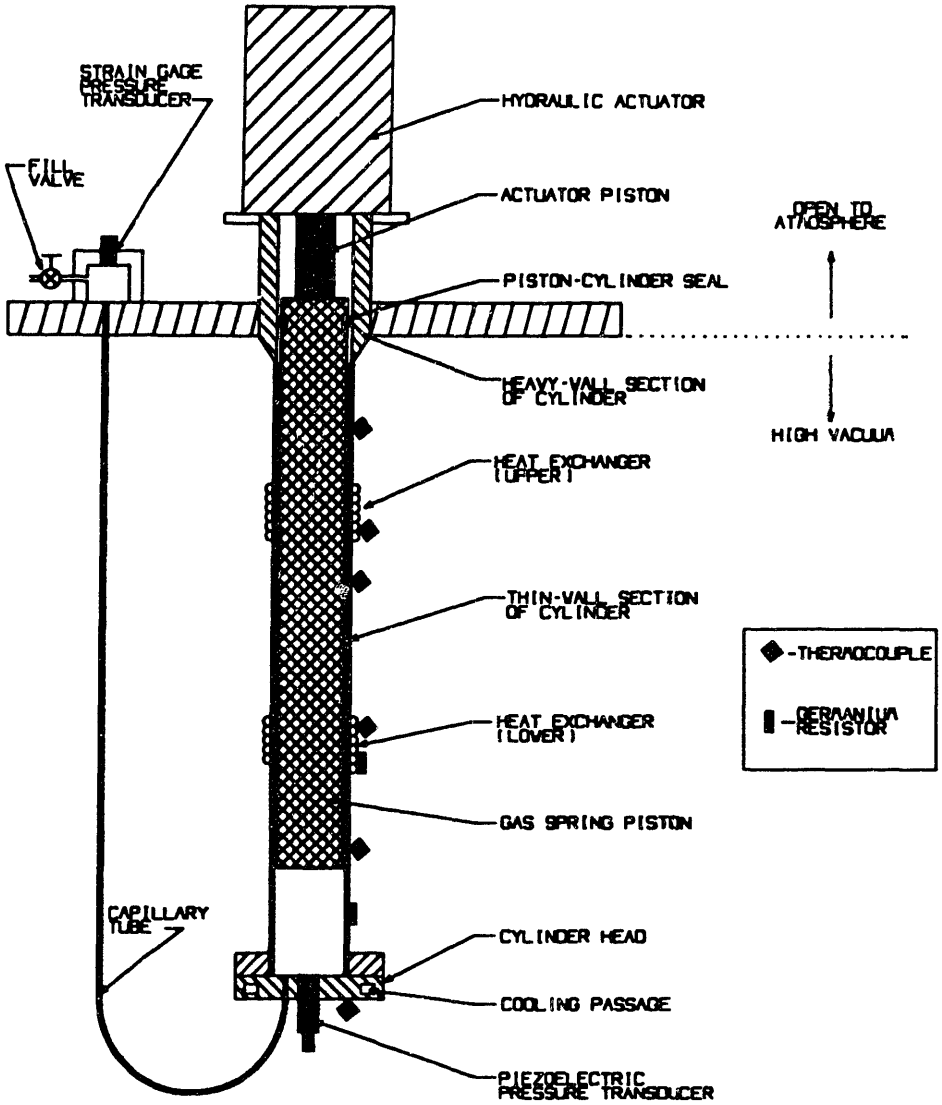

Fig. 3.1 Gas spring heat transfer experiment apparatus.

The cylinder head is a stainless steel plate which is attached to the lower end of the cylinder. There are two holes in the cylinder head. The first hole allows a pressure transducer access to the working space. A capillary tube, used to fill the cylinder with helium, passes through the second hole. This capillary tube is also connected to a pressure transducer located outside of the cylinder.

3.1.2 Drive Mechanism

A linear hydraulic actuator controlled by a servo-valve moves the piston. The actuator is powered with high-pressure hydraulic fluid. Its case is attached to the upper section of the cylinder and its piston attaches to the apparatus piston. The actuator's maximum stroke is 4 in. (100 mm). An electric signal to the servo-valve controls the motion of the actuator piston and thus the apparatus piston. For example, applying a sinusoidal voltage to the servo-valve produces an approximately sinusoidal piston motion. The pressure of the hydraulic fluid as well as the pressure level in the apparatus affect the maximum speed at which it can produce a sinusoidal motion. Other factors such as the mass of the pistons and seal friction also effect the maximum speed. Through experience we found that two cycles per second is a good maximum speed for the apparatus.

3.1.3 Cooling System

The low temperatures for the apparatus are achieved by circulating cryogens through two heat exchangers attached to the outside of the cylinder's lower section . A cooling passage in the cylinder head allows the cryogens to also cool the lowest part of the cylinder. The heat exchangers are designed to give the temperature profile characteristics shown in Figure 3.2. These heat exchangers are long enough to provide isothermal regions longer than the stroke of the piston. These isothermal regions eliminate or at least reduce both the conduction and *shuttle*[11,12] heat transfer to the lower end of the apparatus.

---

11 Zimmerman, F.J., Longsworth, R.C.,*Shuttle Heat Transfer*, Advances in Cryogenic Engineering, Volume 16, Plenum Press, New York NY, 1971.

12 Rios, P.A., *An Approximate Solution to the Shuttle Heat-Transfer Losses in a Reciprocating Machine*, Transactions of the ASME, Journal of Engineering for Power, April 1971.

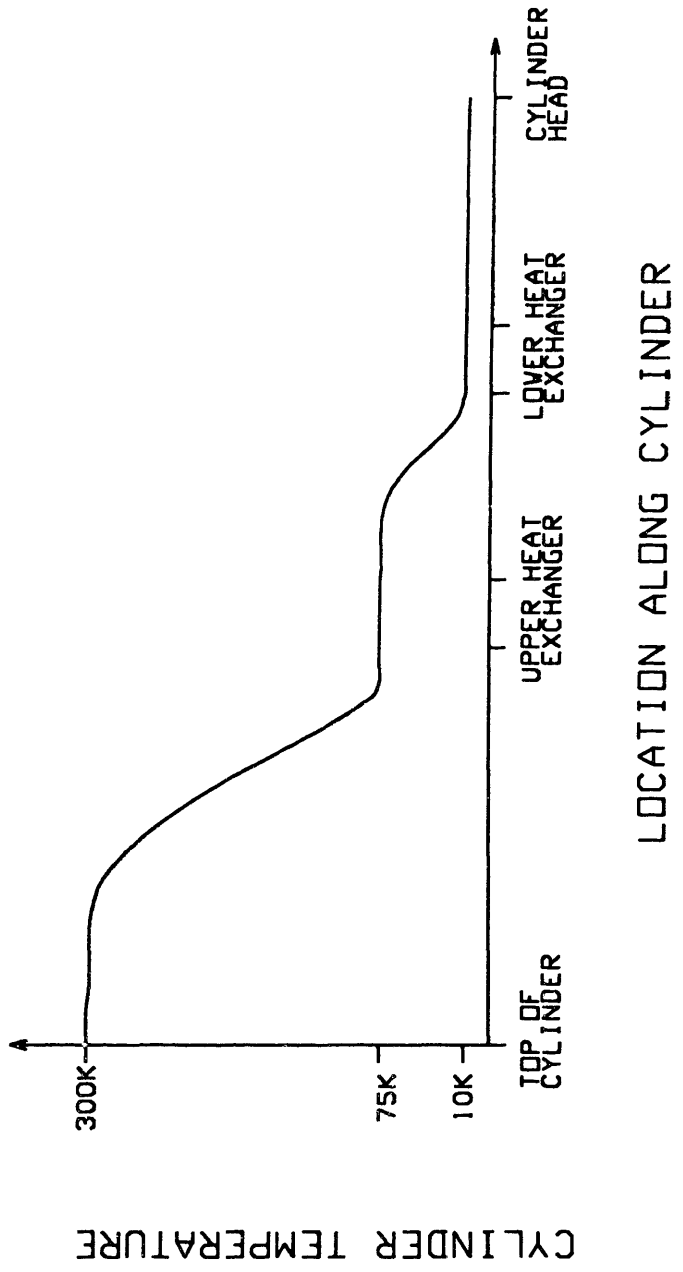

Fig. 3.2 Sketch of temperature profile during a liquid helium temperature experiment run.

The lower section of the cylinder is surrounded by a vacuum chamber. A high vacuum environment minimizes any convection heat transfer to the apparatus when operating at cryogenic temperatures. In order to minimize radiation heat transfer, the experiment is wrapped with four to five layers of aluminized mylar which has high reflectivity. A liquid nitrogen cooled radiation shield lines the vacuum vessel to further reduce radiation heat transfer.

3.1.4 Instrumentation

Instruments incorporated into the apparatus measure the necessary information required to calculate the gas spring heat transfer. This instrumentation consists of pressure and volume sensors and also temperature sensors to record the operating conditions during an experiment. Figure 3.1 shows the locations of this instrumentation.

3.1.4.1 Volume Measurement

A linear-variable-differential-transformer (LVDT) measures the volume of the working space. This device is part of the hydraulic actuator system and it produces a voltage signal proportional to the axial position of the piston. The position of the piston is proportional to the volume of the working space and therefore the voltage output of the LVDT is also proportional to the volume.

3.1.4.2 Pressure Measurement

Two separate transducers are used to measure pressure. One transducer, mounted in the cylinder head, monitors the instantaneous pressure during a data run. The other transducer measures static pressures.

The transducer mounted in the cylinder head is a piezoelectric pressure transducer. It was selected for its ability to operate at cryogenic temperatures. Serious limitations of this transducer include its inability to make absolute pressure measurements as well as its inability to make static pressure measurements.

The other transducer is a strain gage type pressure transducer capable of making absolute and static pressure measurements. However, it is not able to operate at cryogenic temperatures and therefore is mounted on a plate near the upper section of the cylinder outside of the vacuum chamber. It is connected to

the working space via the stainless steel capillary tube. This pressure transducer does not suffer from any serious drift problems. The disadvantage of mounting it in a remote location is that there is a significant time delay for a pressure wave to travel through the capillary tube. Since the time relationship between the pressure and volume of the helium is critical for this experiment, this pressure transducer is used only to make static pressure measurements.

3.1.4.3 Temperature Measurement

Temperature measurements are needed for two reasons. First, we need to know when the desired operating temperature is reached. Second, we need the temperature to help determine the thermodynamic state point for the helium. This state point and the total volume of the working space determine the helium mass in the apparatus.

Figure 3.1 shows the locations of the temperature sensors. Thermocouples are used for temperature measurements as low as 70 K. There are thermocouples located at 6 in. (150 mm) intervals along the length of the cylinder including one in the cylinder head and one on each of the heat exchangers. Germanium resistors are used to measure temperatures below 70 K. There is one germanium resistor located on the lower heat exchanger and one located on the cylinder half way between the lower heat exchanger and the cylinder head.

3.1.5 Data Acquisition and Control System

The data collection equipment consists of several microcomputer-controlled digital-to-analog converters (DACs) and analog-to-digital converters (ADCs). These devices interface the computer with the apparatus's instrumentation. The DACs allow the computer to control the piston position and the ADCs allow the computer to monitor and record the volume, pressure, and temperature of the apparatus. Appendix D presents a detailed description of the data collection equipment.

During a run, the computer sends a series of digital signals which direct a DAC to output a sinusoidal voltage to the drive mechanism creating the desired piston motion. While this is happening, the computer directs the ADCs to digitize the signals coming from the pressure and volume sensors. The computer then records the digitized signals and stores them for later reduction.

3.2 Operating Conditions

Experiments were performed at various operating conditions. These operating conditions were chosen to provide data relevant to *standard* reciprocating equipment and to span the most interesting parameter ranges. Operating at cryogenic temperatures produced data previously not available and also demonstrated the important role of both the geometry of these machines and their material properties.

3.2.1 Temperatures

The cooling system allowed the apparatus to operate at three temperature ranges. The room temperature range was achieved by operating the apparatus with no cooling or heating of any kind. The intermediate temperature range (approximately 77 K) was achieved by circulating liquid nitrogen through the lower heat exchanger and the cooling passage in the cylinder head. The cold temperature range (from 7 K to 15 K) was achieved by circulating two phase helium through the lower heat exchanger and cylinder head while liquid nitrogen is circulated through the upper heat exchanger.

3.2.2 Pressure Levels

The apparatus was filled with helium to several different pressure levels while the piston was at mid stroke. The capillary tube is connected through a valve to a high pressure helium source. To change the mass of helium, the piston was positioned at mid stroke and the valve was opened. Helium then filled the cylinder to a specified pressure. These pressures ranged from 1 atm to approximately 17 atm. They were usually spaced a factor of two apart. For example, if a set of experiments was run at 17 atm then the next set of experiments would be run at 125 psia (8.5 atm).

3.2.3 Speeds

The experiments were run at six different speeds. The maximum speed was a cycle period of 0.5 s (120 rpm) and the slowest speed was a cycle period of 16 s (3.75 rpm). The intermediate speeds included cycle periods of 1.0, 2.0, 4.0, and 8.0 s.

3.3 Data Collection

The following section outlines the procedure used to collect data. The data collection consisted of a complex series of tasks which were carried out for each run. Some of the tasks were performed automatically by the computer and data acquisition system, and some tasks were performed manually.

3.3.1 Cool-Down

The first step was to bring the experiment to the desired operating temperature. For experiments conducted at room temperature, no cool-down was necessary. For experiments conducted at liquid nitrogen temperature (approximately 77 K), liquid nitrogen was circulated through the appropriate cooling passages.

Extra precautions were taken for helium-temperature experiments (7 K to 15 K). Before the apparatus was cooled down, the working space was carefully purged so that there was no air in the system. During the cool-down, the pressure in the apparatus was maintained above atmospheric pressure. These precautions were taken to prevent air from freezing in the thin capillary tube and plugging it. Liquid nitrogen was then circulated through the upper heat exchanger and cold helium was circulated through the lower heat exchanger and the cylinder head.

After the apparatus had cooled down to the operating temperature, the pressure level in the cylinder was adjusted to a set value. The helium mass in the cylinder was determined from the measured temperature, pressure, and the total helium volume. For room temperature and liquid nitrogen temperature runs, the pressure adjustment procedure included purging the apparatus of air. For the helium temperature runs this was done before cool-down.

For a given pressure level, experiments were conducted at each piston speed. The speeds were usually selected automatically by the data acquisition computer program. The first experiment for a given pressure level was run at the slowest speed (16 s period). Then the speeds were increased for each run until the maximum speed (0.5 s period) was reached.

3.3.2 Piezoelectric Pressure Transducer Calibration

The piezoelectric pressure transducer is a system consisting of a pressure sensor and an amplifier. The amplifier converts the output from the sensor to an electrical signal more suitable to conventional electronic measuring equipment. To get the highest resolution and accuracy from the data acquisition system, the amplifier gain was adjusted for individual data collection runs. Calibrations were performed to measure these changes in gain.

Before each experiment run the piezoelectric pressure transducer was calibrated. This calibration consisted of the following steps:

1. The outputs of both the piezoelectric and the strain gage pressure transducers were recorded over a 5 s time interval with the piston at mid volume;
2. The piston was slowly moved to the maximum volume position (bottom dead center) and the gas pressure was allowed sufficient time to settle;
3. At this piston position, the outputs of the pressure transducers were recorded over a 5 s time interval;
4. The piston was then moved to the minimum volume (top dead center) and again the gas pressure was allowed sufficient time to settle;
5. At this piston position, the outputs of the pressure transducers were recorded over a 5 s time interval;
6. The piston was then moved back to the original mid-volume position and the gas pressure was allowed sufficient time to settle;
7. At this piston position, the outputs of the pressure transducers were recorded over a 5 s time interval.

The strain gage pressure transducer was used as the *standard* for these calibrations. Using the pressure levels from the above procedure, sensitivity coefficients for the piezoelectric pressure transducers were then calculated .

3.3.3 P-V Data Collection

After the calibration, the collection of pressure and volume data began. First, the data acquisition system commanded the the hydraulic actuator to oscillate the

piston in the desired wave form. Then the pressure was monitored until the apparatus reached cyclic steady state. The apparatus had reached cyclic steady state when the difference between pressure at the beginning of a cycle and the pressure at the end of the same cycle was within a specified tolerance (this tolerance varied depending on the pressure level). Once cyclic steady state was reached, three full cycles of both pressure and volume data were recorded.

3.3.4 Piezoelectric Pressure Transducer Recalibration

After the P-V data collection, the piezoelectric pressure transducer was calibrated again in the same way as presented in section 3.3.2. The temperature sensitivity of the transducer required that it be calibrated before and after each run. During a run, the piston motion can affect the operating temperature in the working space. This can have a significant effect on the piezoelectric pressure transducer sensitivity especially at low temperatures. The calibrations quantified the changes in sensitivity. The second calibration was compared with the first and if the pressure transducer sensitivity had changed significantly, then the run was considered unreliable.

3.4 Data Reduction

The data collected during a run consist of a series of discrete pressure and volume values. The volume data denote a sine wave since the piston motion was sinusoidal while the pressure data denote some non-sinusoidal period function. Therefore, the pressure data were reduced to a Fourier series consisting of several sine or cosine amplitudes and phase shifts. The Fourier series representations of the volume and pressure data provide a measure of the gas spring heat transfer. Also, the Peclet number, a parameter which represents the operating conditions, was calculated.

When reducing the data, perfect gas relations were used for the thermodynamic properties of helium at room temperature and liquid nitrogen temperature ranges. For the helium temperature range, real gas properties were used (see appendix D for more details).

3.4.1 Peclet Number

The Peclet number is used as the nondimensional speed. This number relates the speed of the apparatus to the properties of the helium. The Peclet number may be written as follows:

$$Pe = \frac{U_p D}{\alpha} \tag{3.1}$$

where U = piston speed,

D = characteristic length, i.e. the cylinder diameter,

α = helium thermal diffusivity.

The piston speed, U_p, is an average speed given by:

$$U_p = \frac{2 \times (\text{stroke distance})}{(\text{period})}. \tag{3.2}$$

Since the helium properties vary over a cycle, we use an average value for the thermal diffusivity of the helium. Therefore since both the piston speed and the thermal diffusivity of the helium are average values, this Peclet number is also an average number.

The Peclet number nondimensionalizes the operating speed with the thermal properties of the helium. Equation 3.1 can be expanded into the following form:

$$Pe = \frac{U_p D \rho C_p}{k} \tag{3.3}$$

where ρ = density of the helium,

C_p = helium specific heat at constant pressure,

k = helium thermal conductivity.

This equation illustrates how these properties influence the Peclet number. In general, for a fixed velocity U_p, the larger the average helium density, ρ, the larger the Peclet number is. The density is influenced by both the pressure and temperature.

Tables 3.1 and 3.2 list the properties of helium at 1.0 atm. and 10.0 atm. which effect *Pe*. The tables show that between 25 K and 300 K the specific heat for helium is relatively constant. As the temperature decreases below 25 K, the specific heat sharply increases and then plunges (this is not fully evident in Table 3.1 because of its

limited temperature range). The point at which the hump occurs depends strongly on the critical point. For this reason, below 25 K, the specific heat can have a significant effect on the Peclet number. Over the whole temperature range, the thermal conductivity decreases as the temperature decreases. The Peclet numbers for experiment runs above 25 K, are primarily affected by the speed and helium density. Below 25 K, the specific heat and the thermal conductivity also influence *Pe*.

Table 3.1 Properties of helium gas at 1 atmosphere.

Temperature	Pressure	Density	Specific Heat	Thermal Conductivity	
T	P	ρ	C_p	k	$\sqrt{\rho C_p k}$
(K)	(atm)	(kg/m^3)	(J/kg K)	(W/m K)	(J/s$^{1/2}$ m^2 K)
5	1	11.9	6799	0.0111	30.0
10	1	4.99	5413	0.0175	21.7
20	1	2.45	5249	0.0267	18.5
30	1	1.62	5217	0.0341	17.0
40	1	1.21	5206	0.0407	16.0
50	1	0.974	5201	0.0468	15.4
100	1	0.486	5195	0.0736	13.6
200	1	0.242	5193	0.1179	12.2
300	1	0.161	5193	0.1547	11.4

Table 3.2 Properties of helium gas at 10 atmospheres.

Temperature	Pressure	Density	Specific Heat	Thermal Conductivity	
T	P	ρ	C_p	k	$\sqrt{\rho C_p k}$
(K)	(atm)	(kg/m³)	(J/kg K)	(W/m K)	(J/s$^{1/2}$ m² K)
5	10	143.5	3756	0.0258	118
10	10	62.37	7633	0.0234	107
20	10	24.57	5734	0.0297	64.6
30	10	15.95	5428	0.0363	56.1
40	10	11.90	5322	0.0425	51.9
50	10	9.53	5273	0.0483	49.3
100	10	4.81	5209	0.0746	43.2
200	10	2.42	5195	0.1186	38.6
300	10	1.62	5193	0.1552	36.1

3.4.2 Harmonic Decomposition

With harmonic decomposition, the nonsinusoidal pressure wave is represented as a Fourier series. The gas spring heat transfer is easily calculated from this series and the volume wave already known to be sinusoidal (the method for calculating this heat transfer will be presented in chapter 5). Discrete Fourier transforms are used to calculate the Fourier series representation of the pressure function. For the volume ratio of this apparatus, approximately 1.75, two harmonics generally provide an excellent representation of the pressure wave. Figure 3.3 shows an adiabatic pressure wave for a volume ratio of 1.75 and also its representation by one, two and three harmonics.

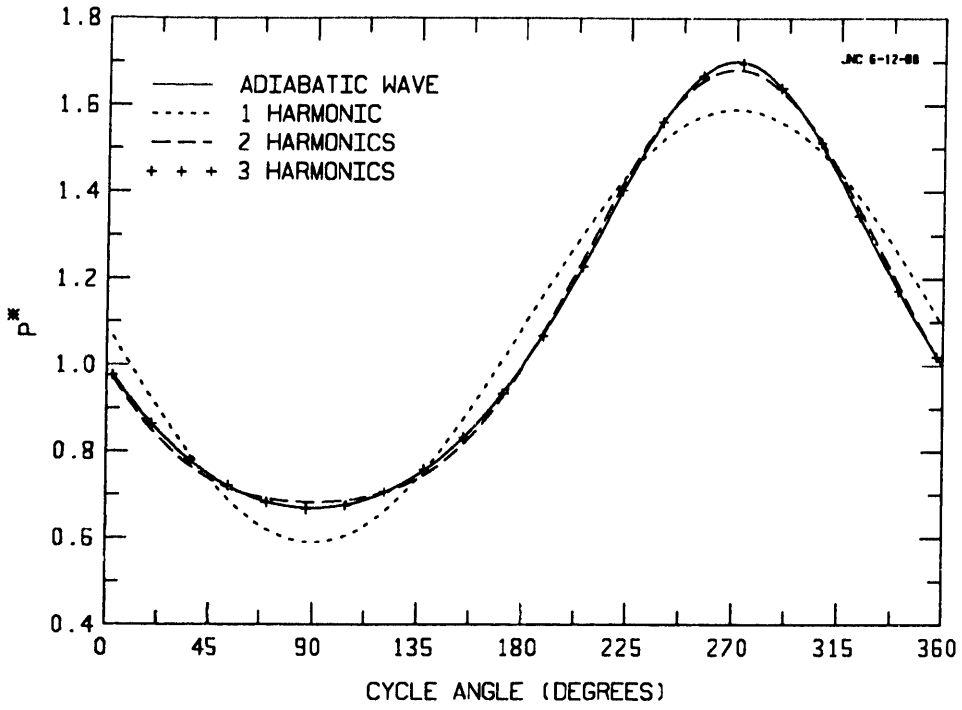

Fig. 3.3 Harmonic composition of a pressure wave (VR = 1.75, P = $P(t)/P(0)$).

The Fourier series representations of the pressure wave by two harmonics may be written in the following form:

$$P(t) = P_0 + P_1\cos(\omega t - \theta_1) + P_2\cos(2\omega t - 2\theta_2)). \tag{3.4}$$

where ω = angular velocity,

t = time,

$P(t)$ = actual pressure wave,

P_0, P_1, P_2 = pressure amplitude for indicated harmonic, and

θ_1, θ_2 = pressure phase angle for indicated harmonic.

In similar fashion, the volume wave may be written in the following form:

$$\upsilon(t) = \upsilon_0 + \upsilon_m\cos(\omega t - \phi) \tag{3.5}$$

where $\upsilon(t)$ = actual volume wave,

υ_0 = average volume,

υ_m = amplitude of the volume wave,

ϕ = phase angle of the volume wave,

For both the pressure and volume waves, two parameters, an amplitude and a phase, are associated with each harmonic. We compare different experiment runs by relating the pressure wave parameters to the equivalent parameters of reversible adiabatic and isothermal pressure waves.

The pressure amplitudes are related to the adiabatic and isothermal amplitudes by normalizing them. First the adiabatic and isothermal pressure waves are calculated using the reversible polytropic relations for a perfect gas as follows:

$$P_{poly}(t) = P_{ref}\left\{\frac{\upsilon_0}{\upsilon(t)}\right\}^z \tag{3.6}$$

where $P_{poly}(t)$ = polytropic pressure wave,

P_{ref} = reference pressure,

z = polytropic exponent.

When z equals 1, $P_{poly}(t)$ is the reversible isothermal pressure wave and when z equals γ, $P_{poly}(t)$ is the reversible adiabatic pressure wave. The reference pressure, P_{ref}, is the pressure in the apparatus when the volume, $\upsilon(t)$, equals υ_0 and $\upsilon(t)$ is increasing. Second, the adiabatic and isothermal pressure waves are decomposed into their harmonic components and each pressure amplitude is normalized by the following equation:

$$P_n^* = \frac{P_n - Pisot_n}{Pad_n - Pisot_n} \tag{3.7}$$

where P_n^* = normalized pressure amplitude,

$Pisot_n$ = isothermal pressure amplitude and

Pad_n = adiabatic pressure amplitude.

The subscript n, in the above equation denotes the n^{th} harmonic.

We decided to use equation 3.7 in normalizing the pressure amplitudes because this would show how closely the apparatus is operating to the adiabatic or isothermal conditions. The closer the normalized amplitude is to 1.0, the more adiabatically the apparatus is operating. Conversely, the closer the normalized amplitude is to 0.0, the more isothermally the apparatus is operating.

The adiabatic and isothermal pressure waves are always 180 degrees out of phase with the volume wave. The phases of the pressure wave, θ_1, θ_2 are related to the adiabatic and isothermal phases by taking the difference between them as follows:

$$\theta_n^* = (\pi + \phi - \theta_n)\frac{180}{\pi}. \tag{3.8}$$

The phase angle θ_n^* is given in units of degrees. The closer θ_n^* is to zero, the closer the apparatus is operating to a reversible process. It is not possible to distinguish by phase angle alone whether that process is adiabatic, isothermal.

3.5 Results

3.5.1 Individual Runs

Figures 3.4, 3.5, and 3.6 show the data collected for three typical experiment runs, one for each temperature range. This data is presented in dimensional form. The figures show the measured pressure and volume waves plotted as functions of time as well as the pressure waves plotted as functions of volume. In Figures 3.7, 3.8, and 3.9, the same pressure waves are plotted again versus time along with the calculated adiabatic and isothermal pressure waves. Comparing the pressure waves to the adiabatic and isothermal pressure waves provides a measure of the heat transfer occurring during a cycle.

For the pressure waves presented in these figures, the nondimensional speeds (Peclet numbers) as well as the normalized amplitudes and phases are given in Table 3.3.

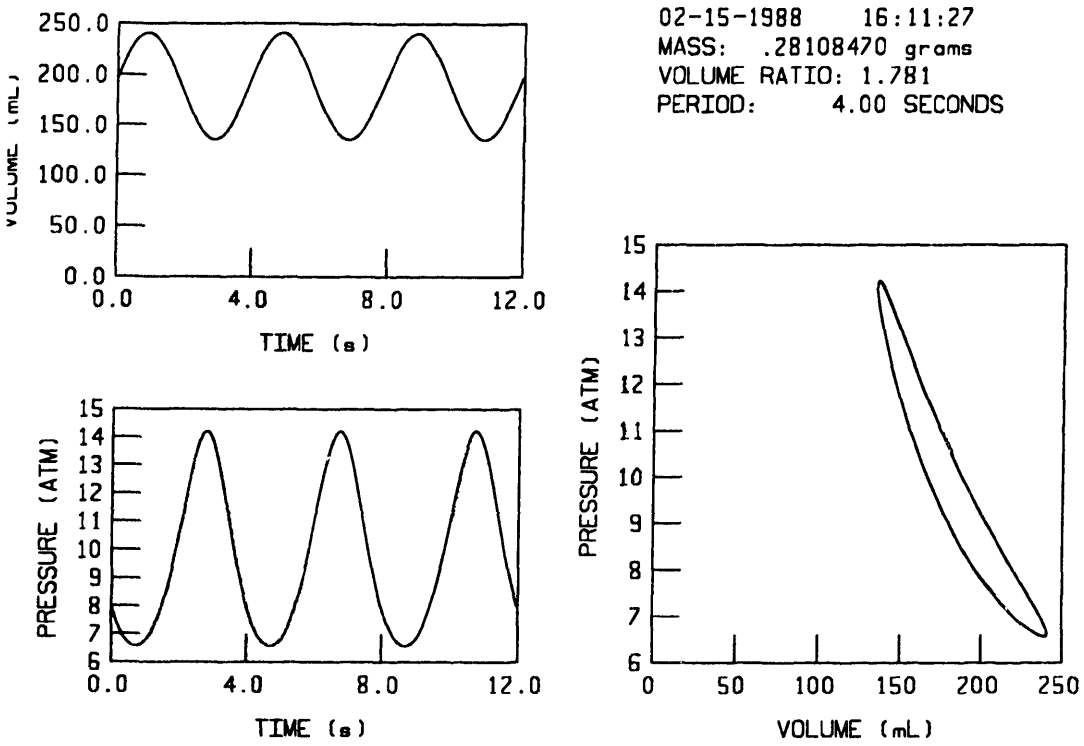

Fig. 3.4 Pressure and volume data for a room temperature experiment run.

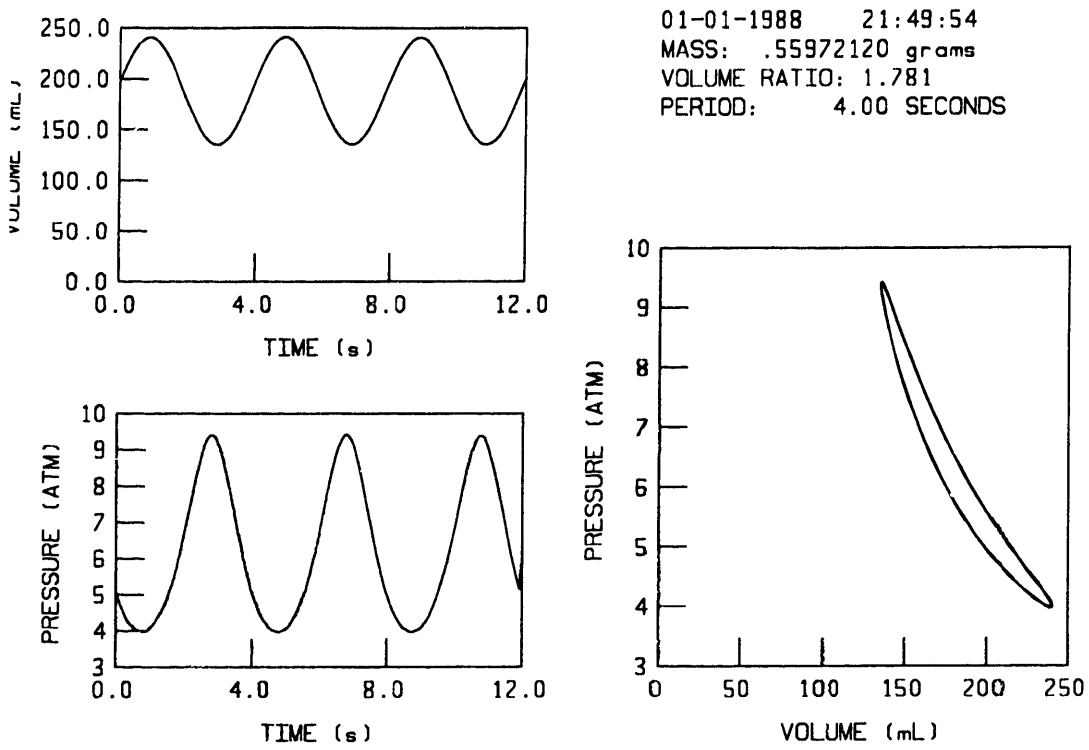

Fig. 3.5 Pressure and volume data for a liquid nitrogen temperature experiment run.

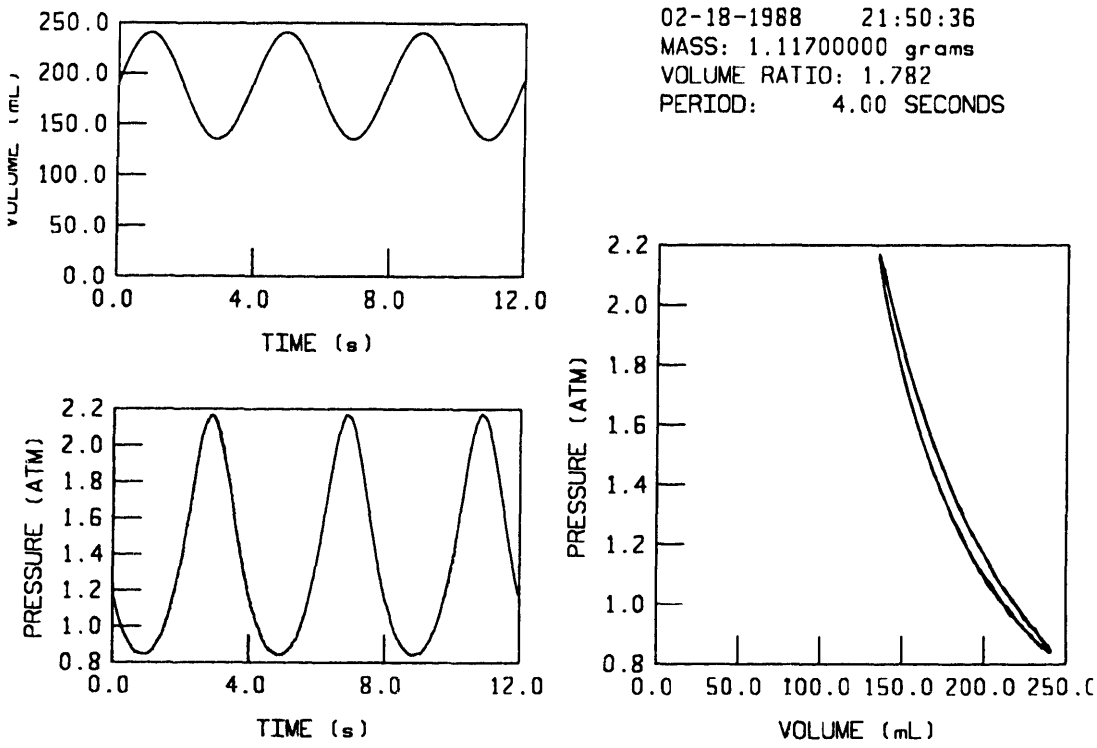

Fig. 3.6 Pressure and volume data for a liquid helium temperature experiment run.

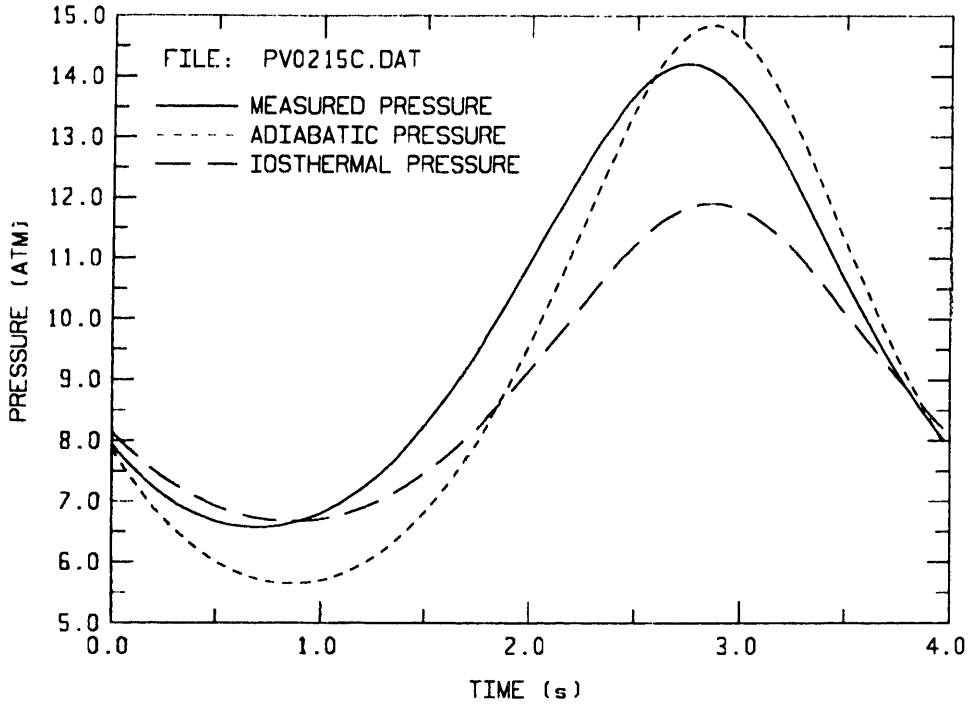

Fig. 3.7 Pressure wave for a room temperature experiment run.

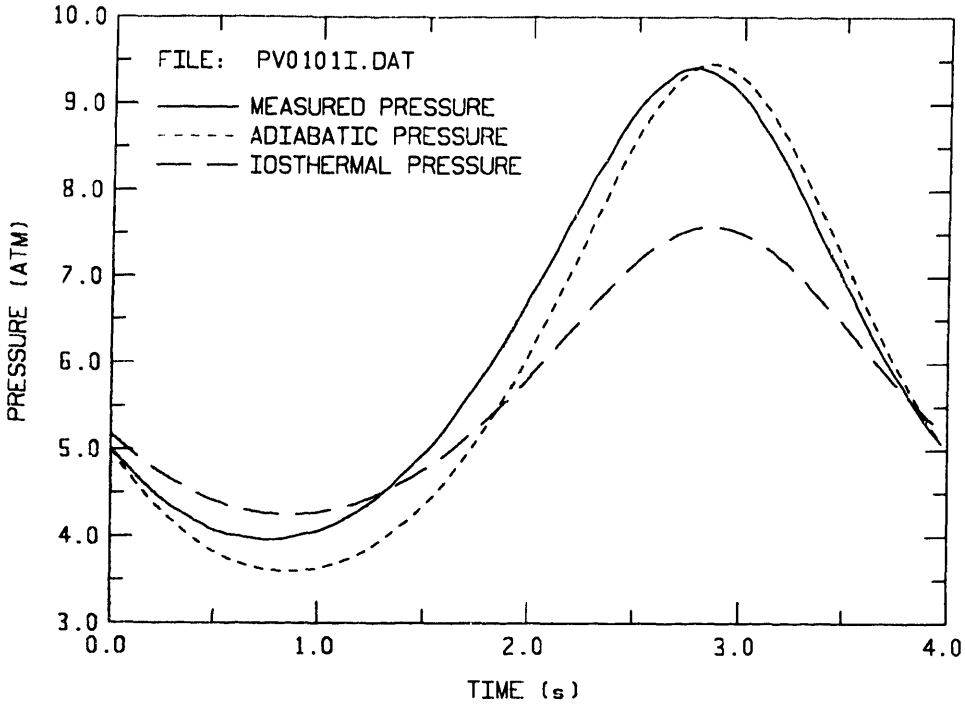

Fig. 3.8 Pressure wave for a liquid nitrogen temperature experiment run.

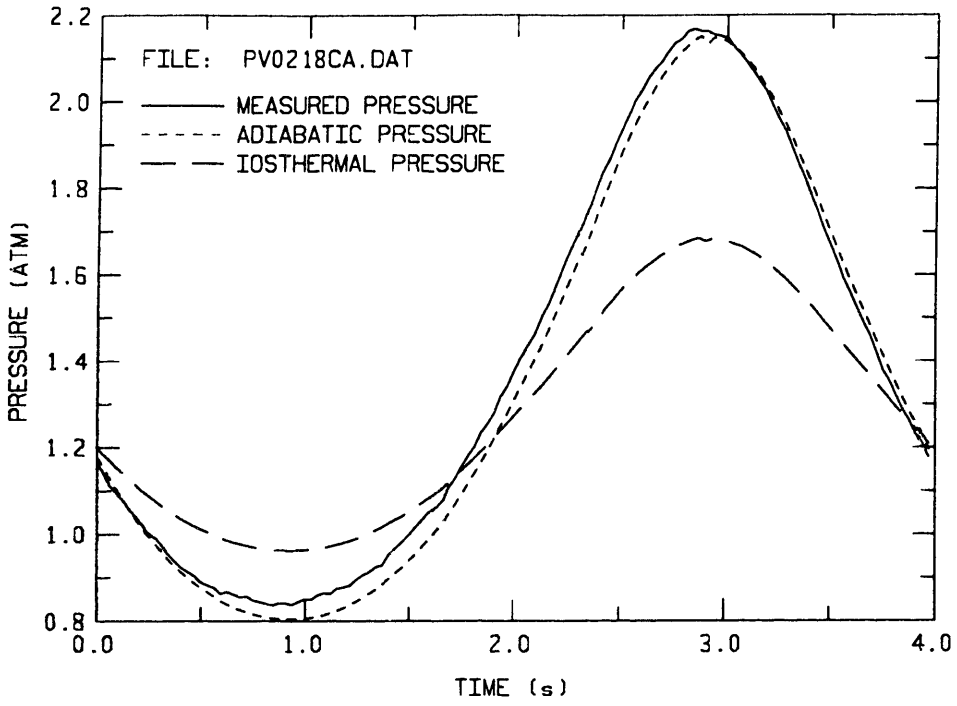

Fig. 3.9 Pressure wave for a liquid helium temperature experiment run.

Table 3.3 Normalized pressure parameters for individual experiment runs.

Run #	*Peclet*	P_1^*	P_2^*	θ_1^*	θ_2^*	plotted in Figure #
PV0215C.DAT (room temp.)	130	0.62	0.46	11.0	7.3	3.4, 3.7
PV0101I.DAT (liquid nitrogen temp.)	260	0.84	0.78	6.8	5.0	3.5, 3.8
PV0218CA.DAT (liquid helium temp.)	4550	0.94	0.91	2.7	1.7	3.6, 3.9

3.5.2 All Temperature Ranges

The normalized pressure amplitudes and phases versus *Pe* for all three temperature ranges are plotted in Figures 3.10 through 3.13. These data form a fairly smooth curve. The room temperature data span from near isothermal to near adiabatic operating conditions. The liquid nitrogen temperature data span from intermediate to adiabatic operating conditions. The helium temperature data stay close to adiabatic operating conditions.

3.5.2.1 Room Temperature Data Runs

In Figure 3.10, the room temperature first harmonic pressure amplitudes are plotted versus *Pe* and denoted by the squares. For small *Pe*, these pressure amplitudes are small indicating that the pressure waves, at least for this first harmonic, are similar to the corresponding isothermal pressure waves. For large *Pe*, the pressure waves behave more like adiabatic waves as can be seen from the high pressure amplitudes.

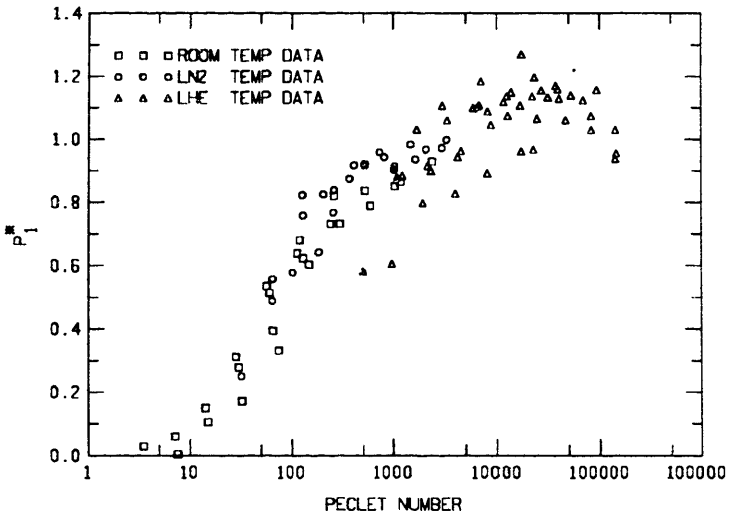

Fig. 3.10 Normalized first harmonic pressure amplitudes (see eqn. 3.7 for definition of P_1^*).

Figure 3.11 is a plot of the second harmonic pressure amplitude versus *Pe*, again the room temperature data is denoted by squares. In general, the room temperature data in this plot is similar to Figure 3.10 except that there is considerably more scatter. This scatter is due to the limitations in measuring and collecting data, the instrumentation has limited resolution and accuracy. Since the second harmonic amplitude, before it is nondimensionalized, is much smaller than the first harmonic amplitude (this is clearly demonstrated in Figure 3.3), the *noise* is more noticeable in Figure 3.11.

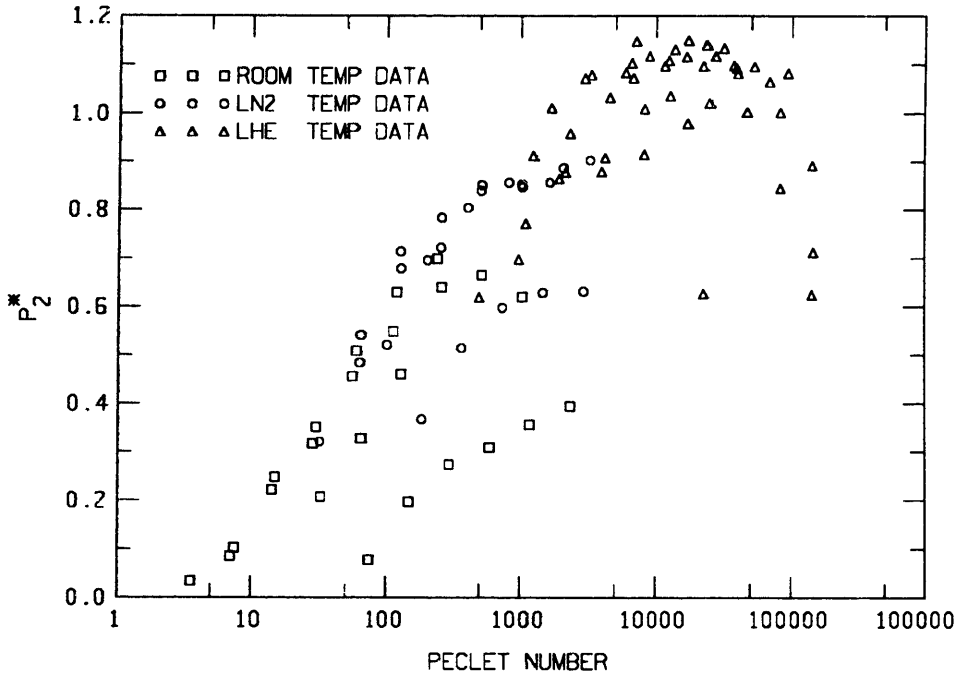

Fig. 3.11 Normalized second harmonic pressure amplitudes (see eqn. 3.7 for definition of P_2^*).

The phase shifts for the first and second harmonic pressure waves are presented in Figures 3.12 and 3.13. For the room temperature data, denoted by squares, these figures show that for small and large *Pe* the phase shifts are relatively small while at intermediate *Pe* the phase shifts reach a maximum. The pressure phases reiterate what the pressure amplitudes showed. For small *Pe*, the pressure waves behave isothermally as indicated by the small phase shift (the phase shift is zero for a truly isothermal pressure wave). For large *Pe*, the phase shifts again are small, accordingly the adiabatic phase shift is also zero.

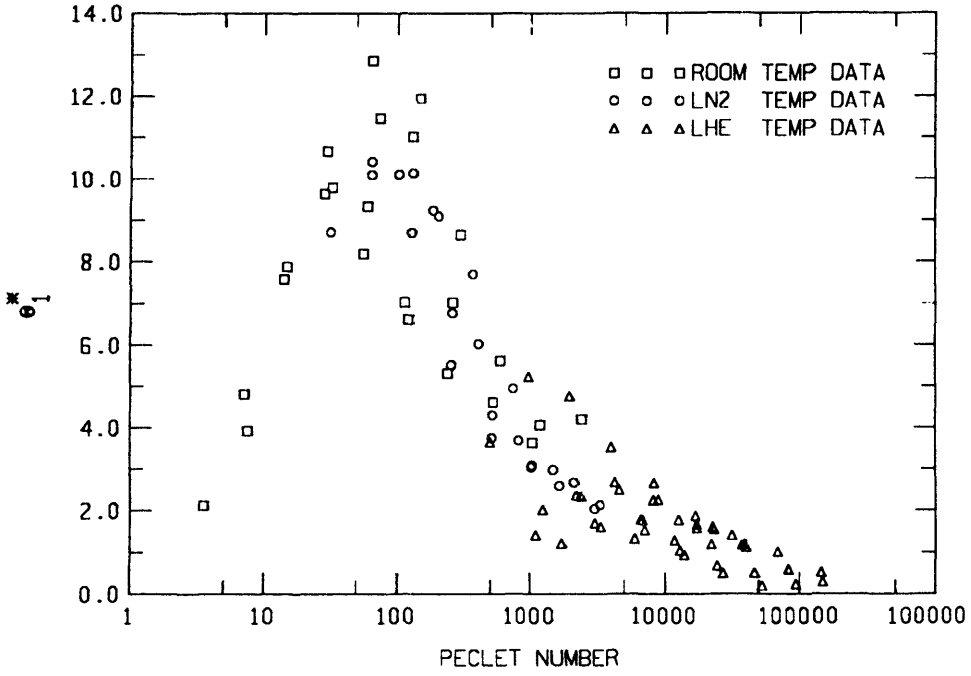

Fig. 3.12 Normalized first harmonic pressure phases (see eqn. 3.8 for definition of θ_1^*)

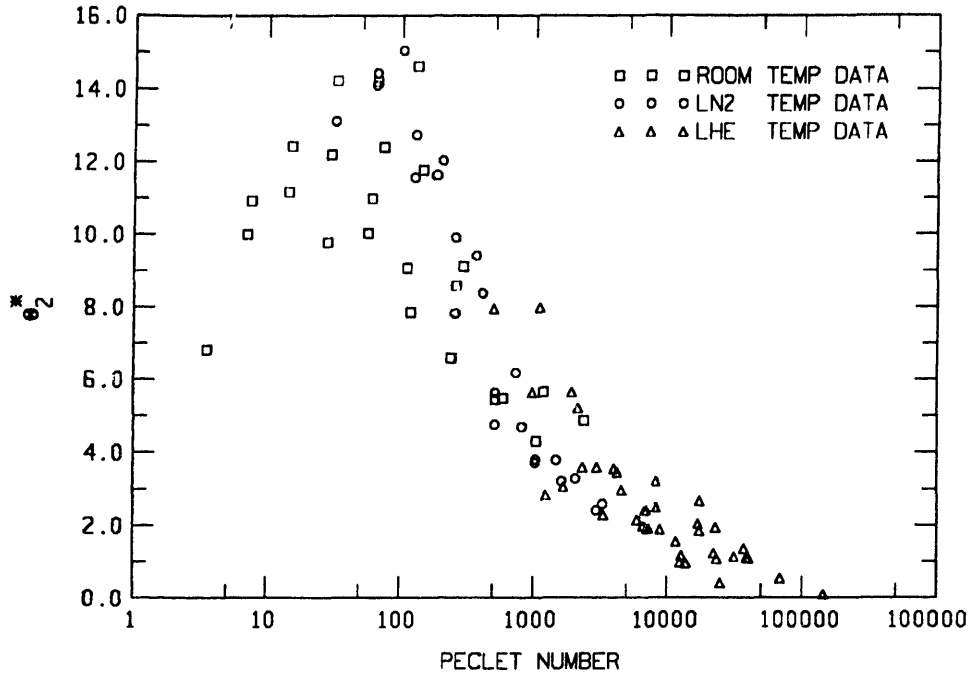

Fig. 3.13 Normalized second harmonic pressure phases (see eqn. 3.8 for definition of θ_2^*)

The phase shifts for the pressure give a general indication of how close an experiment run is to an ideal case. Both the adiabatic and isothermal pressure waves are ideal in the sense that there is no thermodynamic loss (or entropy generation) associated with them. For this apparatus operating at room temperature, low losses occur at the extreme values of *Pe*. Higher losses occur at intermediate values of *Pe*.

3.5.2.2 Liquid Nitrogen Temperature Data Runs

Figures 3.10 though 3.13 also show the pressure amplitudes and phases versus *Pe* for experiment runs performed at liquid nitrogen temperature. In these figures, the liquid nitrogen temperature data is denoted by the circles. This data resembles the room temperature data with the exception that the lower temperatures create higher helium densities and this makes the range of *Pe* higher. Also the apparatus does not operate as closely to isothermal conditions as some of the

room temperature runs. This is shown by the pressure amplitudes which do not approach zero and by the pressure phases which at the smallest *Pe*, also does not approach zero.

3.5.2.3 Liquid Helium Temperature Data Runs

The pressure amplitudes and phases for the cold temperature runs are denoted in Figures 3.10 through 3.13 by the triangles. These amplitudes and phases indicate adiabatic operating conditions. The amplitudes are all quite large, some of them even exceed 1.0. A reasonable explanation of how this is possible will be discussed in the Gap-Flow section of Chapter 4. The phase shifts are all small, also indicating adiabatic conditions.

The average temperatures for the individual runs range from 7 K to 25 K. These temperatures have a significant effect on the density of helium for a given pressure. Since *Pe* is strongly dependent on gas density, these low operating temperatures cause *Pe* to be consistently high.

Chapter 4
Models

This chapter presents several models of gas spring heat transfer. These models were developed to represent the phenomenon with the least amount of complexity while still retaining a high degree of accuracy.

Other researchers have studied gas spring heat transfer. Many of them emphasized a highly accurate but more complex representation of this heat transfer, while using quite crude assumptions of the effect of this heat transfer on the thermodynamics of the system. For example, Pfriem[13] models the heat transfer as conduction through a continuum, but oversimplifies the thermodynamic modeling of the gas. For this reason, his analysis cannot be applied to cases where there are large changes in pressure during a cycle. Another researcher, Keck,[14] uses ideal gas thermodynamic models for the gas but neglects to provide the interaction between the heat transfer and the thermodynamics.

In the models presented in this chapter, the gas volume, having a sinusoidal time function is the controlling input to the system. This makes the models particularly useful because the volume is the controllable input for real machines. Also the heat transfer and thermodynamics involved are represented in a simple and accurate manner. Specific attention is paid to factors which affect cryogenic equipment, particularly, the geometry of these machines and the temperature-dependent material properties of both the working fluid and cylinder.

4.1 One-Space Heat Transfer Model

The one-space heat transfer model is the simplest model presented in this study. This model represents the gas in the cylinder as single lumped mass as opposed to a continuum. There is only a single temperature and pressure associated with the gas as opposed to a continuous distribution. The piston compresses and expands the gas while the gas exchanges heat with the cylinder walls. Figures 4.1a and 4.1b are schematics of the model. Figure 4.1a is a typical thermodynamic system diagram showing the gas in the

---

13 Pfriem,H., *Periodic Heat Transfer at Small Pressure Fluctuations*, National Advisory Committee for Aeronautics, Technical Memorandum No. 1048, September, 1943.

14 Keck, James C., "Thermal Boundary Layer In a Gas Subject to a Time Dependent Pressure", *Letters in Heat and Mass Transfer*, Pergamon Press Ltd., vol. 8, 1981, pp. 313-319.

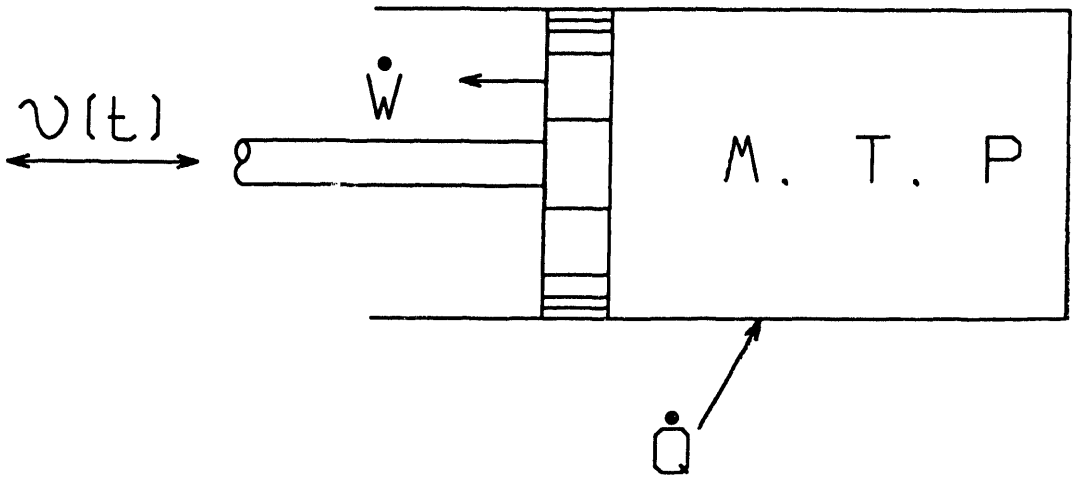

Fig. 4.1a One-space model thermodynamic system diagram.

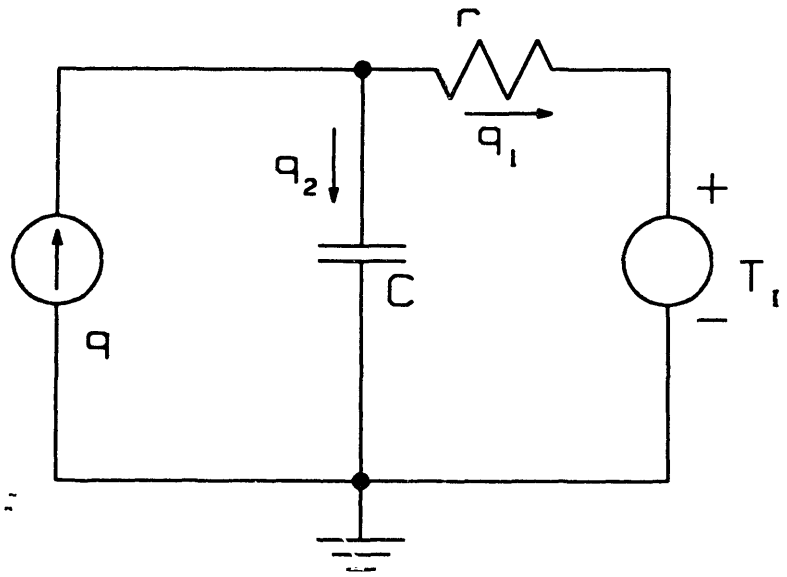

Fig. 4.1b One-space model electrical analog.

cylinder and the energy interactions between the gas and the cylinder. Figure 4.1b is an electric analog which uses a capacitor, C, to represent the mass in the cylinder, a resistor, r, to represent the thermal resistance between the gas and the cylinder wall, and a current source, q, to represent the work input to the gas. The voltage source, T_l, represents the operating temperature of the system and plays a crucial role in evaluating thermodynamic properties.

4.1.1 Development of the One-Space Model

For the analysis of this model the following assumptions are made:

1. The gas obeys the perfect gas law;
2. The temperature and pressure throughout the gas is uniform at all times;
3. The temperature of the cylinder wall is constant and uniform.

The input for this model is the cylinder volume. Typically it is given as a sinusoidal function of time,

$$\upsilon(t) = \upsilon_l + \upsilon_m \sin(\omega t) \tag{4.1}$$

where υ = total volume of gas,

υ_l = time-averaged volume,

υ_m = amplitude of variation of total volume,

ω = angular velocity,

t = time.

The work that the piston applies to the gas is written as follows:

$$q = -P\frac{d\upsilon}{dt} \tag{4.2}$$

where q = work input to the gas,

P = absolute gas pressure.

The work depends on both the pressure and volume of the gas. The volume is the specified input whereas the pressure is a result of the analysis.

The heat transfer which occurs between the gas and the cylinder wall is given by the following equation:

$$q_1 = \frac{T - T_l}{r} \tag{4.3}$$

where q_l = heat transfer to the cylinder wall,

T = temperature of the gas,

T_l = temperature of cylinder wall,

r = thermal resistance.

The rate at which energy is stored in the gas is:

$$q_2 = C\frac{dT}{dt} \tag{4.4}$$

where q_2 = rate of energy storage in the gas and

C = thermal capacitance of the gas.

The thermal capacitance of the gas is:

$$C = C_v M \tag{4.5}$$

where C_v = specific heat of the gas at constant volume,

M = total mass of the gas.

Equations 4.2 through 4.4 are derived using the electrical analog of figure 4.1b. However an equivalent set of equations can be derived using the thermodynamic schematic of figure 4.1a. These equations are as follows:

$$\dot{W} = P\frac{d\upsilon}{dt} = -q \tag{4.2a}$$

where $\dot{W}$ = rate of work done by the gas

$$\dot{Q} = -\frac{T - T_l}{r} = -q_1 \tag{4.3a}$$

where $\dot{Q}$ = heat transfer rate to the gas

$$\frac{dU}{dt} = C_v M\frac{dT}{dt} = q_2 \tag{4.4a}$$

where U = total internal energy of the gas.

Each of these equivalent derivations are used later as the basis for more complex models.

4.1.2 Governing Equation

Referring to figure 4.1a and using the first law of thermodynamics for the gas yields:

$$\dot{Q}-\dot{W}=\frac{dU}{dt}. \tag{4.6}$$

Substituting the relations of 4.2a through 4.4a into equation 4.6 gives the following:

$$\frac{-(T-T_l)}{r}-P\frac{d\upsilon}{dt}=C\frac{dT}{dt} \tag{4.7}$$

Using the perfect gas law,

$$P\upsilon=MRT, \tag{4.8}$$

and defining the ratio of specific heats as follows,

$$\gamma=\frac{C_p}{C_v}, \tag{4.9}$$

equation 4.7 can be reduced to the following form:

$$\frac{dP}{dt}=\frac{1}{r}\left\{\frac{(\gamma-1)T_l}{\upsilon}-\frac{P}{MC_v}\right\}-\frac{d\upsilon}{dt}\frac{\gamma P}{\upsilon}. \tag{4.10}$$

This differential equation relates the pressure of the gas to the volume function. It takes into account the conduction heat transfer between the gas and the cylinder wall. This equation is nondimensionalized in the next section.

4.1.3 Nondimensionalization

To nondimensionalize equation 4.10, the following reference state is introduced. The temperature and pressure at time $t = 0$, are defined to be T_l and P_l. The volume at $t = 0$ is υ_l as indicated by equation 4.1. In addition, the following nondimensional parameters and variables are defined:

$$P^*=\frac{P}{P_l} \tag{4.11}$$

$$t^*=\omega t \tag{4.12}$$

$$VR=\frac{\upsilon_l+\upsilon_m}{\upsilon_l-\upsilon_m} \tag{4.13}$$

$$\tau = \omega r M C_v \tag{4.14}$$

P^* is the nondimensional pressure and t^* is the nondimensional time. The parameter VR is the volume ratio of the cylinder and the parameter τ is a nondimensional speed that relates heat transfer rate to piston speed.

By substituting equations 4.1 and 4.11 through 4.14 into equation 4.10 we get the following nondimensional equation for pressure as a function of time:

$$\frac{dP^*}{dt^*} = \frac{1}{\tau}\left\{\frac{1}{1+\frac{VR-1}{VR+1}\sin(t^*)} - P^*\right\} - \gamma P^* \frac{\cos(t^*)}{\frac{VR+1}{VR-1}+\sin(t^*)} \tag{4.15}$$

4.1.4 Results

The nondimensional pressure specified in equation 4.15 is found through numerical integration methods (see Appendix E for a more complete discussion of the methods used and a listing of the computer code used). This equation is integrated over six full cycles and the sixth cycle is shown in figure 4.2. For this figure, the volume ratio, VR, was taken to be 1.75 since this is close to the volume ratio of the experiment apparatus. Pressure waves are plotted for τ values of 0.1, 1.0, and 10.0. Adiabatic and isothermal pressure waves are also plotted, these waves are calculated using the reference state defined by P_l and υ_l presented earlier. This figure shows that for τ equal to 0.1 the pressure wave is almost isothermal. For τ = 10.0 the pressure wave is almost adiabatic. For $\tau \rightarrow 0.0$, equation 4.15 reverts to the perfect gas equation for an isothermal wave. Similarly, for $\tau \rightarrow \infty$, this equation reduces to the adiabatic relation for a perfect gas. When this model is solved for intermediate values of τ, the values of the pressure amplitudes are between those of the ideal pressure waves. Also, the pressure wave leads the two ideal waves.

The same technique used for the reduction of the experiment data can be applied to pressure waves generated by this model. Pressure waves are generated for τ values which range from 0.01 to 100.0 and also adiabatic and isothermal pressure waves are calculated for each case. The pressure waves generated by the model are decomposed into their Fourier series and the amplitudes and phases are normalized using equations 3.7 and 3.8.

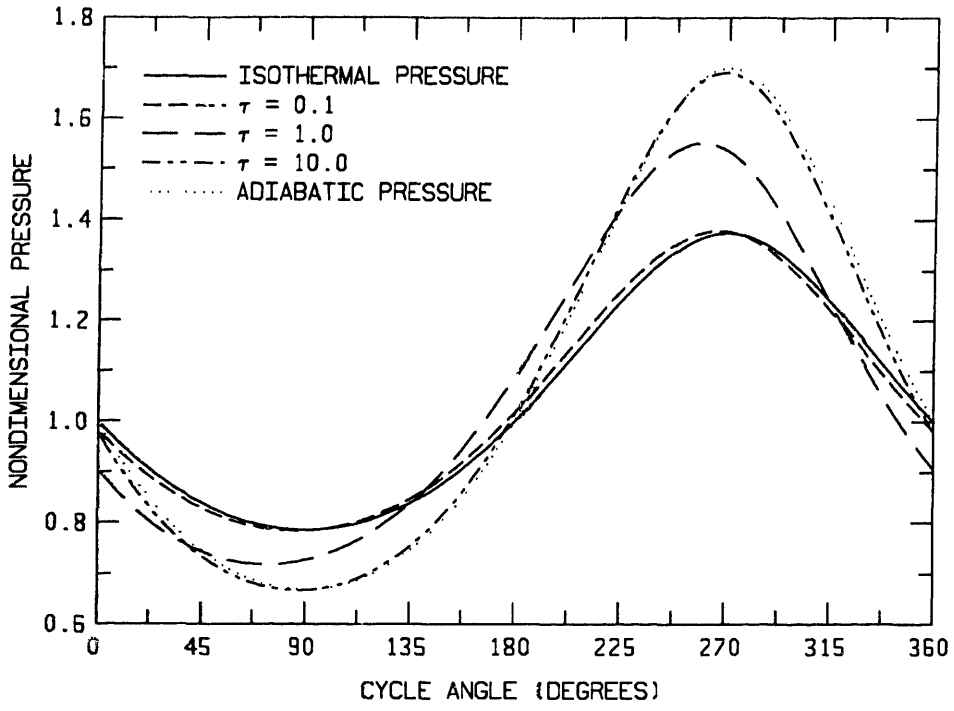

Fig. 4.2 One-space model pressure waves.

Figures 4.3 and 4.4 show the normalized first and second harmonic pressure amplitudes plotted against τ. As τ approaches 0.0 from 0.1, the amplitudes approach 0.0, indicating that the heat transfer is approaching isothermal. As τ increases from 0.1 to 10.0, the pressure amplitudes increase from nearly 0.0 to almost 1.0. Thus the heat transfer changes from isothermal to adiabatic in this intermediate range of τ. For values of τ greater than 10.0, the pressure amplitudes approach 1.0 asymptotically.

Figures 4.5 and 4.6 show the normalized first and second harmonic pressure phases plotted against τ. These phases peak at τ approximately equal to 1.0. As τ increases or decreases from this point, the phases decrease monotonically.

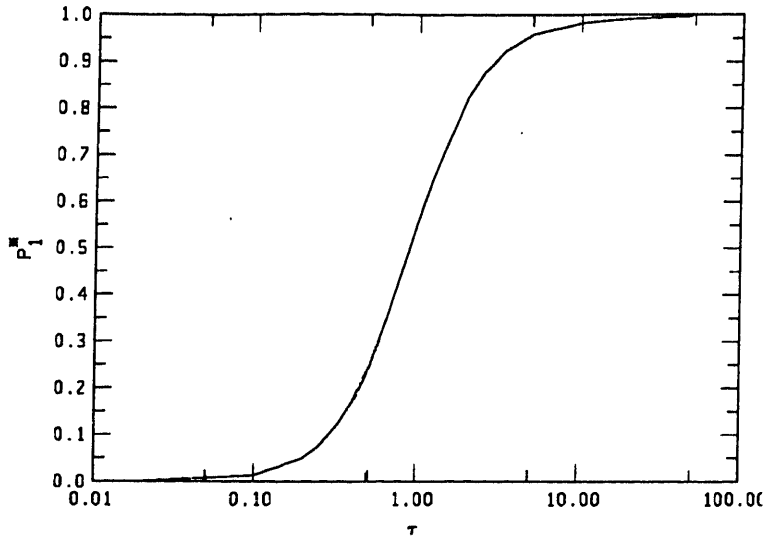

Fig. 4.3 One-space model normalized first harmonic pressure amplitudes.

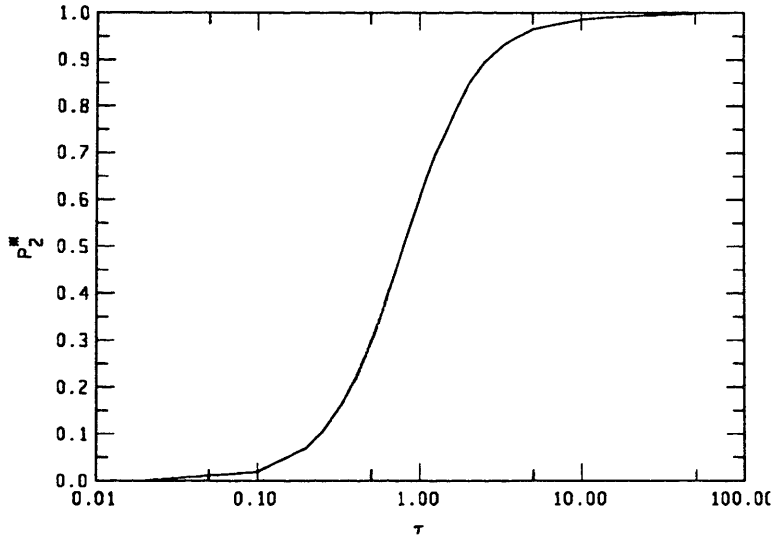

Fig. 4.4 One-space model normalized second harmonic pressure ampiitudes.

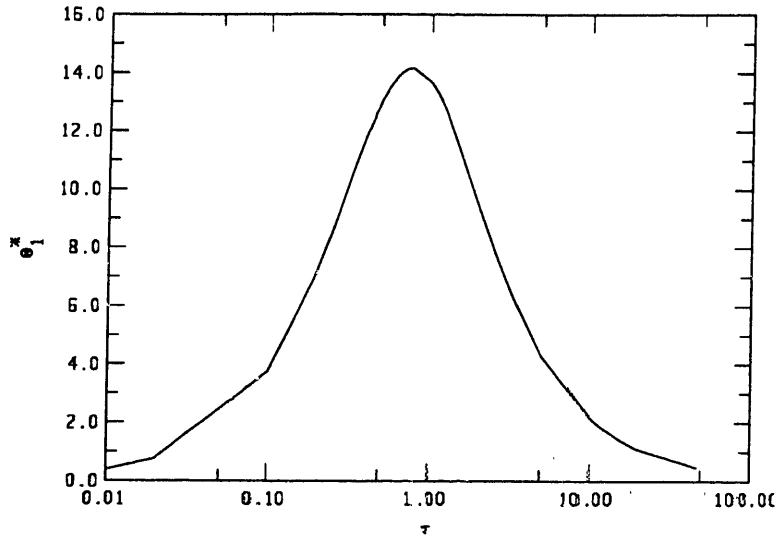

Fig. 4.5 One-space model normalized first harmonic pressure phases.

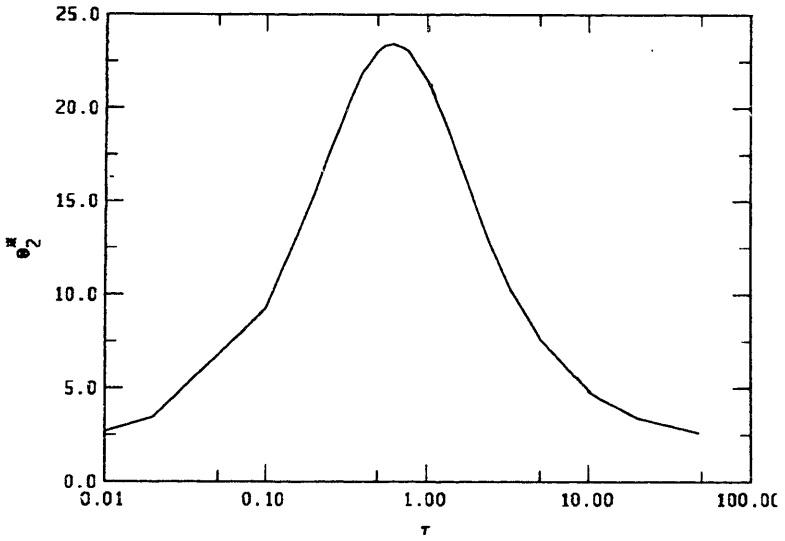

Fig. 4.6 One-space model normalized second harmonic pressure phases.

In order to quantitatively compare the results of this model with the experimental results, the model's parameters must be related to the characteristics of the experiment. For the parameters γ and VR, this is straightforward. γ is the ratio of specific heats which is 1.667 for helium at room temperature. VR is the volume ratio of the apparatus which is approximately 1.75. The parameter τ, defined by equation 4.14, is not so easily related to the experiment. However, direct relations can be easily found for all but one part of this parameter. ω is the angular velocity of the piston motion, M is the total mass of the gas in the cylinder, and C_v is the specific heat at constant volume for helium. The remaining parameter, r, is the thermal resistance between the cylinder wall and the working fluid. To help determine r, we relate it to the film coefficient used in Newton's law of cooling by the following equation:

$$r = \frac{1}{h_f A} \tag{4.16}$$

where h_f = film coefficient, and

A = heat transfer area.

For the following analysis, an average value for the the heat transfer area is used which is equal to the cylinder surface area when the piston is at mid-stroke. The film coefficient, h, may be related to the thermal conductivity, k, of the gas by assuming one dimensional conduction through its thermal boundary layer, where d = the depth of the thermal boundary layer. Thus

$$h_f = \frac{k}{d} \tag{4.17}$$

Now the thermal resistance can be related to the depth of the thermal boundary layer, the heat transfer area and the thermal conductivity as follows:

$$r = \frac{d}{kA} \tag{4.18}$$

To evaluate r, d must be determined. To do so we make the assumption that d is proportional to the thermal wavelength defined as follows:

$$d \propto \lambda = \left\{ \frac{8\pi^2 \alpha}{\omega} \right\}^{1/2} \tag{4.19}$$

λ is the length of a temperature wave induced by the periodic temperature variation of the surface of a semi-infinite wall. Carslaw and Jaeger[15] present a complete derivation of this equation.

With the above analysis, we are now able to write an expression for the model parameter τ. Combining equations 4.14, 4.18 and 4.19, we get the following relation for τ:

$$\tau \propto \frac{1}{\gamma} \frac{\upsilon_l}{A_{sur}} \left\{ \frac{\omega}{\alpha} \right\}^{1/2} \tag{4.20}$$

where α = thermal diffusivity,

A_{sur} = surface area of the cylinder at mid-stroke.

Using equation 3.1, τ can be related to the Peclet number as follows:

$$\tau \propto \frac{1}{\gamma} \frac{\upsilon_l}{A_{sur}\{(stroke)D\}^{1/2}} \sqrt{Pe} \tag{4.21}$$

Figures 4.7 and 4.8 compare the one-space model with the experiment. We chose the proportionality constant indicated by equation 4.21 by empirically matching the results of the model with the results of the experiments. Figure 4.7 is a plot of the first harmonic pressure amplitudes for both the experiment data and the model. The speed parameter plotted along the x-axis is τ for the one-space model and $0.11(Pe)^{1/2}$ for the experiment data (note that 0.11 is the proportionality constant from equation 4.21). This plot illustrates that the assumption that the depth of the thermal boundary layer is proportional to λ provided the correct relation between the model and the physics of the apparatus. This is indicated by the close resemblance between the model curve and the experiment data, especially in the way that the slopes match in the transition region between adiabatic and isothermal conditions. Figure 4.8 is a plot of the first harmonic pressure phases for both the experiment data and the one-space model. The phase shifts predicted by the model are somewhat larger than those measured by the experiments. However, the general shape of the model phase curve is very similar to the experiment data. Also, the maximum phase shifts for both the model and the data coincide well.

---

15 Carslaw, H.S., Jaeger, J.C., *Conduction of Heat in Solids*, Oxford University Press, London, U.K., 1959, pg.66

This model reproduces many features of a gas spring confirmed by the experiment data. Not only does it demonstrate that at low speeds, the system behaves isothermally and at high speeds it behaves adiabatically, but it also predicts the transition from these cases with a good degree of accuracy.

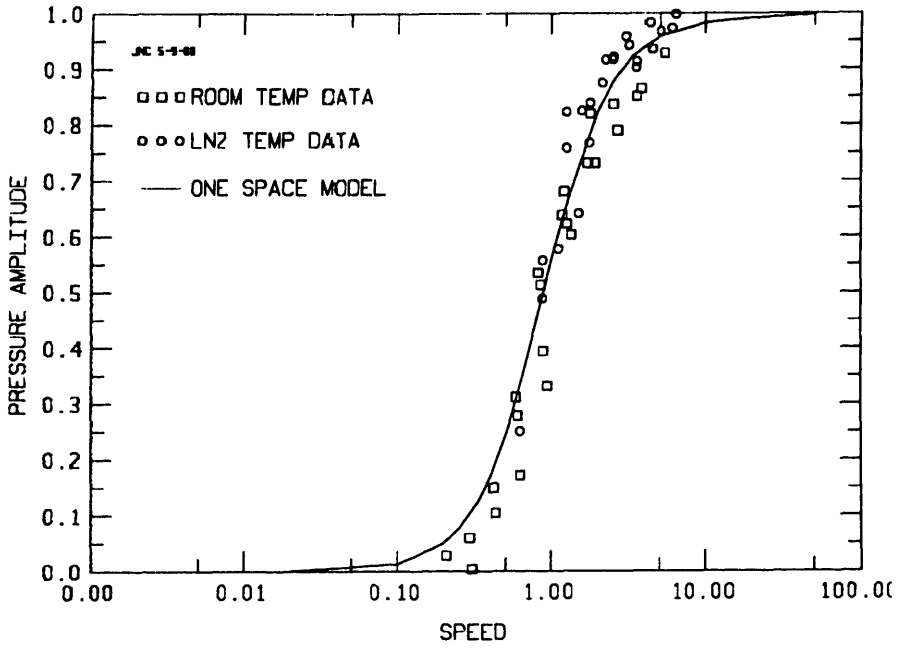

Fig. 4.7 First harmonic pressure amplitudes (normalized) for both the one-space model and experiment data
($SPEED = 0.11(Pe)^{1/2}$ for experiment data; $SPEED = \tau$ for model).

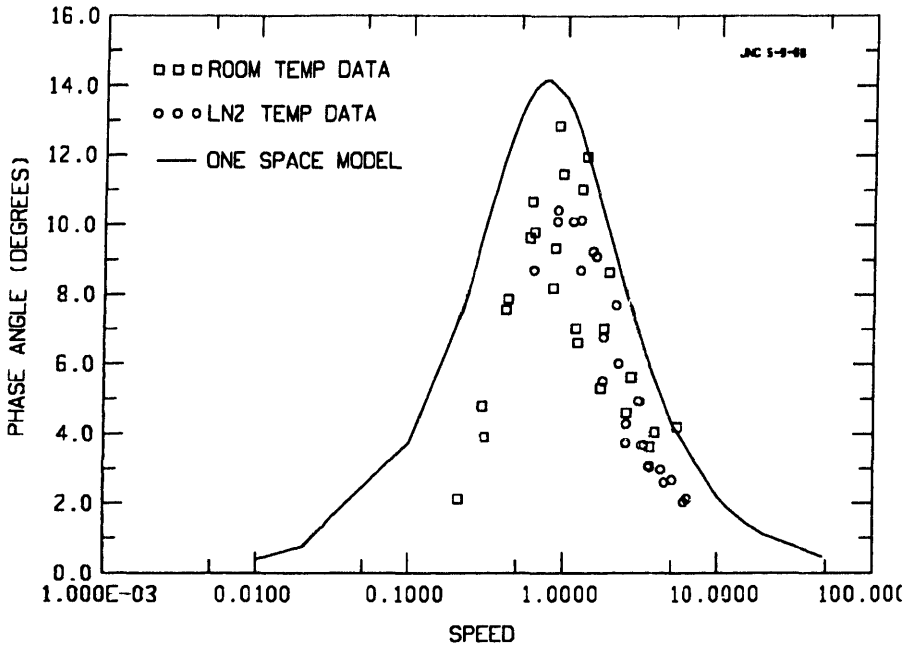

Fig. 4.8 First harmonic pressure phases for both the one-space model and experiment data
(*SPEED* = 0.11(*Pe*)$^{1/2}$ for experiment data; *SPEED* = τ for model).

4.2 Two-Space Heat Transfer Model

The two-space heat transfer model is an extension of the one-space model. Instead of representing the mass in the cylinder as just one lump, this model divides the mass into two regions. These two regions allow a more realistic representation of the heat transfer occurring throughout the gas.

Faulkner[16] presents a model similar to the two-space model presented in this section. However, there are two key differences between these models. The first difference is that Faulkner's model specifies the controlling input to the system as work, while the models in this chapter specify the controlling input as a volume wave. The second difference is that Faulkner's model neglects to specify the operating temperature, which is crucial in modeling the interaction between the heat transfer and the thermodynamics. By formulating the models as is done in this chapter, we are able account for this interaction by using a more realistic controlling input (i.e. the volume wave).

4.2.1 Development of the Two-Space Model

Figure 4.9 is an electrical analog of this model. The gas in the cylinder is divided into two regions represented by node 1 and node 2. Node 1 represents the gas located near the cylinder wall. Its thermal mass (total heat capacity) is denoted by C_1 This region of gas exchanges heat with the cylinder wall and also with the second region of gas represented by node 2. This second region of gas, whose thermal mass is denoted by C_2, is located in the center of the cylinder. The gas at node 2 exchanges heat only with the gas at node 1. The temperature of each region is assumed to be uniform in space, but can vary cyclically with time. The thermal resistance between the regions of gas is denoted by r_2 and the thermal resistance between node 1 and the cylinder wall is denoted by r_1. The temperature of the cylinder wall is constant and uniform and is represented by the potential T_1. Input energy to the system is shown as q_A and q_B. This energy is divided proportionally between the two thermal masses. This energy is actually supplied to the system in the form of work.

---

16 Faulkner, H.B., *An Investigation of Instantaneous Heat Transfer During Compression and Expansion In Reciprocating Gas Handling Equipment*, Ph.D. thesis, Department of Mechanical Engineering, M.I.T., May, 1983.

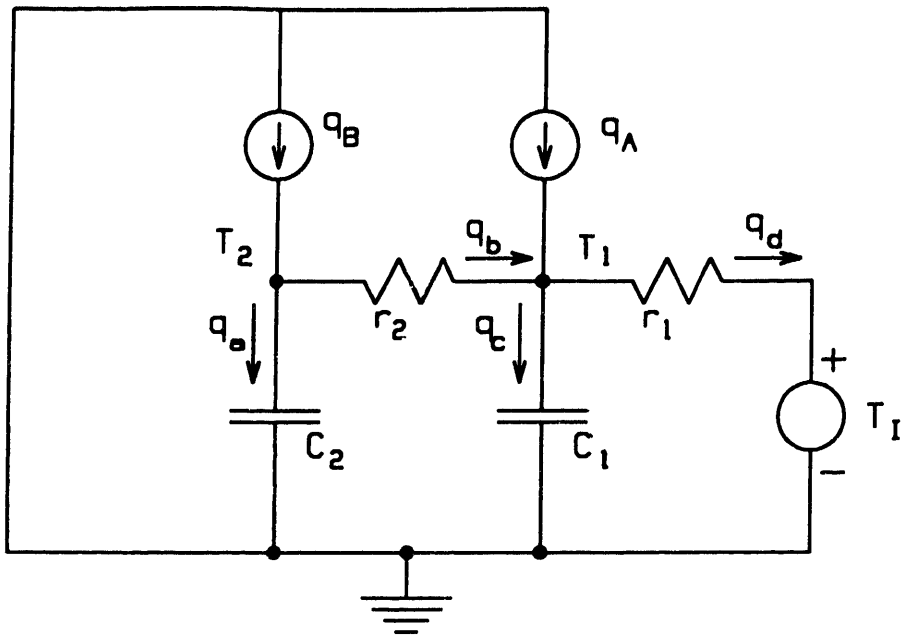

Fig. 4.9 Two-space model electrical analog.

For the development of this model, we make the following assumptions:

1. The gas obeys the perfect gas law;
2. The temperature and pressure is uniform in each gas region at all times;
3. The temperature of the cylinder wall is constant and uniform.

As in the one-space model, the controlling input is a cylinder volume wave. The total work input to the gas is:

$$q_A + q_B = -P\frac{d\upsilon}{dt} \tag{4.22}$$

The pressure at node 1 equals the pressure at node 2 at all times. The work input is divided into two parts. Some of the work goes towards compressing the gas at node 1, while the rest goes towards compressing the gas at node 2. The equation describing this distribution of work is assumed to be

$$\frac{q_A}{C_1} = \frac{q_B}{C_2} \tag{4.23}$$

In this equation, the work is divided proportionally between the regions of gas. If C_1 and C_2 have the same magnitude, then each region of gas will each get equal amounts of work. If one is larger than the other, that region will get a proportionally larger share. Since C_1 and C_2 remain constant over a cycle, equation 4.23 implies that the division of work also remains constant over a complete cycle. Although this assumption greatly reduces the complexity of the governing equations, it does introduce some error. A more rigorous expression that accounts for the varying compressibilities of the two regions of gas and thus allows for a varying distribution of work input is:

$$\frac{q_A}{C_1 d\left(\frac{T_1}{P}\right)} = \frac{q_B}{C_2 d\left(\frac{T_2}{P}\right)} \tag{4.23a}$$

We use equation 4.23a to estimate a more accurate work input distribution for every solution of the governing equations of this model. We estimate that for the cycle that produced the greatest error, when using the more simple expression of equation 4.23, the maximum error in the distribution of work input occurring during that cycle is less

than 15 percent of the root mean squared average rate of work input. We emphasize that even for this worst case cycle, the error averaged over the cycle is much less than 15 percent.

The heat transfer which occurs between the gas and the cylinder wall is:

$$q_d = \frac{T_1 - T_0}{r_1} \tag{4.24}$$

and the heat transfer which occurs between the two regions of gas is:

$$q_b = \frac{T_2 - T_1}{r_2} \tag{4.25}$$

The rate at which energy is stored in each region of gas is presented in the following two equations:

$$q_c = C_1 \frac{dT_1}{dt} \tag{4.26}$$

$$q_a = C_2 \frac{dT_2}{dt} \tag{4.27}$$

4.2.2 Governing Equations

Referring to figure 4.9 and applying the first law of thermodynamics to the two gas regions yields the following equations:

$$q_A + q_b = q_c + q_d \tag{4.28}$$

$$q_B = q_a + q_b \tag{4.29}$$

Equation 4.28 is for the gas at node 1 which has a temperature T_1, while equation 4.29 is for the gas at node 2 which has a temperature T_2.

The perfect gas law for the two regions of gas is

$$P\upsilon = R(m_1 T_1 + m_2 T_2). \tag{4.30}$$

This equation along with equations 4.22 through 4.27 are used to expand equations 4.28 and 4.29 into the following system of equations:

$$\frac{dT_1}{dt} = -\left\{\frac{d\upsilon}{\upsilon}\frac{m_1 T_1 + m_2 T_2}{C_1 + C_2}\right\} + \left\{\frac{T_2 - T_1}{r_2 C_1}\right\} - \left\{\frac{T_1 - T_I}{r_1 C_1}\right\} \tag{4.31}$$

$$\frac{dT_2}{dt} = -\left\{\frac{d\upsilon}{\upsilon}\frac{m_1T_1 + m_2T_2}{C_1 + C_2}\right\} - \left\{\frac{T_2 - T_1}{r_2C_2}\right\} \tag{4.32}$$

These equations govern the temperatures of the two regions of gas.

4.2.3 Nondimensionalization

In this section, equations 4.31 and 4.32 are nondimensionalized. When solved, these equations yield nondimensional temperatures. The perfect gas law, in nondimensional form, can be used to solve for the nondimensional pressure of the two regions of gas.

To nondimensionalize the equations, the following reference state for the gas in both regions is introduced. At time $t = 0$, the pressure is defined as P_I, the temperature for both regions is T_I, and the volume in the cylinder is υ_I. The volume wave is specified to be sinusoidal and is given by equation 4.1. Also the following nondimensional parameters and variables are defined:

$$T_1^* = \frac{T_1}{T_I} \tag{4.33}$$

$$T_2^* = \frac{T_2}{T_I} \tag{4.34}$$

$$\mu = \frac{m_1}{m_1 + m_2} \tag{4.35}$$

$$\tau_1 = (m_1 + m_2)\omega r_1 C_v \tag{4.36}$$

$$\tau_2 = (m_1 + m_2)\omega r_2 C_v \tag{4.37}$$

T_1^* and T_2^* are nondimensional temperatures. μ is a parameter which represents the distribution of mass in the two gas regions. τ_1 and τ_2 are nondimensional speeds which relate cycle speed to the properties of the gas regions.

Using equations 4.33 through 4.35, equations 4.31 and 4.32 reduce to the following nondimensional system of equations:

$$\frac{dT_1^*}{dt^*} = \frac{-\cos(t^*)}{\frac{VR+1}{VR-1} + \sin(t^*)}\{\mu(\gamma-1)T_1^* + (1-\mu)(\gamma-1)T_2^*\} + \left\{\frac{T_2^* - T_1^*}{\tau_2\mu}\right\} - \left\{\frac{T_1^* - 1}{\tau_1\mu}\right\} \tag{4.38}$$

$$\frac{dT_2^*}{dt^*} = \frac{-\cos(t^*)}{\frac{VR+1}{VR-1} + \sin(t^*)} \{\mu(\gamma-1)T_1^* + (1-\mu)(\gamma-1)T_2^*\} - \left\{\frac{T_2^* - T_1^*}{\tau_2(1-\mu)}\right\} \tag{4.39}$$

The solution of this system of equations can be substituted into the the nondimensional perfect gas law as follows:

$$P^* = \frac{\mu T_1^* + (1-\mu)T_2^*}{\upsilon^*} \tag{4.40}$$

This allows us to calculate the pressure in the gas from the two temperature functions. Appendix E discusses the numerical methods used to solve these equations, also the computer code implementation of these solutions is listed there.

4.2.4 Results

Some solutions to the above equations are presented in this section. As in the one-space model, this model uses the parameters *VR* and γ as the volume ratio and the specific heat ratio. These values are fixed to match the experiment apparatus. This model also has three other parameters, τ_1, τ_2, and μ. All three of these parameters relate the operating speed to the gas properties.

Figure 4.9 shows that in this model the heat transfer to the cylinder wall is always in phase with the temperature T_1^*. Therefore it is possible for the heat transfer to the wall to be out of phase with the temperature T_2^*. If the heat transfer is in phase with T_1^* and out of phase with T_2^*, then it is also out of phase with the the mixed mean temperature. For this model, the mixed mean temperature may be written as follows:

$$T_{mm}^* = \frac{T_1^* m_1 + T_2^* m_2}{m_1 + m_2} \tag{4.41}$$

There are three important time constants for this model. These time constants may be written in terms of the nondimensional parameters as follows:

$$t_1 = \frac{1}{\mu\tau_1\omega} \tag{4.42}$$

$$t_2 = \frac{1}{(1-\mu)\tau_2\omega} \tag{4.43}$$

$$t_3 = \frac{1}{\mu\tau_2\omega} \tag{4.44}$$

For a significant phase shift to occur between the heat transfer and the temperature T_2^*, t_1 must be large and t_2 and t_3 must be small. This happens when the heat transfer rate from the gas at node 1 to the cylinder wall is high, and the heat transfer rate between the two gas regions is low. Figure 4.10a shows a solution to the system of equations 4.38, 4.39 when this is the case. For this figure μ equals 0.1, τ_1 equals 1.0, and τ_2 equals 10.0. The temperatures T_1^* and T_2^* are out of phase with each other. Since μ equals 0.1, 90 percent of the gas is at temperature T_2^*. For this reason, the mixed mean temperature is approximately equal to T_2^* and thus the heat transfer is also significantly out of phase with the mixed mean temperature. Figure 4.10b shows the nondimensional pressure for this solution. This pressure wave is nearly adiabatic and slightly out of phase with the ideal pressure waves. This is also the case in one-space model. The phase shift between the heat transfer and the mixed mean temperature does not show up on a pressure wave plot.

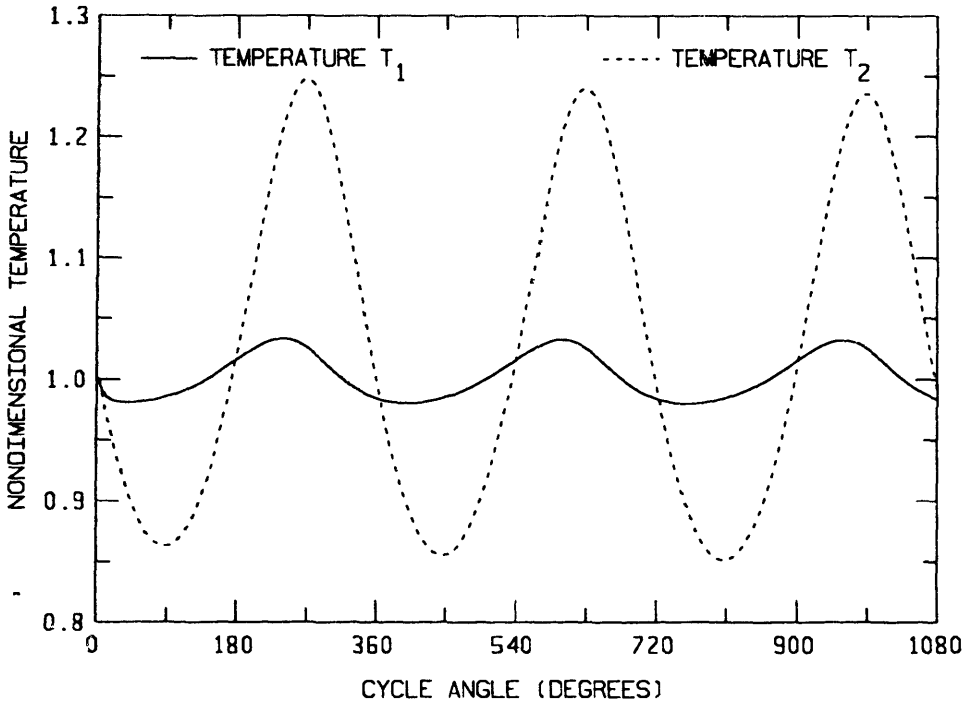

Fig. 4.10a Two-space model temperature waves (μ=0.1, τ_1=1.0, τ_2=10.0).

To more directly compare the two-space model to the experiment data, pressure waves generated by the model are harmonically decomposed and then normalized in the same way as the experiment data. These pressure waves are calculated for several parameter values chosen to closely represent the characteristics of the apparatus and the operating conditions of the experiment. The parameter γ is the ratio of specific heats for helium and the parameter *VR* is the volume ratio of the apparatus. The parameters μ, τ_1, and τ_2 do not have such an obvious connection to the apparatus. As in the one-space model, we make use of a thermal boundary layer in the gas near the cylinder wall to find relations for these parameters. Again we assume that the depth of the thermal boundary layer is proportional to the thermal wavelength for the gas.

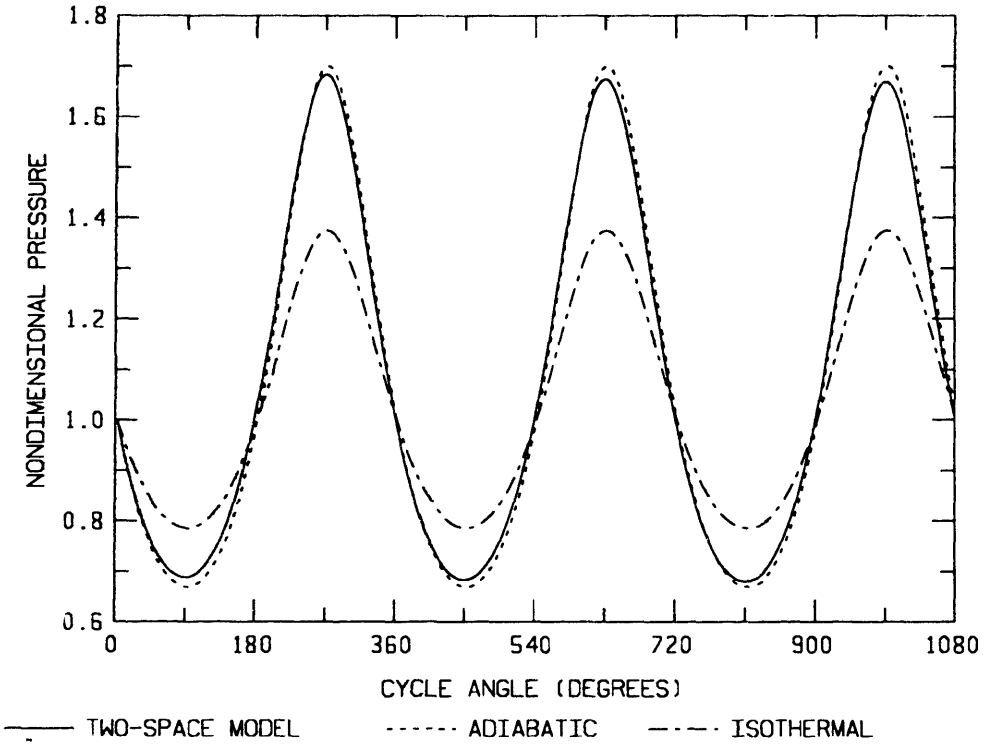

Fig. 4.10b Two-space model pressure waves (μ=0.1, τ_1=1.0, τ_2=10.0).

Referring to equations 4.19, 4.35 and 3.1, we write the following relations for μ:

$$\mu = \frac{m_1}{m_1 + m_2} \propto \frac{A_{sur}\lambda}{\upsilon_l} \propto \frac{A_{sur}}{\upsilon_l \sqrt{Pe}} \tag{4.45}$$

where m_l = the mass of helium within the thermal boundary layer.

We use average values to evaluate gas properties and cylinder volumes and areas.

Referring to equations 4.36 and 4.37, we can use the same type of analysis to conclude that τ_1 and τ_2 are proportional to the square root of the Peclet number as in the one-space model. Thus we get the following relation:

$$\tau_1 \propto \tau_2 \propto \frac{1}{\gamma A_{sur}\{(stroke)D\}^{1/2}} \frac{\upsilon_l}{\sqrt{Pe}} \tag{4.46}$$

μ, τ_1, and τ_2 have not been completely related to the experiment since the constants of proportionality have yet not been determined. We use the experiment data to empirically find these constants.

Figures 4.11 and 4.12 are plots of the first harmonic pressure amplitudes and phases for both the model and the experiment data. In these plots *SPEED* is equivalent to τ_2 for the model curve and $0.11(\mathrm{Pe})^{1/2}$ for the experiment data. The model parameters τ_1 and μ relate to *SPEED* by the following equations:

$$\tau_1 = 0.1 \times SPEED \tag{4.47}$$

$$\mu = \frac{0.1}{SPEED} \tag{4.48}$$

In figure 4.11 the shape of the first harmonic amplitude curve for the model is very similar in appearance to one-space model plot (figure 4.3). The amplitudes asymptotically approach 0.0 as the speed approaches zero and asymptotically approach 1.0 as the speed approaches infinity. In the region where the amplitudes change from 0.0 to 1.0, the slope of the model matches the slope in the data, considering the scatter present in the data. Figure 4.12 shows the first harmonic pressure phases. The model curve matches the data quite well. In both this model and the data the slope is steeper for speeds less than for speeds greater than 1.0. This asymmetry is not shown by the one-space model. In addition, the peak value of phase shift for the two-space model is somewhat less than that for the one-space model, the lower value matches the data better.

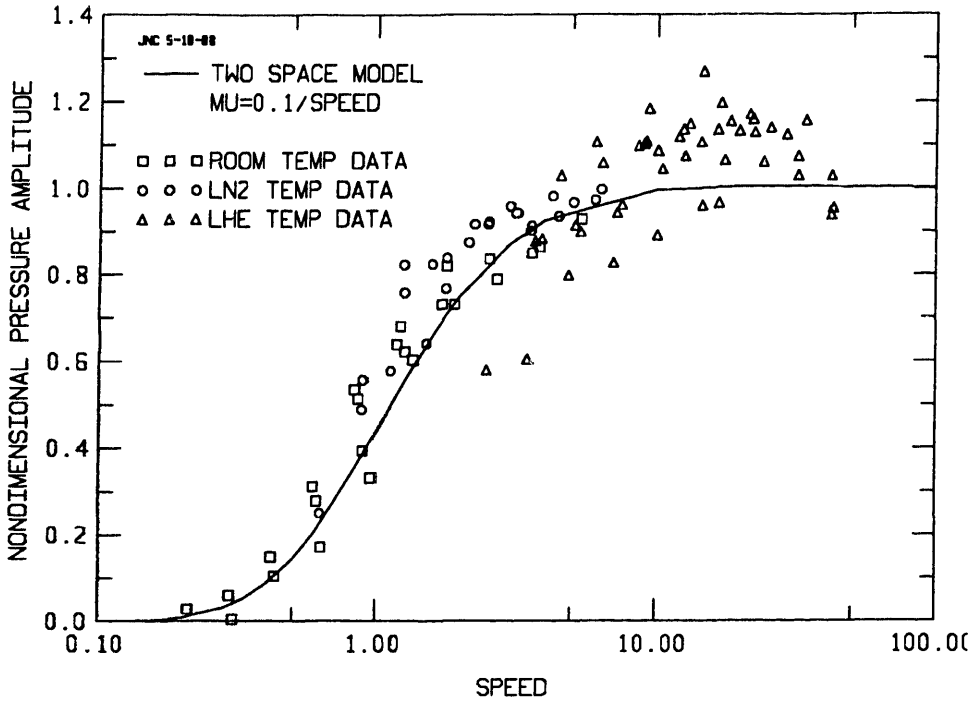

Fig. 4.11 First harmonic pressure amplitudes (normalized) for both two-space model and experiment data ($SPEED = 0.11(Pe)^{1/2}$ for experiment data; $SPEED = \tau_2$ for model).

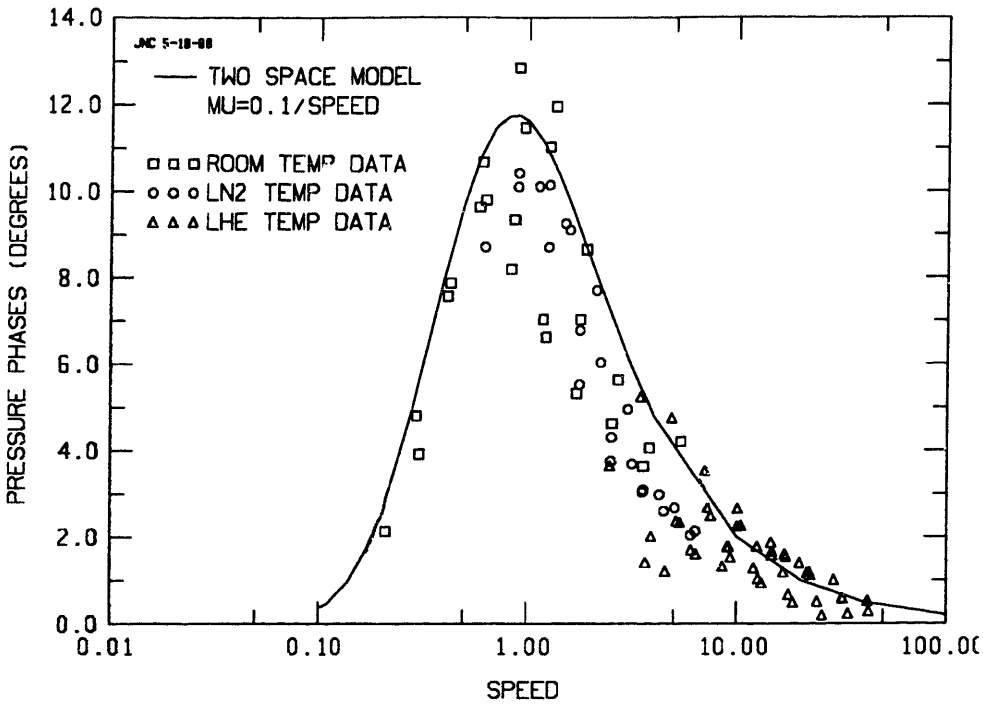

Fig. 4.12 First harmonic pressure phases for both the two-space model and experiment data ($SPEED = 0.11(Pe)^{1/2}$ for experiment data; $SPEED = \tau_2$ for model).

4.3 Gap-Flow Model

Reciprocating machines used at cryogenic temperatures typically have a geometry which is quite different from more standard reciprocating machines. The gap-flow model presented in this section demonstrates how this geometry can effect the shape of a pressure wave.

Figure 4.13 is a sketch of this type of machine. The volume enclosed by the piston and cylinder may be divided into two separate regions. We will refer to the cylindrical region between the piston face and the cylinder head as the main working space and the annular region between the piston and surrounding cylinder as the appendix gap. The total working space is the sum of these two regions. The volume of the main working space varies with piston motion while the volume of the appendix gap is constant since the seal rides with the piston. The temperature in the appendix gap varies along its length from room temperature at the seal to operating temperature at the main working space. In order to keep the conduction heat leak to the main working space small, the piston and cylinder are made very long compared to their diameter. The shape of the appendix gap provides a large amount of surface area for a very small volume, which acts as a very effective heat exchanger.

When the pressure in the total working space fluctuates due to the piston motion, mass flows in and out of the appendix gap. For example when the pressure in the gas rises, mass from the cold main working space is forced into the warm appendix gap. This gas is warmed as it flows in the gap. It is this effect that we will now attempt to model.

4.3.1 Development of the Gap-Flow Model

In order to develop a model which demonstrates the effects of gap-flow we will make the following assumptions:

1. The wall temperature distribution along the appendix gap is constant in time;

2. The main working space is adiabatic, i.e. there is no heat transfer to or from the walls;

3. When mass leaves the main working space and enters the appendix gap it has a temperature equal to the mixed mean temperature of the mass in the main working space;

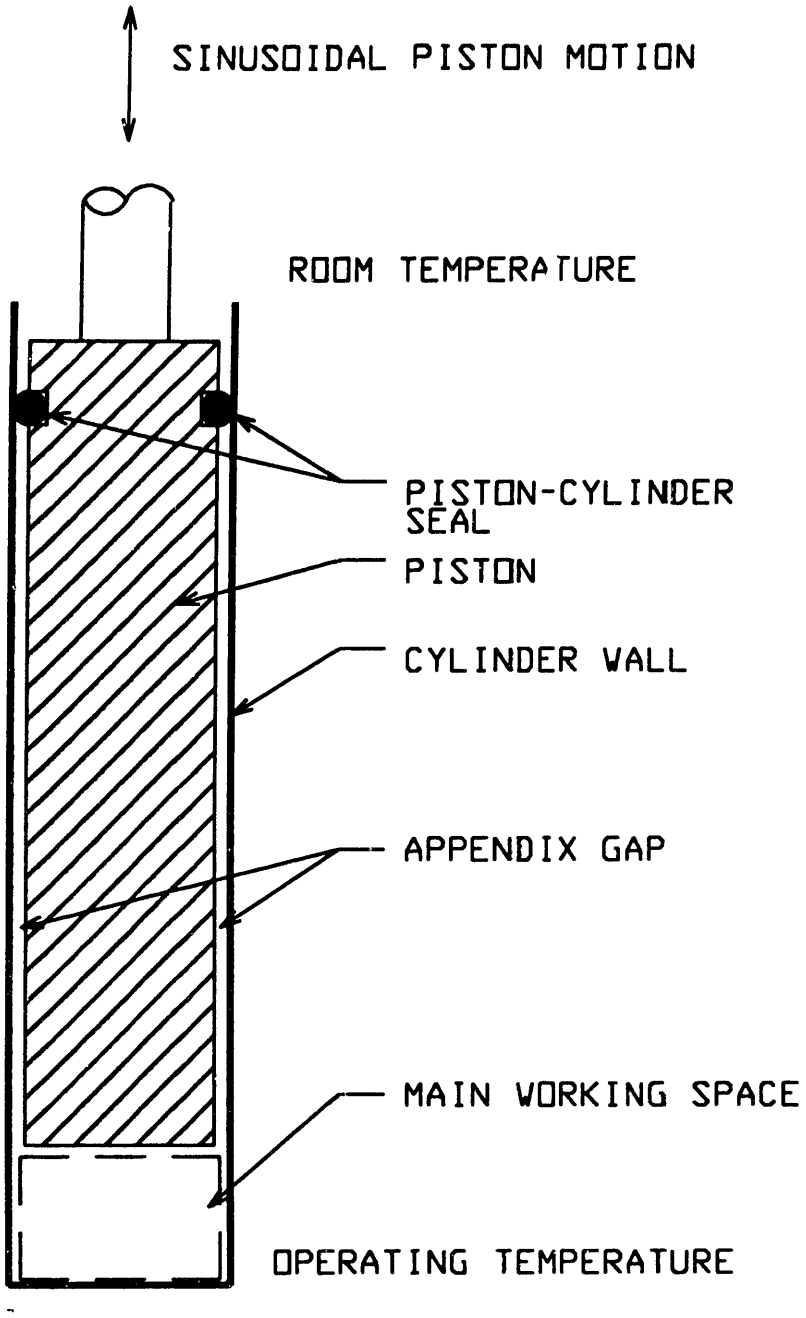

Fig. 4.13 Sketch of a typical cryogenic expander.

4. When mass leaves the appendix gap it has a temperature equal to the temperature of the cold end of the appendix gap;

5. The gas obeys the perfect gas law.

Figure 4.14 is a schematic of this model. The appendix gap shown in the schematic is now located in the cylinder head. This is done to avoid confusion caused by the piston motion. The piston motion causes the gap temperature to vary over a cycle, but this model assumes that this variation is negligible. The main working space is assumed to be adiabatic while the gap is locally isothermal. The main working space has a volume which varies sinusoidally and we denote as V. Thus

$$V(t) = V_l + V_m \sin(t) \tag{4.49}$$

where V_l = time averaged volume in main working space,

V_m = amplitude of variation in volume of main working space.

The appendix gap has a volume which is constant in time and we denote as V_g. The volume, υ, of the total working space is simply the sum of V and V_g thus;

$$\upsilon(t) = V_g + V_l + V_m \sin(t). \tag{4.50}$$

We may calculate the amount of mass in the appendix gap, M_g, by integrating the gas density over the volume of the appendix gap as follows:

$$M_g = \int_0^{V_g} \rho_g dv \tag{4.51}$$

where ρ_g = the density of the gas in the gap.

If we use the perfect gas law then the mass in the appendix gap becomes:

$$M_g(t) = \frac{P(t)V_g}{R}\frac{1}{L}\int_0^L \frac{dx}{T_g} \tag{4.52}$$

where x = location along the appendix gap,

T_g = temperature of the gas in the gap, and T is a function of x alone,

L = length of appendix gap

We have assumed that the appendix gap has a constant cross section and a length of L.

From equation 4.52, the total mass in the appendix gap is proportional to the pressure only. From the discussion of the physics above we see that any mass leaving the gap has a temperature equal to T(0), the temperature at the end of the appendix

gap. For simplicity we make the assumption that this temperature is equal to the operating temperature of the main working space, T_l. To simplify the thermodynamic analysis which follows it is convenient to introduce an equivalent volume, V_{eq}, such that;

$$M_g(t) = \frac{P(t)V_{eq}}{RT_l} = \frac{P(t)V_g}{R}\frac{1}{L}\int_0^L \frac{dx}{T} \tag{4.53}$$

or,

$$V_{eq} = T_l V_g \frac{1}{L}\int_0^L \frac{dx}{T} \tag{4.54}$$

The equivalent volume, V_{eq}, may be thought of as the volume which the mass M_g would occupy if it were at the uniform temperature T_l and subjected to the same pressure P.

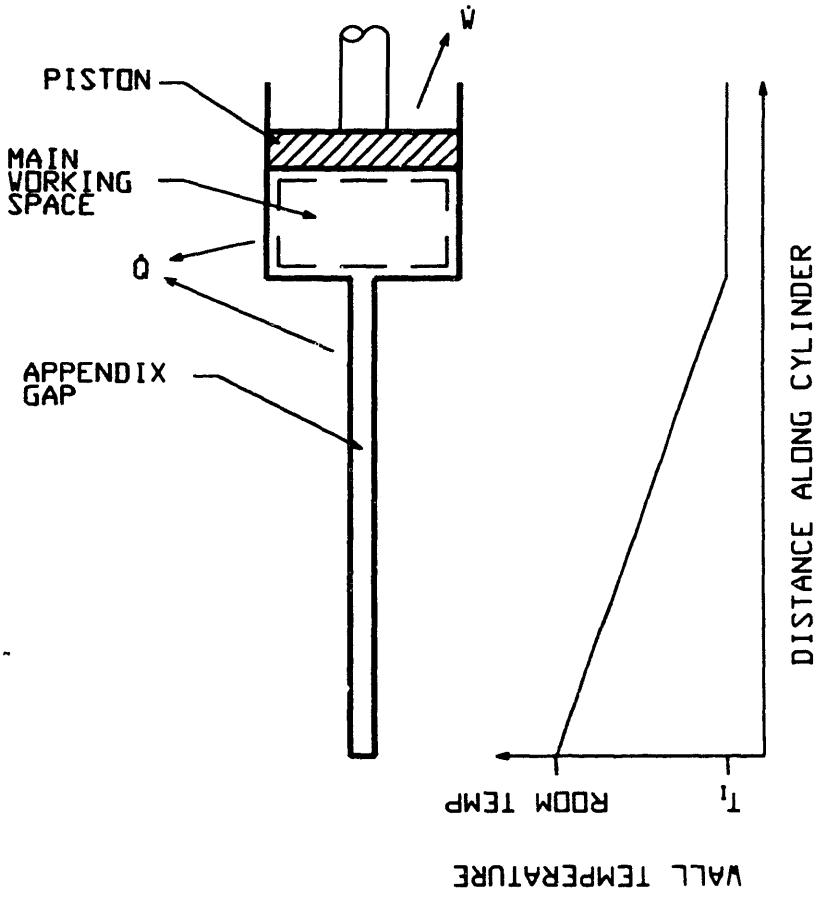

Fig. 4.14 Thermodynamic system model for the gap-flow and hybrid models.

4.3.2 Governing Equations

Referring to figure 4.14, we write the rate form of the first law of thermodynamics for the main working space as:

$$\dot{Q} - \dot{W} + h\frac{dm}{dt} = \frac{dU}{dt} \tag{4.55}$$

where $\dot{Q}$ = heat transfer rate to the main working space (always equals 0),

$\dot{W} = PdV$ = rate of work done by the gas in the main working space

h = specific enthalpy of the mass entering or exiting the main working space,

dm/dt = rate of mass entering the main working space,

dU/dt = rate of change of total internal energy in the main working space.

The specific internal energy and specific enthalpy for a perfect gas are

$$u(T) = C_v(T - T_l) \tag{4.56}$$

$$h(T) = C_p T - C_v T_l \tag{4.57}$$

The perfect gas laws for the gas in the main working space is

$$PV = MRT \tag{4.58}$$

The continuity equation for the total working space is

$$M_T = M + M_g \tag{4.59}$$

where M_T = the total mass in the working space.

By combining equations 4.53, and 4.55 through 4.59, and taking dM to be positive (mass flowing into main space) we get

$$dP = P\frac{-\gamma dV}{V + \gamma V_{eq}} \tag{4.60}$$

and when dM is negative (mass flowing out of the main space) we get

$$dP = \frac{dV}{V} \frac{1}{\frac{1}{P - P_\infty} - \frac{1}{\gamma P}} \tag{4.61}$$

where

$$P_\infty = \frac{RT_l M_T}{V_{eq}} \tag{4.62}$$

P_∞ can be thought of as the pressure that would occur if all the gas were located in the appendix gap.

Equations 4.60 and 4.61 are the governing equations for this model of gap-flow. When solved they point out some interesting facts about how a long thin gap can influence the heat transfer in a reciprocating machine. In the next section we will nondimensionalize these equations before looking at some solutions, but first we will emphasize some limitations and reiterate some of the major assumptions made in deriving them.

4.2.3.1 Discussion of Assumptions

This model assumes that the temperature of the gap is independent of time. When mass is forced in and out of the appendix gap it must exchange heat with the appendix gap walls. This implies that the heat capacity of the walls must be quite large compared to the heat capacity of the gas. For most actual situations this will be the case. At very low temperatures however, this assumption may be a poor one, at least for the coldest portion of the appendix gap.

Another assumption is that the piston motion has little or no effect on the appendix gap temperature. If the piston is quite long compared with its stroke this is a good assumption. Also for this to be a reasonable assumption the temperature gradient in the appendix gap must be relatively small.

This model assumes that the main working space is adiabatic. This was done so that we could emphasize the heat transfer in the appendix gap. With some other assumption the effects of the appendix gap might not be so apparent.

This model assumes that the working gas behaves as an ideal gas. Consequently the analysis was much simpler but a different more complex equation of state could have been used and is more appropriate for low temperatures. Note here that equation 4.53 which introduced the relation

$$M_g = P\frac{V_{eq}}{RT_l}$$

is not the perfect gas relation for the mass in the appendix gap but is rather a relation which emphasizes the fact that M_g is proportional to pressure.

When mass enters the main working space from the appendix gap, this model assumes that the temperature of the gas is at T_l and that calorimetric mixing occurs with the gas in the main working space instantaneously. That is, as the gas enters the working space, an irreversible mixing occurs. Another irreversible process occurs when mass in the main working space, having a temperature T, enters the appendix gap and its temperature instantaneously changes to T_l.

4.3.3 Nondimensionalization

In order to nondimensionalize equations 4.60 and 4.61 the following reference state is introduced. At time t = 0 we define that the temperature throughout the main working space is T_l and that the pressure is P_l. This reference state and equation 4.60 specify the mass in the total working space.

We also define the following variable and parameters:

$$P^* = P/P_l, \tag{4.63}$$

$$TR = \frac{L}{T_l\int_0^L \frac{dx}{T}}, \tag{4.64}$$

$$VR = \frac{V_g + V_l + V_m}{V_g + V_l - V_m}, \tag{4.65}$$

$$VRG = \frac{V_l}{V_{eq}}. \tag{4.66}$$

P^* is the nondimensional pressure. TR is the temperature ratio that relates the temperature distribution in the appendix gap to the operating temperature. VR is the volume ratio for the total working space and VRG is the gap volume ratio which relates the average volume of the main working space to the equivalent volume of the appendix gap.

In terms of the above parameters equations 4.60 and 4.61 may be reduced to the following nondimensional form:

$$dP^* = \frac{-P^*\gamma(TR)\cos(t)}{\frac{VR+1}{VR-1}\{\gamma+\frac{VRG(TR-\gamma)}{VRG+TR}\}+TR\sin(t)} \tag{4.67}$$

$$dP^* = \frac{\cos(t)}{\{\frac{VR+1}{VR-1}\frac{VRG}{VRG+TR}+\sin(t)\}\left\{\frac{1}{P^*-VRG-1}-\frac{1}{\gamma P^*}\right\}} \tag{4.68}$$

4.3.4 Results

Solving equations 4.67 and 4.68 for several cases which have different values of parameters illustrates some interesting ways these parameters effect the pressure wave (the method of solution is as well as the computer code implementation of this solution is presented in Appendix E). Table 4.1 lists the values of these parameters for each case. Figure 4.15 shows the effect of allowing *VRG* to vary while keeping *TR* constant. This is equivalent to varying the size of the appendix gap and running at similar operating temperatures. Figure 4.16 shows the effect of holding the gap volume ratio, *VRG*, constant and allowing the temperature ratio, *TR*, to vary. This is not exactly like keeping the gap size constant and varying the operating temperature (since *VRG* is a function of *TR*) but is roughly equivalent to it. Both figures 4.15 and 4.16 have an adiabatic pressure wave plotted to aid in relating the various pressure waves to the heat transfer. This adiabatic pressure wave is simply

$$P^* = \left\{\frac{\upsilon(t)}{\upsilon(0)}\right\}^{\gamma} \tag{4.69}$$

Table 4.1. Values of parameters used in the solution to equations 4.67 and 4.68.

-	$VR = 1.75$		
Case #	*VRG*	*TR*	Plotted in figure #
1	2.0	1.0	4.15, 4.16
2	4.0	1.0	4.15
3	8.0	1.0	4.15
4	2.0	2.0	4.16
5	2.0	4.0	4.16

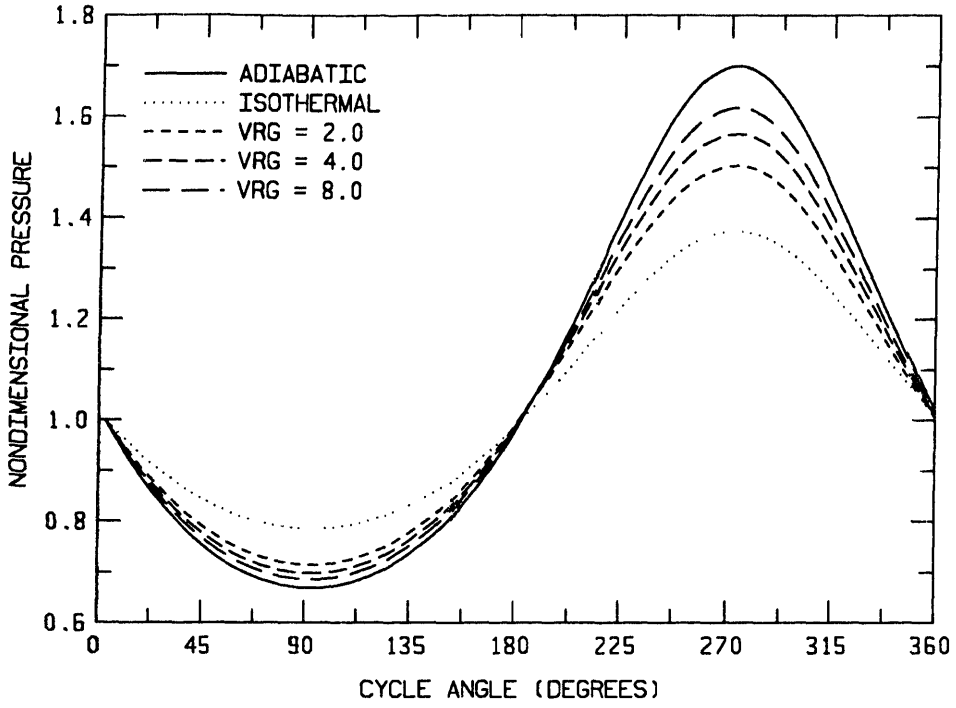

Fig. 4.15 Gap-flow model pressure waves (*TR*=1.0, *VR*=1.75, γ=1.667).

Referring to figure 4.15, we note that the temperature ratio *TR* = 1.0 implies that the temperature throughout the gap is uniform. This is the situation for a device operating at room temperature. Thus the model is primarily an adiabatic one with some small portion of it, the appendix gap, isothermal. The higher the value of *VRG*, the smaller the size of the appendix gap and the closer to adiabatic the model operates. Indeed if *VRG* is allowed to go to infinity, then appendix gap volume goes to zero and equations 4.67 and 4.68 revert to the adiabatic case.

In figure 4.16, the temperature ratio *TR* is allowed to vary as the gap volume ratio *VRG* stays constant. As mentioned earlier a temperature ratio of 1.0 implies that the temperature of the appendix gap is uniform. When the operating temperature T_l is lower than ambient temperature, the temperature ratio *TR* is greater than 1.0. This is the case for cryogenic machines. When the operating T_l is greater than ambient temperature, then *TR* is less than 1.0. This might be the case for reciprocating air

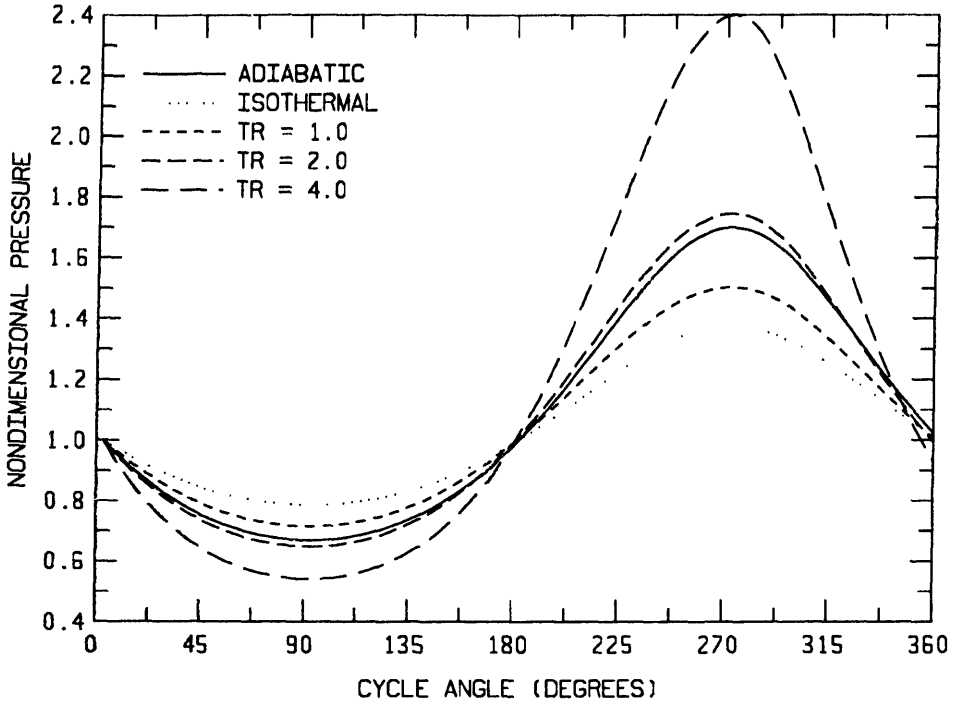

Fig. 4.16 Gap-flow model pressure waves (*VRG*=2.0, *VR*=1.75, γ=1.667).

compressors or other similar machines. In figure 4.16, as in figure 4.15, when the temperature ratio is 1.0, the pressure wave is less than the adiabatic wave. However it is interesting to see that as the temperature ratio increases, it becomes possible to have a pressure wave that exceeds the adiabatic case. This happens because the appendix gap allows heat transfer to the gas at a point in the cycle where the average temperature of the gas in the total working space is warmer than the operating temperature T_r.

This model combines an adiabatic space and an isothermal space. The solutions to the governing equations of this model fall not only between the adiabatic and isothermal cases but also go beyond them. This is possible only because the appendix gap's temperature is different than the operating temperature.

4.4 Hybrid Model

The hybrid model presented in this section is a combination of the one-space model and the gap-flow models. For the gap-flow model the main working space was assumed adiabatic to emphasize the effect of the gap on the heat transfer. In this model we abandon this assumption and replace it with the heat transfer assumption used in the one-space model.

4.4.1 Governing Equations

The development of the hybrid model proceeds in the same manner as the gap-flow model. The major difference is that the main working space is no longer adiabatic but rather has heat transfer proportional to the temperature of the gas in the main working space. For this model we again use the schematic of figure 4.14. Using equation 4.55, the first law of thermodynamics for this system, and the heat transfer relationship of equation 4.3a for the gas in the main working space, we get the following relationship:

$$\frac{T_I - T}{r} - P\frac{dV}{dt} + \frac{dM}{dt}h = \frac{dU}{dt} \tag{4.70}$$

As with the gap-flow model, we have two governing equations for pressure; one for the case where dM is positive and a second for the case when dM is negative. Respectively, these equation are as follows:

$$\frac{dP}{dt} = \frac{(\gamma-1)PT_I\left\{\frac{1}{P} + \frac{V}{V_{eq}(P-P_\infty)}\right\} - \gamma P\frac{dV}{dt}}{r(V+\gamma V_{eq})} \tag{4.71}$$

$$\frac{dP}{dt} = \frac{(\gamma-1)T_I\left\{\frac{1}{P} - \frac{V}{V_{eq}(P_\infty-P)}\right\} - \gamma\frac{dV}{dt}}{V\left(\frac{1}{P} + \frac{\gamma}{P_\infty-P}\right)} \tag{4.72}$$

4.4.2 Nondimensionalization

These equations are nondimensionalized using the reference state and nondimensional parameters defined for the gap-flow model (equations 4.63 through 4.66) and the following relation:

$$\tau_H = \omega r \frac{P_l V_l}{T_l R} C_v \tag{4.73}$$

In this equation, $(P_l\ V_l)/T_l$ is proportional to the mass in the main working space at initial conditions. Using these parameters, equations 4.71 and 4.72 can be reduced to the following form:

$$\frac{dP^*}{dt^*} = \frac{\frac{(\gamma-1)}{\tau_H} P^* \left\{ \frac{1}{P^*} + \frac{VRG + \frac{(VRG+TR)(VR-1)}{VR+1}\sin(t^*)}{P^* - VRG - 1} \right\} - P^* \gamma \cos(t^*)}{\frac{VR+1}{VR-1}\frac{VRG+\gamma}{VRG+TR} + \sin(t^*)} \tag{4.74}$$

$$\frac{dP^*}{dt^*} = \frac{\frac{(\gamma-1)}{\tau_H} \left\{ \frac{1}{P^*} + \frac{VRG + \frac{(VRG+TR)(VR-1)}{VR+1}\sin(t^*)}{P^* - VRG - 1} \right\} - \gamma \cos(t^*)}{\left\{ \frac{1}{P^*} + \frac{\gamma}{VRG+1-P^*} \right\} \frac{VR+1}{VR-1}\frac{VRG}{VRG+TR} + \sin(t^*)} \tag{4.75}$$

4.4.3 Results

Figures 4.17, 4.18, and 4.19 show the solution to equations 4.74 and 4.75 for several values of the parameters τ_H, TR, and VRG. Appendix E presents the method used to solve these equations. Table 4.2 lists the values of these parameters for the pressure waves shown in the figures.

Figure 4.17 shows the pressure waves for several values of τ_H. The values of the other parameters do not change from one pressure wave to another. As in the one-space and two-space models, τ_H is a parameter which is proportional to the cycle speed. It also has been nondimensionalized so that as it approaches zero the main working space becomes more and more isothermal. As τ_H approaches infinity, the main working space becomes adiabatic. For τ_H approximately equal to 1.0, the working space operates somewhere between the isothermal and adiabatic conditions.

Figure 4.18 shows the pressure waves for several values of the temperature ratio, TR. The parameters τ_H and VRG are held constant for the various curves in this figure. This figure shows, as did the gap-flow model, that the pressure wave can exceed the adiabatic wave. This occurs for the same reasons given in the gap-flow model section.

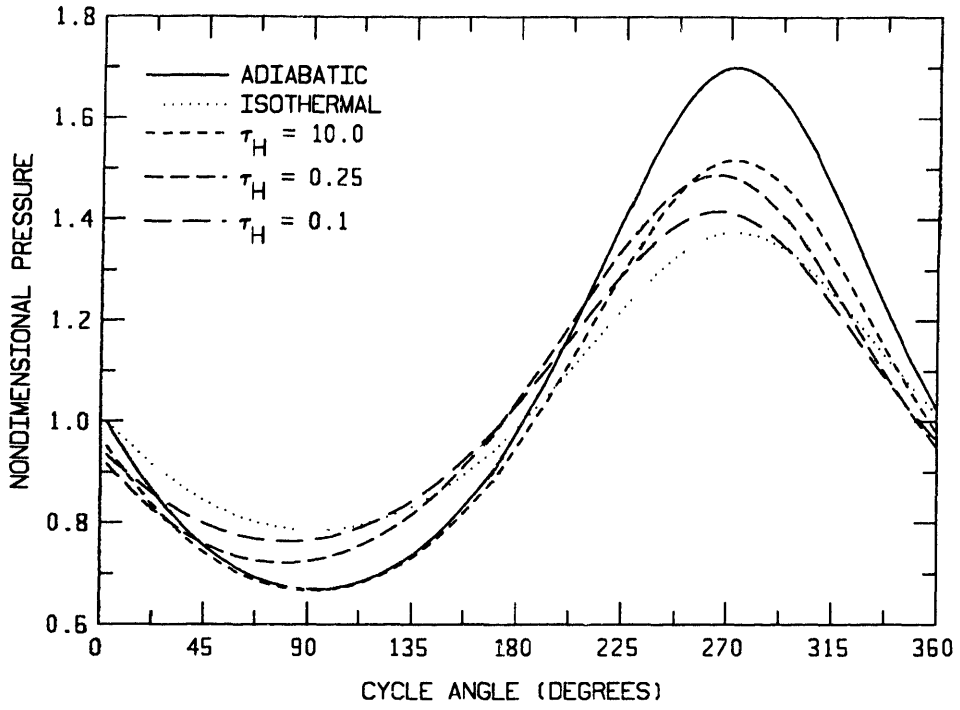

Fig. 4.17 Hybrid model pressure waves (*TR*=1.0, *VRG*=4.0, *VR*=1.75, γ=1.667).

Figure 4.19 shows pressure waves for several values of the gap volume ratio, *VRG*. All the other parameters are kept constant. As discussed in the gap-flow model, this ratio indicates the size of the gap relative to the main working space. As *VRG* increases the size of the gap becomes small in comparison to the main working space, thus the gap plays a smaller and smaller role in affecting the pressure wave. The effect of the gap is further diminished when the main working space is nearly isothermal. This is because the gap is also isothermal and it therefore behaves just like the rest of the working space.

In order to present a quantitative comparison between the hybrid model and the experiment data, the model is used to calculate several pressure waves for a range of model parameters. These pressure waves are decomposed into their Fourier series and normalized using equations 3.7 and 3.8. The model parameters used to calculate the

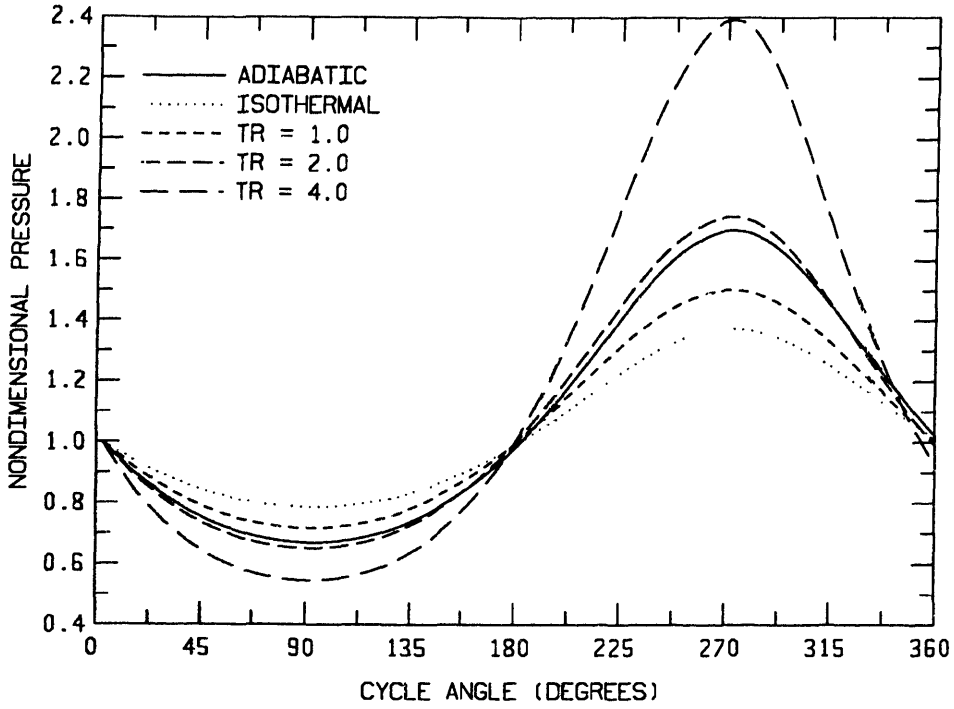

Fig. 4.18 Hybrid model pressure waves (τ_H=10.0, *VRG*=2.0, *VR*=1.75, γ=1.667).

pressure waves are carefully chosen to match the characteristics of the apparatus. As in the previous models, several of the parameters have direct counterparts. γ is the ratio of specific heats for helium (1.667 at room temperature). *VR* is the volume ratio of the apparatus (1.75). For the same reasons discussed for the one-space model, we make the assumption that τ_H is proportional to the square root of the Peclet number. The parameter *VRG* is written as a function of both the apparatus geometry and the parameter *TR* in the following way (see equations 4.54, 4.64 and 4.66):

$$VRG = TR\frac{V_l}{V_g} \tag{4.76}$$

For this apparatus the ratio of the average volume of the main working space, V_l, and the volume of the appendix gap, V_g is approximately 8.0. The temperature ratio, *TR*, is dependent on the temperature distribution along the appendix gap (see equation 4.64). Estimates of *TR* and the corresponding values of *VRG* are presented in table

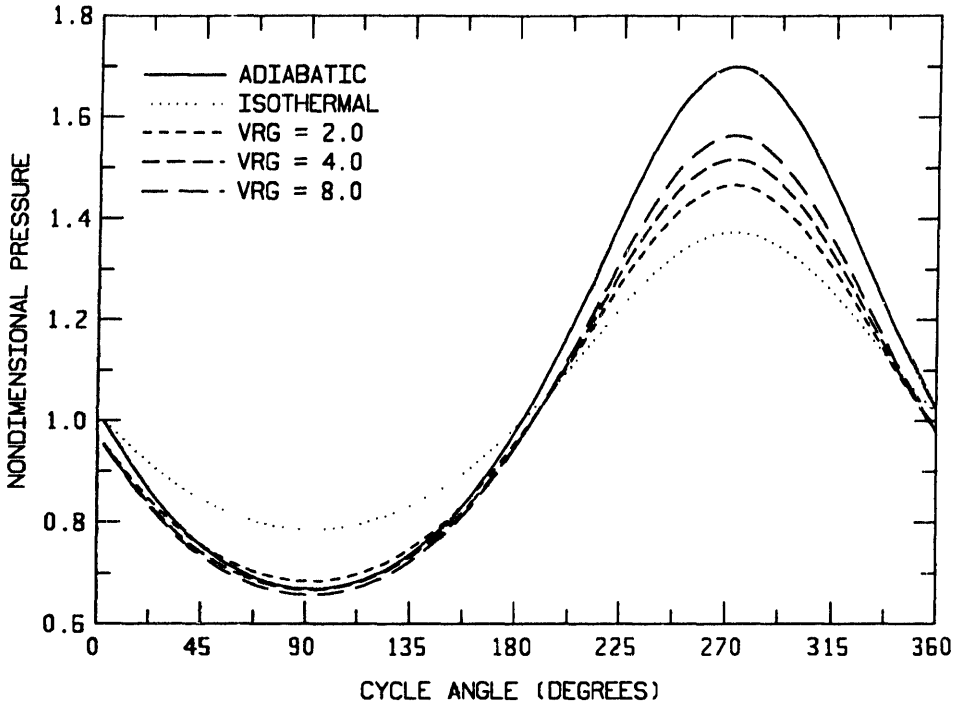

Fig. 4.19 Hybrid model pressure waves (τ_H=10.0, *TR*=1.0, *VR*=1.75, γ=1.667).

Table 4.2 Values of parameters used in the solution to equations 4.74 and 4.75

VR = 1.75			γ = 1.667	
Case #	τ_H	*VRG*	*TR*	Plotted in figure #
1	0.1	4.0	1.0	4.17, 4.19
2	4.0	4.0	1.0	4.17
3	10.0	4.0	1.0	4.17
4	0.1	2.0	1.0	4.18
5	0.1	2.0	2.0	4.18
6	0.1	2.0	4.0	4.18
7	0.1	2.0	1.0	4.19
8	0.1	8.0	1.0	4.19

Table 4.3 Estimated values of *TR* and *VRG* for the experiment operating temperature ranges

Temperature range	Temperature ratio	Gap volume ratio
	TR	*VRG*
Room (300K)	1.0	8.0
Liquid nitrogen (77K)	1.3	10.4
Liquid helium (10K)	2.0	16.0

4.3 for the three operating temperature ranges. When first deriving the gap-flow model, we made the assumptions that the piston motion had little effect on the appendix gap temperature distribution and that the temperature gradient along the appendix gap is small relative to the piston stroke. By making these assumptions, the values of *TR* and *VRG* remain constant over a cycle. However, for the experiment apparatus operating at liquid helium temperatures, we estimate that *TR* varies by 15 percent over a cycle.

Figure 4.20 and 4.21 compare the hybrid model with the experiments. In both of these figures, the three curves correspond to the three temperature ranges. As was done for the previous models, the proportionality constant which relates τ_H for the model to $(Pe)^{1/2}$ for the experiments was chosen empirically so the results of the model matched the results of the experiments well. The speed parameter plotted along the x-axis in these figures is τ_H for the hybrid model and $0.011(\mathrm{Pe})^{1/2}$ for the experiment data. Figure 4.20 is a plot of the first harmonic pressure amplitudes for both the experiment data and the model. It highlights several interesting features of the hybrid model. For the curve which corresponds to room temperature operating conditions (*TR* = 1.0) at very high values of τ_H the amplitudes do not approach 1.0 as they did for the one-space and two-space models but instead approach approximately 0.8. This implies that no matter how fast the apparatus is run, it will not reach adiabatic conditions. At low values of τ_H, the amplitudes do approach 0.0 as the other models did. As τ_H increases from 0.05 to approximately 0.75, the amplitudes increase from almost 0.0 to a maximum of just less than 0.9. At values of τ_H greater than 0.75, the amplitudes decrease slightly. In the transition region, from small amplitudes to large ones, the slope of this model matches well with the data. The amplitude curves of the other two temperature ranges have the same shape as the one for room

temperature but they are displaced to higher amplitude values. The curve for *TR* equal to 1.3 corresponds to liquid nitrogen temperature operation. It has a maximum amplitude of just under 1.0 and approaches 0.9 as τ_H approaches infinity. As τ_H approaches 0.0 this curve does not approach 0.0 but instead approaches 0.5. This implies that no matter how slow the apparatus is run, it will not reach isothermal conditions. The curve for *TR* equal to 2.0 corresponds to liquid helium temperature operation. It has a maximum value greater than 1.1 and approaches a value slightly larger than 1.0 as τ_H approaches infinity. As τ_H approaches 0.0, the pressure amplitudes approach 0.1. The small hump which occurs in all three pressure amplitude curves at τ_H slightly less than 1.0 is unique to this model. The data collected at helium temperature indicates that this is indeed the behavior of the real system at low temperatures. The room and liquid nitrogen temperature data do not show this behavior but their limited range of Peclet numbers do not coincide with conditions where this hump would be evident. Figure 4.21 is a plot of the first harmonic pressure phases for the data and the model. The hybrid model's phase shift curves have the same general shape as those of the one-space model. However this model matches the experiment data better because the maximum values of phase shifts are closer to the maximum phase shifts measured by the experiments.

This model combines the important features of both the one-space and the gap-flow models. It also represents the interaction between the heat transfer and thermodynamics of the gas that is crucial for accurately predicting the behavior of the pressure during a cycle. It provides a simple representation of the cyclic heat transfer and also accounts for the appendix gap effects which can be significant for cryogenic reciprocators. The model predicts that the maximum first harmonic pressure amplitude for the room temperature range is less than 1.0, that of the liquid nitrogen temperature range is approximately equal to 1.0, and that of the liquid helium temperature range is greater than 1.0. This is in agreement with the experiment results.

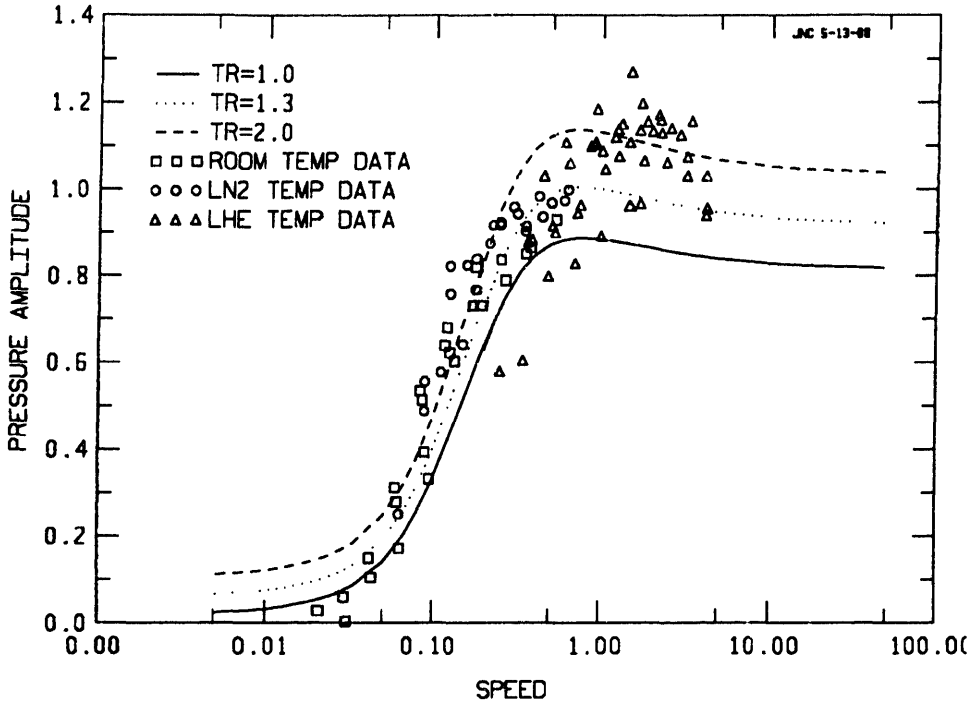

Fig. 4.20 First harmonic pressure amplitudes (normalized) for both the hybrid model and experiment data.
($SPEED = 0.011(Pe)^{1/2}$ for experiment data; $SPEED = \tau_H$ for model)

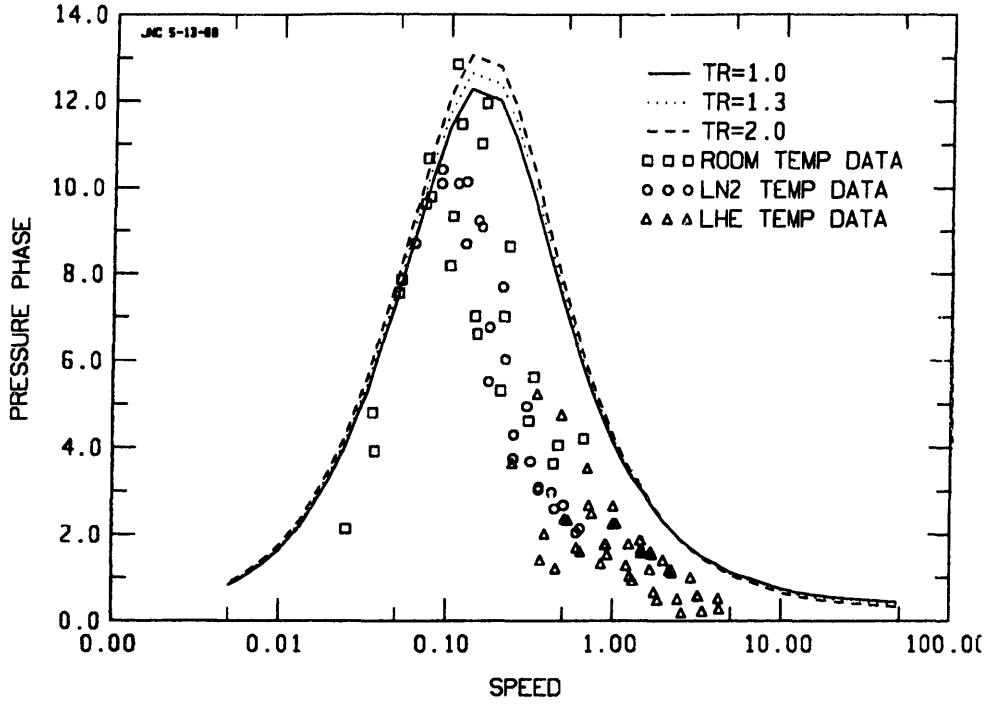

Fig. 4.21 First harmonic pressure phases for both the hybrid model and experiment data. ($SPEED = 0.011(Pe)^{1/2}$ for experiment data; $SPEED = \tau_H$ for model)

Chapter 5
RESULTS AND CONCLUSIONS

This chapter shows how to calculate the heat transfer from the experiment data and presents the thermodynamic loss associated with the apparatus and the models. A new parameter which represents a nondimensional speed is presented as an improvement on the experiment correlations. This chapter also consolidates the most important and interesting features of this research.

5.1 Heat Transfer from Pressure and Volume

The spatially averaged instantaneous heat transfer for a gas spring can be calculated from the pressure and volume phases and amplitudes. To do this it is helpful to write the first law of thermodynamics for a gas spring:

$$\dot{Q} = \frac{dU}{dt} + P\,d\frac{\upsilon}{dt} \tag{5.1}$$

The rate of change of energy, *dU/dt*, may be written in terms of pressure and volume using an equation of state. For this analysis the perfect gas law is used as follows:

$$\frac{dU}{dt} = \frac{C_v}{R}\frac{d(P\upsilon)}{dt} = \frac{C_v}{R}\left\{\upsilon\frac{dP}{dt} + P\frac{d\upsilon}{dt}\right\} \tag{5.2}$$

Using this equation and γ to represent the ratio of specific heats equation 5.1 may be rewritten as follows:

$$\dot{Q} = \frac{\gamma}{\gamma-1}P\frac{d\upsilon}{dt} + \frac{1}{\gamma-1}\upsilon\frac{dP}{dt} \tag{5.3}$$

The pressure and volume functions are represented as Fourier series which may be written as follows:

$$P(t) = P_0 + P_1\cos(\omega t - \theta_1) + P_2\cos(2\omega t - 2\theta_2) + \ldots \tag{5.4}$$

$$\upsilon(t) = \upsilon_0 + \upsilon_m\cos(\omega t - \phi) + \ldots \tag{5.5}$$

Equation 5.3 involves not only the pressure and volume functions but also their time derivatives which can be easily calculated when the pressure and volume functions are represented by Fourier series. Thus:

$$\frac{dP}{dt} = -\omega P_1 \sin(\omega t - \theta_1) - 2\omega P_2 \sin(2\omega t - 2\theta_2) + \ldots \tag{5.6}$$

$$\frac{d\upsilon}{dt} = -\omega \upsilon_m \sin(\omega T - \phi) + \ldots \tag{5.7}$$

The heat transfer can be calculated by substituting equations 5.4, 5.5, 5.6, and 5.7 into equation 5.3. An example calculation of the heat transfer for the experiment apparatus is given in Appendix C.

5.2 Thermodynamic Loss

The inefficiency of a gas spring averaged over a full cycle is represented by the thermodynamic loss. This loss is indicative of the energy lost to irreversibility and is proportional to the area enclosed by the cycle's P-V diagram. The precise definition of the thermodynamic loss is somewhat arbitrary but is chosen so that it can be easily calculated for the models and the experiment data.

5.2.1 Definition of Thermodynamic Loss

We define the nondimensional thermodynamic loss for a gas spring as follows:

$$Loss = -\oint \frac{P}{P_l} \frac{d\upsilon}{\upsilon_l} = \frac{1}{P_l \upsilon_l} \oint P d\upsilon \tag{5.8}$$

The loss is nondimensionalized using P_l and υ_l as reference points. P_l and υ_l are the values of the pressure and volume waves at a point in the cycle when the piston is at mid-stroke and the volume is increasing. This particular reference point is chosen because it is the same reference point used to calculate the adiabatic and isothermal pressure waves for the data reduction of the experiment. This greatly simplifies the algebra required to calculate the loss.

5.2.2 Method of Calculation

To calculate the thermodynamic loss, the integral in equation 5.8 must be solved. This can be done by numerically differentiating the volume data, then multiplying by the pressure data, and then finally numerically integrating this result. The numerical differentiation of the volume data is difficult to do accurately because noise in the data is greatly amplified. Therefore some form of digital filtering is

required. Instead of solving this integral numerically, we chose to use the Fourier series representation of the pressure and volume waves to calculate the loss. In a sense the harmonic decomposition used in obtaining the Fourier series representation is a form of digital filtering and it also allows us to perform the differentiation and integration required to calculate the loss by analytical methods. Using the Fourier series representation of the pressure and volume waves, equation 5.8 may be rewritten as follows:

$$Loss = -\oint \frac{\upsilon_1}{\upsilon_l}\cos(t^*)\frac{P_0}{P_l}dt^* - \oint \frac{\upsilon_1}{\upsilon_l}\cos(t^*)\frac{P_1}{P_l}\sin(t^* + \theta_1)dt^* - \oint \frac{\upsilon_1}{\upsilon_l}\cos(t^*)\frac{P_2}{P_l}\sin(2(t^* + \theta_2))dt^* \tag{5.9}$$

The cyclic integrals in the above equation can be evaluated by integrating over the interval of 0 to 2π. When the first and third integrals are evaluated over this interval, the result is always 0.0. Thus for the pressure waves considered in this thesis, only the first harmonic components have an effect on the loss. The loss can be written as a function of the first harmonic amplitudes and phases:

$$Loss = -\pi\frac{P_1}{P_l}\sin(\theta_1) \tag{5.10}$$

In chapters 3 and 4, the pressure amplitudes are normalized by relating them to the amplitudes of adiabatic and isothermal pressure waves (see equation 3.7). The reference pressure and volume, P_l and υ_l, used to calculate the adiabatic and isothermal waves are the same reference properties used to calculate the loss. To calculate the loss, the first harmonic pressure amplitudes and phases are needed. They are obtained from the normalized amplitudes and phases by using the volume ratio of 1.75 to calculate the *nondimensional* adiabatic and isothermal pressure pressure waves as follows:

$$P^*_{ad} = \frac{P_{ad}}{P_l} = \left\{\frac{1}{\upsilon^*}\right\}^{\gamma} \tag{5.11}$$

$$P^*_{isot} = \frac{P_{isot}}{P_l} = \left\{\frac{1}{\upsilon^*}\right\} \tag{5.12}$$

where

$$\upsilon^* = 1 + \frac{VR - 1}{VR + 1}\sin(t^*) \tag{5.13}$$

These waves are decomposed into phases and amplitudes and the first harmonic amplitudes are found to be 0.50 for the adiabatic wave and 0.29 for the isothermal wave. These values are used to calculate the loss for the experiment and models. To do this, the equation for the normalized pressure amplitudes (equation 3.7) may be rewritten as a function of the nondimensional adiabatic and isothermal pressure amplitudes.

$$P_1^* = \frac{\frac{P_1}{P_I} - (P_{isot}^*)_1}{(P_{ad}^*)_1 - (P_{isot}^*)_1} = \frac{\frac{P_1}{P_I} - 0.29}{0.50 - 0.29} \tag{5.14}$$

Note that in the above equation the subscript n has been replaced by 1 which denotes the first harmonic amplitude since this is the only harmonic required. We solve equation 5.14 for P_1/P_I:

$$\frac{P_1}{P_I} = 0.71P_1^* + 0.29 \tag{5.15}$$

The loss can be evaluated by substituting the nondimensional pressure amplitudes and phases, calculated for each model or experiment run, into the following equation:

$$Loss = -(0.71P_1^* + 0.29)\sin(\theta_n^* + \pi) \tag{5.16}$$

Figure 5.1 is a plot of the loss calculated from equation 5.16 for the experiment data and the one-space and two-space models. Along the x-axis, $0.11(Pe)^{1/2}$ is plotted for the experiments, τ is plotted for the one-space model, and τ_2 is plotted for the two-space model (the reason for choosing these abscissas is to provide the best empirical fit, see sections 4.1.4 and 4.2.2). This figure shows that both models provide a reasonable representation of the thermodynamic loss for the experiments. The curves of the thermodynamic loss for the models look quite similar to their first harmonic pressure phase curves in Figures 4.8 and 4.12. This similarity is directly attributable to equation 5.16. Since the loss and phase curves are similar and since the two-space model phase curves match the experiment data better than the one-space model curves, especially at low nondimensional speeds, the two-space model provides a better representation of the loss.

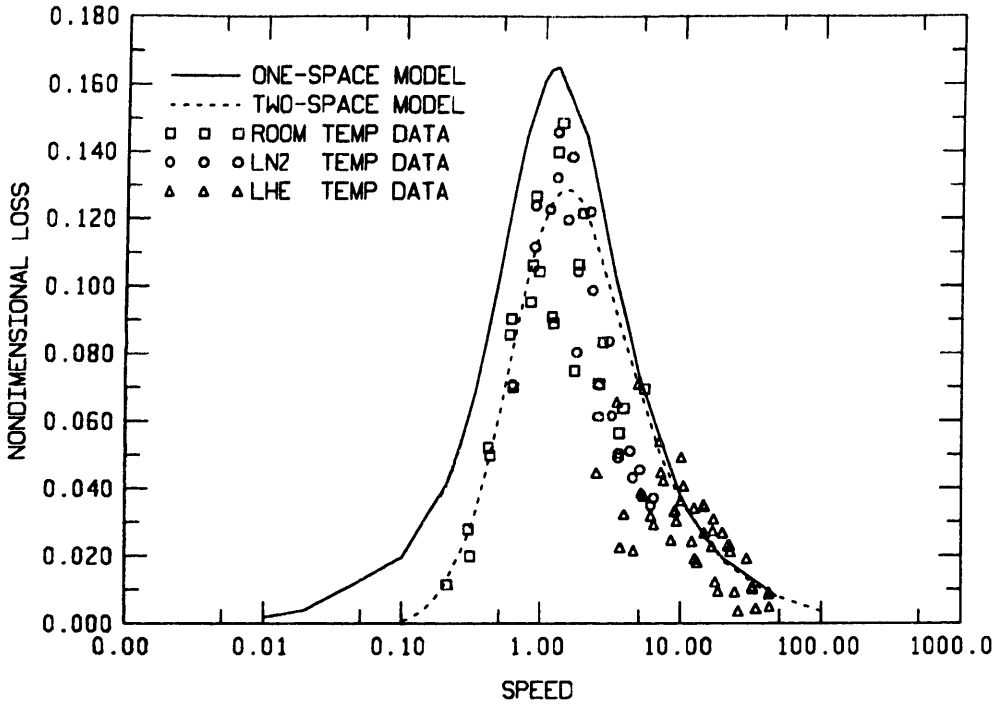

Fig. 5.1 Nondimensional thermodynamic loss for one-space model, two-space model, and experiment data ($SPEED = 0.11(Pe)^{1/2}$ for experiment data; $SPEED = \tau$ for one-space model; $SPEED = \tau_2$ for two-space model).

Figure 5.2 compares the loss for the hybrid model to the experiment loss. As discussed in section 4.4.3, the three curves for the model correspond to the three different operating temperatures. Along the x-axis, $0.02(Pe)^{1/2}$ is plotted for the experiment data and τ_H is plotted for the hybrid model (the reason for choosing these abscissas is to provide the best empirical fit, see sections 4.4.3). As would be expected, the loss curves for the hybrid model look similar to the loss curves for the one-space model. The main difference between them is the peak values for the loss. The hybrid model's maximum loss depends on the temperature ratio, *TR*, the larger *TR* is the larger the maximum value of the loss is. When the nondimensional speed τ_H reaches infinity, the hybrid model reverts to the gap flow model. In this case, the main working space becomes perfectly adiabatic but the gap remains isothermal. There is no loss associated with these ideal heat transfer processes but there is loss

associated with the calorimetric mixing which occurs when the gas flows out of the appendix gap and into the main working space. There is also loss associated with the instantaneous temperature change of the gas as it enters the gap. We find the loss associated with the gap-flow model is quite small, approximately two percent of the maximum loss calculated for the hybrid model at the same *VRG* and *TR*.

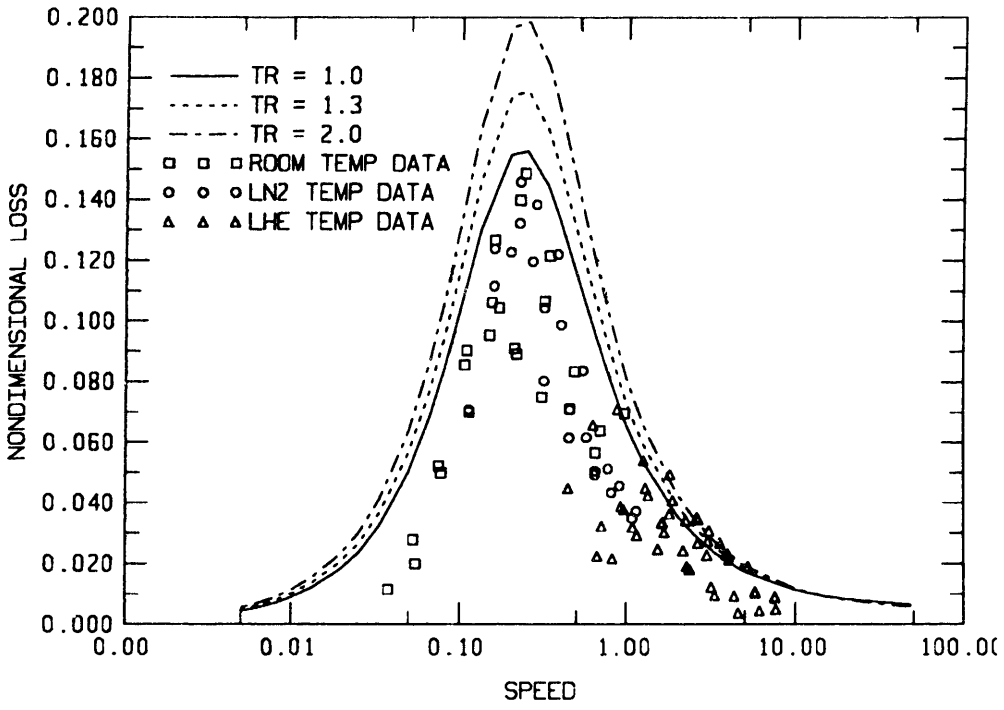

Fig. 5.2 Nondimensional thermodynamic loss for hybrid model, and experiment data (*SPEED* = $0.02(Pe)^{1/2}$ for experiment data; *SPEED* = τ_μ for hybrid model).

The plots of the thermodynamic loss presented in this section show that it is strongly dependent on the nondimensional speed. We found that at small nondimensional speeds the loss is small, at moderate nondimensional speeds the loss reaches a maximum, and at large nondimensional speeds the loss becomes small again. We found this to be true not only for the experiments but also for all of the analytic models.

5.3 New Speed Parameter

Up until this point the experiment data has been presented as functions of Peclet number. This number was suggested by Kornhauser as an improvement over the Reynolds number which Faulkner originally proposed. The Reynolds number is equivalent to the Peclet number, defined by equation 3.3, divided by the Prandtl number. The Prandtl number is the ratio of the momentum diffusivity to the thermal diffusivity. For gases at room temperature, the Prandtl number is relatively constant and thus the Reynolds number is proportional to the Peclet number for all work Faulkner performed. The Peclet number is more pertinent to gas spring heat transfer because it relates the operating speed of the system to the ability of the working gas to diffuse heat as opposed to momentum.

In this section, a more complex number is proposed to improve the correlations of the experiment data. This new speed parameter, *SPEED**, is defined as follows:

$$SPEED^* = Pe_{he}\frac{(b_{he}+b_{ss})}{(b_{he}b_{ss})^{1/2}} \tag{5.17}$$

where $b = \sqrt{\rho C_p k}$

The subscript *he* denotes helium properties and *ss* denotes stainless steel properties. We use stainless steel properties because that is what the cylinder material is. The parameter *b* is sometimes referred to as the coefficient of heat penetration. It is proportional to the amount of heat that enters a semi-infinite body when its surface is subjected to an instantaneous temperature change. The cyclic temperature variations of the gas appear to the cylinder wall as an infinite number of small temperature steps, therefore the use of *b* is appropriate. The part of *SPEED** that involves the coefficients of heat penetration provides a way of accounting for the relative speeds at which the gas and cylinder wall can absorb a given quantity of energy. Its form was chosen empirically to provide an improvement of the correlations of the experimental data for each temperature range as well as between temperature ranges. Values for the coefficient of heat penetration for stainless steel are listed in Table 5.1. Values for the coefficient of heat penetration for helium at 1.0 and 10.0 atmospheres are listed in Tables 3.1 and 3.2.

Table 5.1 Thermal transport properties for stainless steel.

PROPERTIES OF STAINLESS STEEL (ρ = 7800 kg/m^3)				
TEMPERATURE	THERMAL CONDUCTIVITY	SPECIFIC HEAT[17]	SPECIFIC HEAT[18]	
T	k	C_p	C_p	$\sqrt{\rho C_p k}$
(K)	(W/m-K)	(J/kg-K)	(J/kg-K)	(J/s$^{1/2}$-m^2-K)
2		0.03		
4	0.24			
5		0.2		22.3*
10	0.77	0.8		85.1*
20	1.95	6	4.6	324.3*
30	3.30			709.3*
40	4.70			1235.9*
50	5.80	74	66.93	1740**
60	6.80			
70	7.60			
77			158.97	
80	8.26			
100	9.40	251	238.45	4180**
120	10.36			
140	11.17			
150			327.13	
160	11.86			
180	12.47			
200	13.00		375.24	6170**
250	14.07			
300	14.90	490	426.28	7040**

* Value calculated from the formula $\sqrt{\rho C_p k} = T^{1.93}$

** Value calculated from tabulated values, specific heats used are from Scott (see footnote).

17 Gopal, E.S.R., "Specific Heats at Low Temperatures", Plenum Press, New York, 1966.

18 Scott, Russel B., "Cryogenic Engineering", D. Van Nostrand Company, Inc., Princeton, New Jersey, 1959.

5.3.1 Effects of New Nondimensional Speed on Experiment Data

Figure 5.3a is a plot of the first harmonic pressure amplitudes for the room temperature experiment versus $SPEED^*$. The effectiveness of $SPEED^*$ becomes evident when Figure 5.3a is compared with Figure 5.3b which shows the same amplitudes plotted against the Peclet number.

The scatter present in Figure 5.3b is dramatically reduced in Figure 5.3a. This reduction of data scatter is the benefit of the new nondimensional speed parameter. The first and second harmonic pressure amplitudes and phases for the data collected at all three temperature ranges are presented in Figures 5.4 through 5.7. When these figures are compared with Figures 3.10 through 3.13 we can see that again the scatter is greatly reduced when speed is used. This parameter offers significant improvement over the whole temperature spectrum.

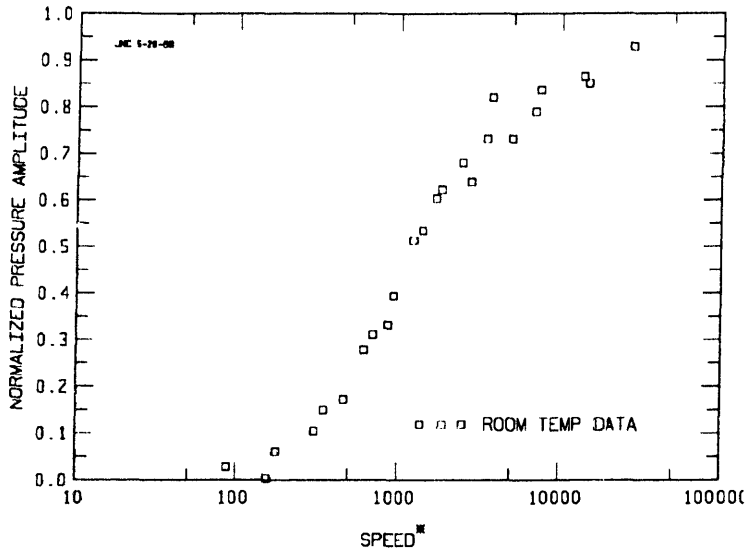

Fig. 5.3a First harmonic pressure amplitudes (normalized) for room temperature experiment runs plotted as a function of new nondimensional speed.

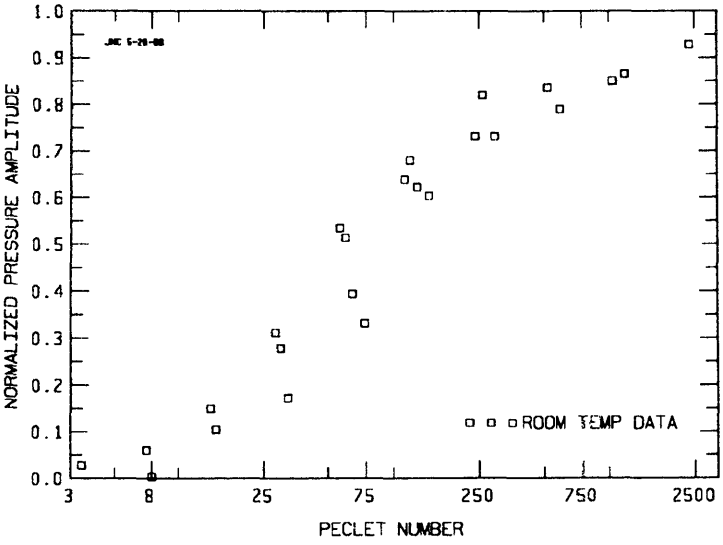

Fig. 5.3b First harmonic pressure amplitudes (normalized) for room temperature experiment runs plotted as a function of Peclet number.

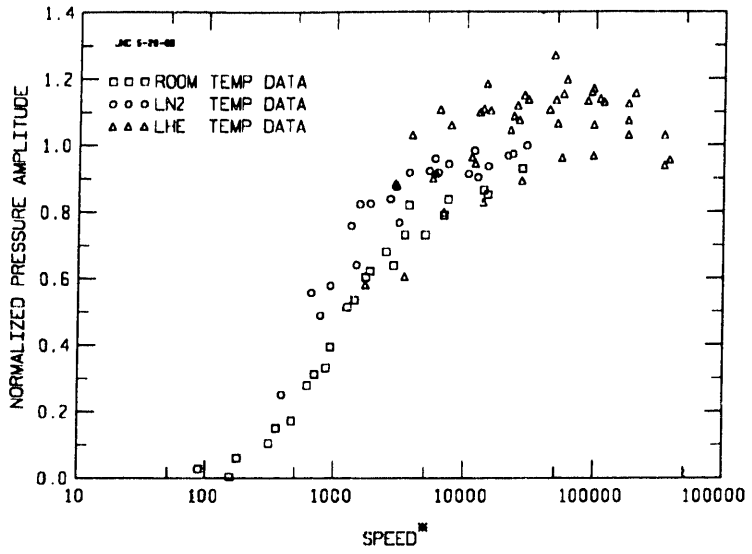

Fig. 5.4 First harmonic pressure amplitudes (normalized) for all experiment runs plotted as a function of new nondimensional speed.

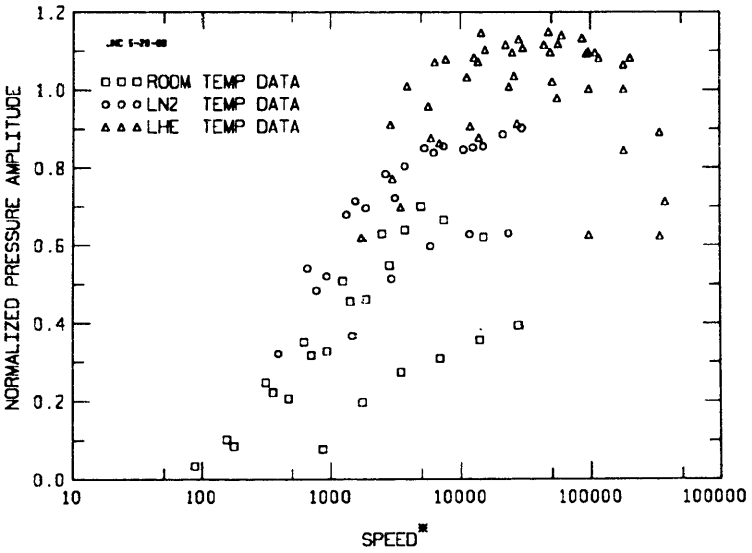

Fig. 5.5 Second harmonic pressure amplitudes (normalized) for all experiment runs plotted as a function of new nondimensional speed.

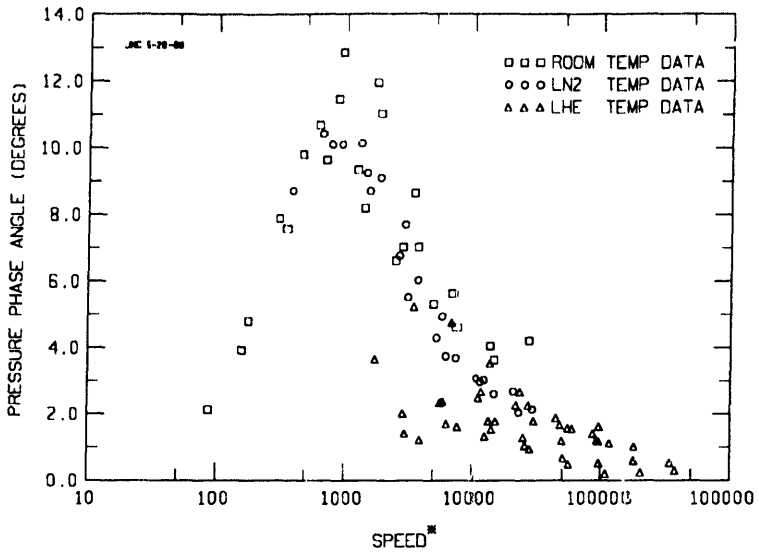

Fig. 5.6 First harmonic pressure phases for all experiment runs plotted as a function of new nondimensional speed.

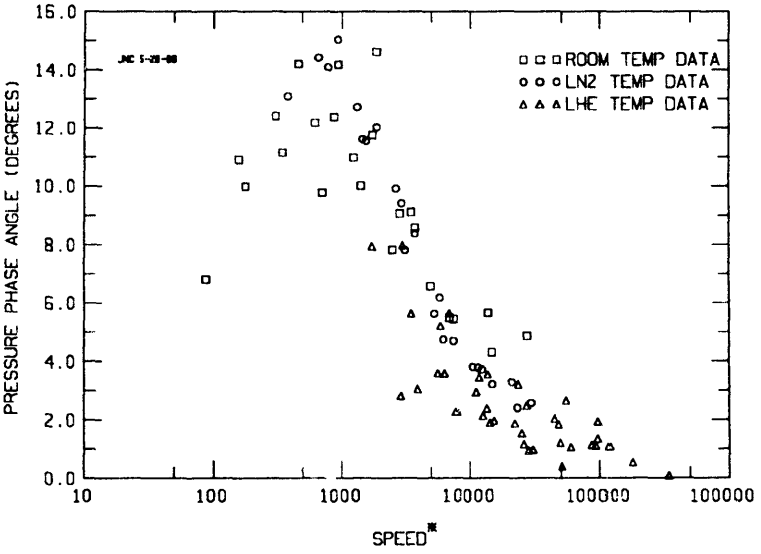

Fig. 5.7 Second harmonic pressure phases for all experiment runs plotted as a function of new nondimensional speed.

5.3.2 Modeling

The models presented in this thesis do not account for the effects of the cylinder wall properties. We will show by examining the thermal boundary layer in the cylinder wall that these properties can be important and we will present some suggestions for including them in analytic models. As in the gas near the wall, the thermal boundary layer of the cylinder wall is the region where there is a significant radial temperature gradient. We suggest that a model of a gas spring which includes the relationship between the thermodynamics and the heat transfer and also accounts for the effects of the thermal boundary layer in the cylinder wall would lead to a more complete physical understanding of why the new nondimensional speed parameter improves the correlation of the experiment data.

The behavior of the thermal boundary layer in the cylinder wall changes greatly when the operating conditions change from room temperature to cryogenic temperatures. To help show this, the relative temperature swings that occur at the inner face of the cylinder wall are estimated. When two semi-infinite bodies, which initially have uniform but different temperatures, are brought into contact with each other the following relation will hold[19]:

$$\frac{\Delta T_1}{\Delta T_2} = \frac{b_2}{b_1} \tag{5.18}$$

In the above equation ΔT equals the absolute value of the difference between the initial and final temperatures at the interface of the two bodies. The subscripts designate the two bodies. Using this relation and referring to Tables 3.1, 3.2, and 5.1, it is estimated that at room temperature operating conditions the temperature of the inner face of the cylinder wall will change 0.005 K for every 1.0 K of temperature swing the helium undergoes. At liquid helium temperature operating conditions, however, the inner face of the cylinder wall experiences temperature swings on the same order of magnitude as the helium.

One approach to modeling the effects of the cylinder wall's thermal boundary layer would be to represent this layer as a single lumped mass as we did for the gas's thermal boundary layer in the two-space model. This lump would have an independent temperature associated with it as well as a thermal capacitance. The gas in the

---

19 Grigull, U., Sandner, H.; *Heat Conduction*; Springer-Verlag, New York, NY, 1984, pg. 81.

cylinder could then be modeled as one or two lumps and the gap effect could also be included. This kind of model would give some insight into how the cylinder wall properties affect the performance correlations.

Grigull and Sandner use a more complex approach to modeling the thermal boundary layer in a solid[20]. They analyze the physics of a semi-infinite solid, modeled as a continuum, in contact with a fluid which undergoes periodic temperature fluctuations. This model provides detailed representation of the heat transfer but lacks the relationship between the heat transfer and the thermodynamics. However it might be possible to reformulate this analysis to include the thermodynamic relationships.

The development and analysis of new models which account for the cylinder wall's thermal boundary layer will provide more insight into the role of the cylinder material properties in gas spring heat transfer. We suggest further study into this topic.

5.4 Conclusions

The heat transfer between the working fluid and cylinder walls of a reciprocating machine is extremely complicated even for the relatively simple case of a gas spring. The piston-cylinder geometry, the operating speed, the thermal properties of both the working fluid and the cylinder all influence the heat transfer. The heat transfer is further complicated when operating at cryogenic temperatures because the magnitudes of the thermal transport properties of the fluid become comparable to those of the cylinder wall. Also, the behavior of the fluid becomes non-ideal, and the design of the apparatus (long piston with a short stroke) causes complex flow and heat transfer. This thesis studied gas spring heat transfer by performing experiments with a gas spring and developing models which were compared with the results of the experiments.

The experiments performed for the three operating temperatures point out some interesting points. We found that no matter what the operating temperature is, the thermodynamic loss and the amount of gas spring heat transfer during a cycle is strongly dependent on the nondimensional speed. For low nondimensional speeds, the heat transfer is nearly isothermal and the thermodynamic loss is small. For high nondimensional speeds, the heat transfer is nearly adiabatic and again the thermodynamic loss is

---

20 Grigull, U., Sandner, H., *Heat Conduction*, Translated by J. Kestin, Springer-Verlag, Hemisphere Publishing Corp., Washington, 1984, pp. 111-117.

small. As the heat transfer increases from isothermal to adiabatic conditions, the thermodynamic loss increases until it reaches a maximum approximately halfway between the two conditions and then begins to decrease again.

The range of nondimensional speeds possible for a given apparatus is dependent on the operating temperature as well as the cycle frequency. At room temperature, the apparatus for this study can attain nondimensional speeds which produce gas spring heat transfer which spans from isothermal to adiabatic operation. To keep the thermodynamic loss due to gas spring heat transfer small at this temperature, a reciprocator should be designed to operate at either end of the isothermal and adiabatic spectrum. At liquid nitrogen temperatures, the nondimensional speeds attainable produce gas spring heat transfer which spans from approximately halfway between isothermal and adiabatic operation to adiabatic operation. Since at this temperature isothermal conditions are not practical, a reciprocator should be designed to operate as close to adiabatic conditions (high nondimensional speed) as possible to minimize the thermodynamic loss due to gas spring heat transfer. At liquid helium temperatures, the nondimensional speeds attainable produce gas spring heat transfer which is always nearly adiabatic. Therefore the cycle frequency has little effect on the thermodynamic loss which is always small. In the design of reciprocators, the total thermodynamic loss should be minimized and the loss due to gas spring heat transfer is only part of this total loss. The experiment work presented in this thesis measured the actual magnitude of the gas spring loss under a wide variety of conditions. This information will be helpful especially at cryogenic temperature conditions where it had not been previously available.

The analytic models showed that even though gas spring heat transfer is affected by many different factors, only a few affect it significantly. The models purposefully neglect many details such as the flow patterns within a gas spring which depend on the exact cylinder geometry, the temperature gradients in the cylinder wall etc. yet they still provide reasonable representations of the gas spring heat transfer. By using simple representations of the heat transfer we were able to include the thermodynamic behavior of the system in the models. The models are then able to provide more accurate representation of the heat transfer over a full cycle. The thermodynamic behavior and the heat transfer were interrelated by using a controlling input volume function representative of the nearly sinusoidal piston motion which is typically the controlling input to a real reciprocator.

The one space model showed that by modeling the gas in the apparatus as a single lump with uniform pressure and temperature, we can approximate the general relations for gas spring heat transfer. This model shows that the gas density, specific heat, thermal conductivity, and piston speed all affect gas spring heat transfer. The two space model adds to the complexity of the one space model by representing the gas in the apparatus as two lumps with independent but uniform temperatures and pressures. This added complexity refined the results of the model especially at low nondimensional speeds. The gap flow and hybrid models show that the effect of the appendix gap, found in many cryogenic reciprocators, can be modeled in a simple fashion by assuming the appendix gap to be purely isothermal. These models show that the size of the appendix gap and its temperature distribution affect the heat transfer to and from the gas and also give some insight into why the pressure swings for many of the liquid helium temperature experiments were greater than true adiabatic pressure swings.

The comparison of the results of the models to the results of the experiments show that the models account for the most important factors of gas spring heat transfer. Even though the new speed correlations presented in section 5.3 were an improvement over the Peclet number correlations, the Peclet number correlations still provided good results when compared with the data. We found that even at liquid helium temperatures, the models still predict the correct behavior of the system even though the cylinder wall properties were neglected. This seems to indicate that the *most important* factors of gas spring heat transfer are related to the gas properties not the cylinder wall properties.

Appendix A
APPARATUS

This appendix presents some important details of the experiment apparatus. A general description of this apparatus has been given in Chapter 3. This appendix discusses the instrumentation of the apparatus, and the hydraulic actuator used to control the piston motion.

A.1 Volume Instrumentation

A linear variable differential transformer (LVDT) is used to measure the volume of the working gas. The LVDT is a system comprised of two major sub-systems. The first is the electromechanical transformer and the second is an electronic circuit. The transformer is made of two solenoidal windings which are held stationary and of a core which is attached to the actuator piston and slides inside the windings. When the primary winding is excited with an AC voltage of fixed magnitude and frequency, an AC voltage of the same frequency is induced in the secondary winding. This induced voltage has an amplitude proportional to the length of the core within the windings. The electronic circuit associated with the LVDT provides the AC excitation of the primary winding and converts the AC voltage level of the secondary winding to a DC voltage proportional to the position of the actuator piston and thus the volume of the working gas.

Calibration of the LVDT consisted of moving the actuator to several positions, recording the output of the LVDT electronics and taking careful measurements with a micrometer of the distance between the piston face and the end of the cylinder. The LVDT output is then correlated with the measured distance. Several calibrations were performed to ascertain the repeatability of the LVDT. Figure A.1 is a linear least squares curve fit for the LVDT. The output appears to be quite linear, however if we look at the error curve, or the difference between the actual data points and the calibration curve as shown by the symbols "x" in fig. A.2 we see that there is a maximum error of approximately 0.9%. By comparing several calibration runs we can see that the repeatability of this device is much better than the linearity. By using a more complex curve fit we can take advantage of the high repeatability. We can reduce the calibration

curve error to almost the resolution of our recording instrumentation (i.e. 0.024 % full scale), by using discrete Fourier transforms. The error curve for this calibration curve is distinguished by the squares in fig A.2. The form of this calibration curve is as follows:

$$\text{OUTPUT} = mx + b + \sum_{k=1}^{N}\left(A_k \cos\left(\frac{2\pi k x}{x_0}\right) + B_k \sin\left(\frac{2\pi k x}{x_0}\right)\right). \tag{A.1}$$

Table A.1 lists coefficients for this equation used in the curve fit shown in fig. A.2. In addition to simply calibrating the LVDT for distance measurements, careful measurements were made of the inside diameter of the cylinder, as well as the clearance volume between the piston and cylinder and also the volume of the capillary tube leading to the strain gage pressure transducer. This information was used to convert the LVDT displacement calibration to a volume calibration.

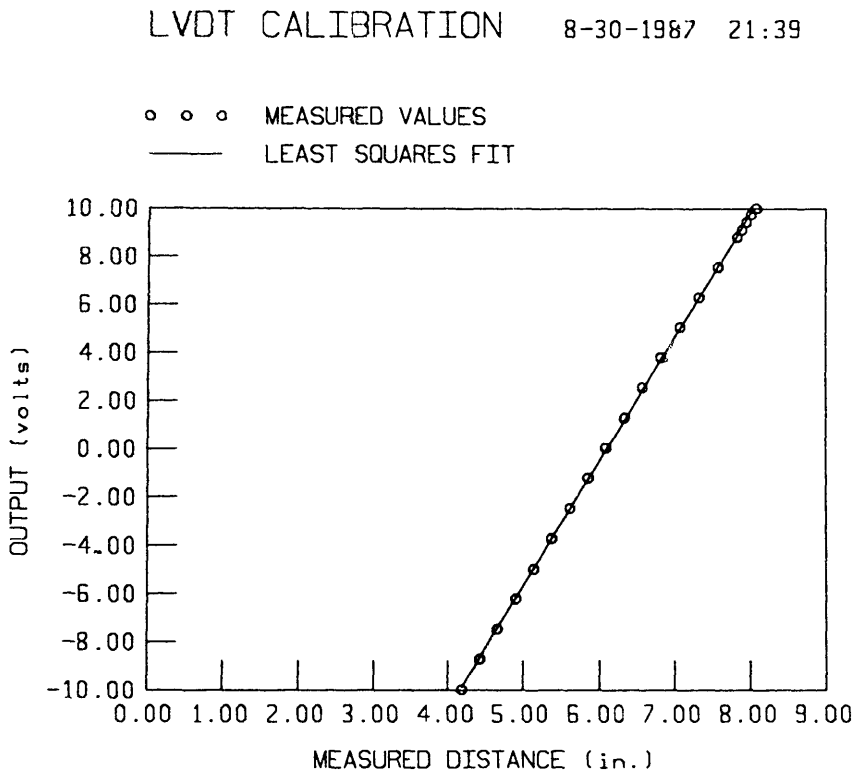

Figure A.1 LVDT Calibration, Linear Curve Fit

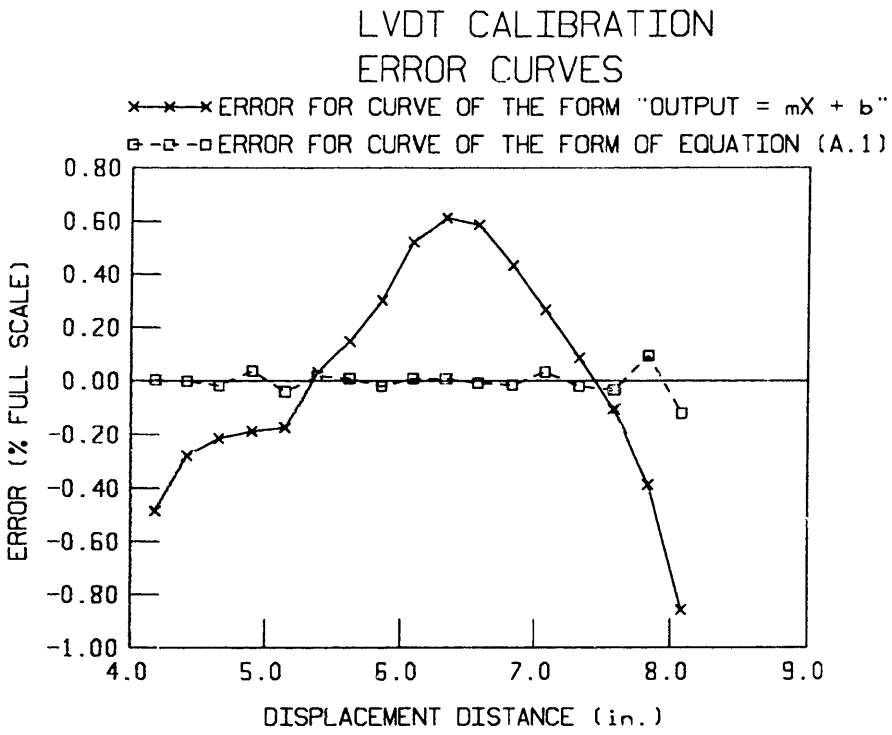

Figure A.2 LVDT Calibration, Error Curves

In addition to the quasi-static performance of the LVDT, the dynamic performance is also important. The specifications for this particular device indicates that for the frequency response, the '*phase lag of (the) demodulator output signal increases linearly from* 0° *to* 180° *as (the) frequency of modulation increases from zero to the carrier frequency*'.[21] This means that if we attempt to measure, with the LVDT, the piston position while it is stationary, there would be no phase lag, but if we attempt to measure the piston position while the piston was oscillating with a frequency equal to the excitation frequency of the LVDT, then the phase lag between the actual position of the piston and that indicated by the LVDT would be 180°.

21 Moog Incorporated, Industrial Division, "MOOG AO85 Servoactuators", *Data Sheet #850 184*, East Aurora, N.Y.

Table A.1 Fourier Coefficients for LVDT Curve Fit

Fourier Coefficients for LVDT Curve Fit		
m = 1053.295		b = -4381.523
k	A_k	B_k
1	-2.224	0.212
2	-7.231	1.394
3	3.272	0.961
4	-0.896	0.359
5	0.013	-0.007
6	-0.990	0.636
7	0.364	-0.286
8	-0.793	0.756
9	0.037	-0.042
10	-0.339	0.476

The original factory configuration of the associated electronics provided an excitation frequency of 400 Hz. Thus for a cycle period of 0.5 seconds there is an intrinsic phase lag of approximately 0.9°. This appears to be small but it is on the order of magnitude of the phases we are attempting to measure. For this reason we modified the circuitry (per recommendations of the manufacturer) to provide an excitation frequency of 6000 Hz. This reduces the phase lag associated with a 0.5 second period from 0.9° to 0.06°. This value is more acceptable but it is compensated for during data reduction.

A.2 Static Pressure Measurement

The transducers used for static pressure measurements were CEC Type 4-313 un-bonded 4-wire strain gage pressure transducers. Two different units were used, one for low pressures (0-250 psia) and one for higher pressures (0-1000 psia). At the recommended excitation of 5 volts, the nominal full scale output for these transducers is 20 mV. The output signal of these transducers were measured directly, that is without

any additional signal conditioning equipment, by the data acquisition system. The sensing face of each pressure transducer was coated with a thin layer of RTV to prevent the cyclic temperature changes of the gas from affecting the sensitivity. These transducers were calibrated with an Amthor type 452 dead weight gauge tester. Figure A.3 is a sample calibration curve for the high range pressure transducer.

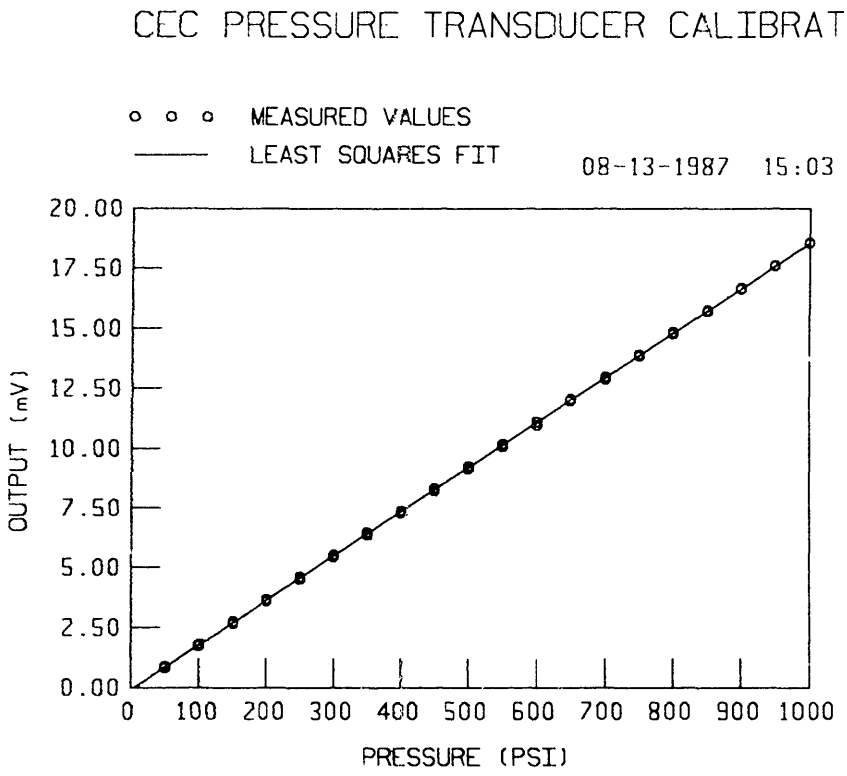

Figure A.3 Sample calibration for static pressure transducer.

A.3 Dynamic Pressure Measurement

The pressure transducer used to make dynamic pressure measurements is a Dytran model 2300C2 piezoelectric pressure sensor. This transducer was chosen primarily for its ability to operate at cryogenic temperatures. This sensor outputs a charge proportional to

the pressure applied to its face. The nominal sensitivity for this transducer is 0.35pC/psi. A special charge amplifier is required to convert its output to a voltage which can be measured by the data acquisition system. The pressure transducer, the charge amplifier and the interconnecting electrical cables comprise a system particularly susceptible to leakage of the charge produced by the transducer due to poor electrical insulation. Therefore the system requires careful handling but some charge leakage will inevitably occur. There is also some charge leakage which occurs inside the transducer. The charge leakage has the effect of *filtering* out the steady part of the pressure signal. This is the primary reason for needing a different transducer for static pressure measurements.

This pressure transducer is mounted in the cylinder head and is therefore subjected to the operating temperatures of the apparatus. In an effort to minimize the effects of the cyclic temperature variation of the working gas, the face of the transducer is coated with a thin layer of silicon grease.

As outlined in Chapter 3, this pressure transducer is calibrated twice for each experiment run. The calibrations are simple four point pressure measurements which rely on the static pressure transducer to act as the standard. These calibrations assume that the output of the piezoelectric transducer is linear between the calibration points, and that its dynamic performance is acceptable.

A.4 Thermocouples

Thermocouples are used to monitor the temperature of the cylinder wall, cylinder head and the heat exchangers during experiment runs and during experiment cool-down. The thermocouples used are type T (copper - constantan) and they are soldered directly to the point of the apparatus where a temperature reading is desired. We use these thermocouples for temperature measurements from room temperature (300 K) to below liquid nitrogen temperature (77 K). The accuracy below 77 K is questionable but at these lower temperatures, they are only used to monitor the cool-down of the apparatus.

The voltage outputs from the thermocouples are converted to temperatures using a fourteenth order power series expansion. The coefficients of this expansion is given in Table A.2.

Table A.2 Power series expansion for the thermoelectric voltage of Type T thermocouples.[22]

Term	Coefficients
T^1	$3.8740773840 \times 10^{1}$
T^2	$4.4123932482 \times 10^{-2}$
T^3	$1.1405238498 \times 10^{-4}$
T^4	$1.9974406568 \times 10^{-5}$
T^5	$9.0445401187 \times 10^{-7}$
T^6	$2.2766018504 \times 10^{-8}$
T^7	$3.6247409380 \times 10^{-10}$
T^8	$3.8648924201 \times 10^{-12}$
T^9	$2.8298678519 \times 10^{-14}$
T^{10}	$1.4281383349 \times 10^{-16}$
T^{11}	$4.8833254364 \times 10^{-19}$
T^{12}	$1.0803474683 \times 10^{-21}$
T^{13}	$1.3949291026 \times 10^{-24}$
T^{14}	$7.9795893156 \times 10^{-28}$

A.5 Germanium Resistance Thermometers

For temperature measurements below 20 K, two germanium resistors are used. These instruments have excellent response at these temperatures. One of these resistors is located on the lower heat exchanger and the second one is located adjacent to the main working space on the cylinder wall . The resistance of these instruments is dependent on their temperature. The functions used in the data reduction for temperature-resistance dependence is given in the following relations:

For resistor #233

22 Powell, "Thermocouple Thermometry", *Measurements in Heat Transfer*, Edited by Eckert & Goldstein, Hemisphere Publishing Corp., Washington, (Springer-Verlag), pg. 147.

$$T = 10^{\left(1.662 - (\log R)^{1}/3\right)/0.3742} \qquad \text{(A.2)}$$

For resistor #1005

$$T = 10^{\left(1.674 - (\log R)^{1}/3\right)/0.3748} \qquad \text{(A.3)}$$

These equations have been empirically derived with the aid of calibration work done by Bowman.[23] The resistance of these sensors is determined by supplying a known current through them and measuring the induced voltage across them.

A.6 Hydraulic Actuator

The hydraulic actuator used to control the apparatus' piston motion is a MOOG® model AO85 1 FF 4 servoactuator with a MOOG® model 760-103 servovalve. This actuator has a piston area of 1.1 in^2, a 4 in stroke, and is rated for pressures up to 3000 psi. The servoamplifier used is a MOOG® model 121-132 servocontroller. The LVDT which is used for both actuator piston control and for gas volume measurements, is part of the hydraulic actuator assembly.

---

23 Bowman, H.F., *Influence of Nuclear Radiation on Pool-Boiling Heat Transfer to Liquid Helium*, Ph.D. Thesis, Department of Nuclear Engineering, M.I.T., 1968.

Appendix B
EXPERIMENT DATA

This appendix presents the results of the gas spring heat transfer experiments in tabular form. Because of the large number of paramters of interest, each table has been presented in three parts. The first part presents information regarding the controllable input paramters. The second part presents the results in dimensional form which includes both volume and pressure information. The volume data is given in units of $in.^3$, and the pressure is given in units of psia. The last part of each table presents the nondimensional pressure data. Tables B.1a,b,c present the data collected at room temperature. Tables B.2a,b,c present the data collected at liquid nitrogen temperature, and Tables B.3a,b,c present the data collected at liquid helium temperature.

Table B.1a Room temperature experiment data.

File #	Period (seconds)	Mass (grams)	Peclet #	$\sqrt{\rho C_p k}$ (J/s$^{1/2}$-m^2-K)
PV1103B.DAT	8	.0305	7.1	11.4
PV1103A.DAT	16	.0305	3.5	11.4
PV1103C.DAT	4	.0305	14.1	11.4
PV1103D.DAT	2	.0305	28.2	11.4
PV1103E.DAT	1	.0305	56.4	11.4
PV1103F.DAT	.5	.0305	112.8	11.4
PV1113A.DAT	16	.0307	3.5	11.5
PV1113B.DAT	8	.0307	7.1	11.5
PV1113C.DAT	4	.0307	14.2	11.5
PV1113D.DAT	2	.0307	28.3	11.5
PV1113E.DAT	1	.0307	56.7	11.5
PV1113F.DAT	.5	.0307	113.4	11.5
PV1113G.DAT	16	.0651	7.5	16.7
PV1113H.DAT	8	.0651	15.0	16.7
PV1113I.DAT	4	.0651	30.1	16.7
PV1113J.DAT	2	.0651	60.2	16.7
PV1113K.DAT	1	.0651	120.3	16.7
PV1113L.DAT	.5	.0651	240.7	16.7
PV0215A.DAT	16	.2811	32.5	34.7
PV0215B.DAT	8	.2811	65.0	34.7
PV0215C.DAT	4	.2811	129.9	34.7
PV0215D.DAT	2	.2811	259.8	34.7
PV0215E.DAT	1	.2811	519.6	34.7
PV0215F.DAT	.5	.2811	1039.2	34.7
PV0215G.DAT	16	.6408	74.0	52.4
PV0215H.DAT	8	.6408	148.1	52.4
PV0215I.DAT	4	.6408	296.2	52.4
PV0215J.DAT	2	.6408	592.3	52.4
PV0215K.DAT	1	.6408	1184.7	52.4
PV0215L.DAT	.5	.6408	2369.4	52.4

Table B.1b Room temperature experiment data.

File #	P_1^*	P_2^*	θ_1^* (degrees)	θ_2^* (degrees)
PV1103B.DAT	.096	.101	4.88	4.56
PV1103A.DAT	.051	.100	1.90	2.79
PV1103C.DAT	.212	.210	7.82	6.02
PV1103D.DAT	.442	.363	9.61	6.49
PV1103E.DAT	.597	.479	8.13	4.85
PV1103F.DAT	.689	.593	6.91	4.50
PV1113A.DAT	.028	.0323	2.13	3.40
PV1113B.DAT	.060	.123	4.80	4.99
PV1113C.DAT	.149	.285	7.57	5.58
PV1113D.DAT	.311	.223	9.64	4.88
PV1113E.DAT	.534	.397	8.19	5.01
PV1113F.DAT	.639	.491	7.02	4.53
PV1113G.DAT	.0038	.180	3.92	5.45
PV1113H.DAT	.105	.242	7.87	6.21
PV1113I.DAT	.278	.399	10.67	6.09
PV1113J.DAT	.513	.539	9.34	5.49
PV1113K.DAT	.680	.611	6.61	3.92
PV1113L.DAT	.731	.697	5.31	3.29
PV0215A.DAT	.172	.190	9.80	7.11
PV0215B.DAT	.394	.090	12.85	7.10
PV0215C.DAT	.622	.298	11.01	7.31
PV0215D.DAT	.820	.344	7.02	4.29
PV0215E.DAT	.837	.398	4.61	2.72
PV0215F.DAT	.851	.157	3.63	2.15
PV0215G.DAT	.331	.157	11.46	6.19
PV0215H.DAT	.603	.196	11.94	5.88
PV0215I.DAT	.731	.356	8.64	4.55
PV0215J.DAT	.790	.320	5.62	2.74
PV0215K.DAT	.865	.412	4.05	2.83
PV0215L.DAT	.929	.241	4.20	2.43

Table B.1c Room temperature experiment data.

File #	V_0 (in³)	V_m (in³)	P_0 (psia)	P_1 (psia)	P_2 (psia)
PV1103B.DAT	11.44	3.21	15.6	4.6	.65
PV1103A.DAT	11.43	3.21	15.4	4.5	.59
PV1103C.DAT	11.44	3.21	15.7	4.9	.74
PV1103D.DAT	11.44	3.21	15.6	5.5	.83
PV1103E.DAT	11.44	3.21	15.5	5.9	.91
PV1103F.DAT	11.44	3.21	15.7	6.3	.97
PV1113A.DAT	11.44	3.21	15.5	4.4	.60
PV1113B.DAT	11.44	3.21	15.7	4.6	.65
PV1113C.DAT	11.44	3.21	15.8	4.7	.74
PV1113D.DAT	11.44	3.21	15.8	5.2	.80
PV1113E.DAT	11.44	3.21	16.0	5.9	.91
PV1113F.DAT	11.44	3.21	16.0	6.2	.97
PV1113G.DAT	11.44	3.21	33.2	9.2	1.39
PV1113H.DAT	11.44	3.21	33.3	9.8	1.61
PV1113I.DAT	11.44	3.21	33.9	10.9	1.77
PV1113J.DAT	11.44	3.21	34.1	12.4	2.0
PV1113K.DAT	11.44	3.21	33.7	13.3	2.2
PV1113L.DAT	11.44	3.21	33.8	13.7	2.3
PV0215A.DAT	11.44	3.21	141.1	42.5	6.4
PV0215B.DAT	11.44	3.21	142.9	48.1	7.2
PV0215C.DAT	11.44	3.21	144.4	54.8	8.1
PV0215D.DAT	11.44	3.21	141.7	59.6	9.2
PV0215E.DAT	11.44	3.21	144.2	61.4	9.6
PV0215F.DAT	11.44	3.21	145.3	63.2	9.5
PV0215G.DAT	11.44	3.21	314.9	103.3	12.2
PV0215H.DAT	11.44	3.21	310.6	117.3	13.7
PV0215I.DAT	11.44	3.21	314.9	128.2	15.2
PV0215J.DAT	11.44	3.21	313.9	133.1	16.0
PV0215K.DAT	11.44	3.21	313.3	137.2	16.6
PV0215L.DAT	11.44	3.21	310.1	140.8	17.1

Table B.2a Liquid nitrogen temperature experiment data.

File #	Period (seconds)	Mass (grams)	Peclet #	$\sqrt{\rho C_p k}$ (J/s$^{1/2}$-m^2-K)
PV0101A.DAT	16	.8817	101.9	61.5
PV0101B.DAT	8	.8817	203.7	61.5
PV0101D.DAT	2	.8817	814 9	61.5
PV0101C.DAT	4	.8817	407.5	61.5
PV0101E.DAT	1	.8817	1630	61.5
PV0101F.DAT	.5	.8817	3260	61.5
PV0101G.DAT	16	.5597	७4.7	49.0
PV0101H.DAT	8	.5597	129.3	49.0
PV0101I.DAT	4	.5597	258.7	49.0
PV0101J.DAT	2	.5597	517.4	49.0
PV0101K.DAT	1	.5597	1034	49.0
PV0101L.DAT	.5	.5597	2069	49.0
PV0101M.DAT	16	.2768	32.0	34.4
PV0101N.DAT	8	.2768	64.0	34.4
PV0101O.DAT	4	.2768	127.9	34.4
PV0101P.DAT	2	.2768	255.8	34.4
PV0101Q.DAT	1	.2768	511.7	34.4
PV0101R.DAT	.5	.2768	1023	34.4
PV0101S.DAT	16	1.596	184.4	82.7
PV0101T.DAT	8	1.596	368.8	82.7
PV0101U.DAT	4	1.596	737.6	82.7
PV0101V.DAT	2	1.596	1475	82.7
PV0101W.DAT	1	1.596	2950	82.7

Table B.2b Liquid nitrogen temperature experiment data.

File #	P_1^*	P_2^*	θ_1^* (degrees)	θ_2^* (degrees)
PV0101A.DAT	.578	.371	10.10	7.52
PV0101B.DAT	.825	.517	9.09	6.01
PV0101D.DAT	.943	.703	3.68	2.34
PV0101C.DAT	.917	.709	6.02	4.19
PV0101E.DAT	.936	.699	2.60	1.61
PV0101F.DAT	.998	.729	2.13	1.29
PV0101G.DAT	.556	.462	10.42	7.21
PV0101H.DAT	.758	.445	10.14	6.36
PV0101I.DAT	.839	.766	6.77	4.95
PV0101J.DAT	.922	.760	4.30	2.81
PV0101K.DAT	.914	.747	3.08	1.90
PV0101L.DAT	.968	.762	2.67	1.64
PV0101M.DAT	.250	.214	8.71	6.55
PV0101N.DAT	.488	.364	10.10	7.05
PV0101O.DAT	.823	.598	8.70	5.77
PV0101P.DAT	.768	.637	5.518	3.91
PV0101Q.DAT	.917	.762	3.74	2.37
PV0101R.DAT	.903	.778	3.04	1.86
PV0101S.DAT	.641	-.109	9.24	5.81
PV0101T.DAT	.875	.039	7.70	4.71
PV0101U.DAT	.959	.192	4.94	3.09
PV0101V.DAT	.983	.167	2.97	1.90
PV0101W.DAT	.973	.174	2.04	1.20

Table B.2c Liquid nitrogen temperature experiment data.

File #	V_0 (in³)	V_m (in³)	P_0 (psia)	P_1 (psia)	P_2 (psia)
PV0101A.DAT	11.44	3.21	134.5	50.3	7.99
PV0101B.DAT	11.44	3.21	136.9	56.8	9.14
PV0101D.DAT	11.44	3.21	138.3	61.9	10.54
PV0101C.DAT	11.44	3.21	138.3	60.7	10.08
PV0101E.DAT	11.44	3.21	139.4	63.2	10.77
PV0101F.DAT	11.44	3.21	138.9	64.4	10.91
PV0101G.DAT	11.44	3.21	85.4	31.5	5.14
PV0101H.DAT	11.44	3.21	88.1	35.4	5.80
PV0101I.DAT	11.44	3.21	91.5	38.8	6.58
PV0101J.DAT	11.44	3.21	94.6	42.0	7.17
PV0101K.DAT	11.44	3.21	100.1	44.8	7.66
PV0101L.DAT	11.44	3.21	108.2	49.3	8.43
PV0101M.DAT	11.44	3.21	43.6	13.9	2.25
PV0101N.DAT	11.44	3.21	43.8	15.6	2.53
PV0101O.DAT	11.44	3.21	43.6	18.2	2.97
PV0101P.DAT	11.44	3.21	44.5	18.4	3.10
PV0101Q.DAT	11.44	3.21	44.4	19.9	3.37
PV0101R.DAT	11.44	3.21	44.7	19.8	3.41
PV0101S.DAT	11.44	3.21	254.2	98.6	13.30
PV0101T.DAT	11.44	3.21	255.6	109.9	15.08
PV0101U.DAT	11.44	3.21	255.0	115.4	16.32
PV0101V.DAT	11.44	3.21	253.4	116.9	16.75
PV0101W.DAT	11.44	3.21	252.7	117.8	16.99

Table B.3a Liquid helium temperature experiment data.

File #	Period (seconds)	Mass (grams)	Peclet #
PV0213A.DAT	16	1.249	1094.9
PV0213B.DAT	8	1.249	2141.6
PV0213C.DAT	4	1.249	4206.4
PV0213D.DAT	2	1.249	8203.4
PV0213E.DAT	1	1.249	17345.1
PV0213F.DAT	.5	1.249	22797.9
PV0213G.DAT	.5	1.249	37336.5
PV0213H.DAT	16	.636	498.5
PV0213I.DAT	8	.636	974.1
PV0213J.DAT	4	.636	1920.7
PV0213K.DAT	2	.636	3972.9
PV0213L.DAT	1	.636	8152.2
PV0213M.DAT	.5	.636	17341.4
PV0218A.DAT	.5	7.214	147141.8
PV0218B.DAT	16	7.214	7057.8
PV0218C.DAT	8	7.214	13874.2
PV0218D.DAT	4	7.214	27328.8
PV0218E.DAT	2	7.214	52633.9
PV0218F.DAT	1	7.214	93913.9
PV0218G.DAT	16	6.173	6771.9
PV0218H.DAT	8	6.173	12868.1
PV0218I.DAT	4	6.173	24817.7
PV0218J.DAT	2	6.173	46950.7
PV0218K.DAT	1	6.173	83348.5
PV0218L.DAT	.5	6.173	143681.7
PV0218M.DAT	1	6.173	82920.5
PV0218N.DAT	.5	6.173	143893.9

Table B.3a (cont.) Liquid helium temperature experiment data.

File #	Period (seconds)	Mass (grams)	Peclet #
PV0218O.DAT	16	2.805	2972.0
PV0218P.DAT	8	2.805	5924.2
PV0218Q.DAT	4	2.805	11695.8
PV0218R.DAT	2	2.805	22189.5
PV0218S.DAT	1	2.805	39286.6
PV0218T.DAT	.5	2.805	68586.8
PV0218U.DAT	16	1.633	1681.9
PV0218V.DAT	8	1.633	3307.2
PV0218W.DAT	4	1.633	6564.2
PV0218X.DAT	2	1.633	12570.6
PV0218Y.DAT	1	1.633	23368.0
PV0218Z.DAT	.5	1.633	40318.1
PV0218AA.DAT	16	1.117	1232.6
PV0218BA.DAT	8	1.117	2317.1
PV0218CA.DAT	4	1.117	4552.6
PV0218DA.DAT	2	1.117	8853.7
PV0218EA.DAT	1	1.117	16914.8
PV0218FA.DAT	.5	1.117	31490.8

Table B.3b Liquid helium temperature experiment data.

File #	P_1^*	P_2^*	θ_1^* (degrees)	θ_2^* (degrees)
PV0213A.DAT	.879	.771	1.41	3.99
PV0213B.DAT	.916	.877	2.36	2.60
PV0213C.DAT	.944	.907	2.67	1.73
PV0213D.DAT	1.087	1.008	2.66	1.60
PV0213E.DAT	1.269	1.140	1.66	.92
PV0213F.DAT	.967	.626	1.61	.96
PV0213G.DAT	1.170	1.090	1.17	.67
PV0213H.DAT	.581	.620	3.65	3.97
PV0213I.DAT	.605	.698	5.24	2.81
PV0213J.DAT	.799	.863	4.75	2.82
PV0213K.DAT	.828	.879	3.53	1.77
PV0213L.DAT	.892	.914	2.25	1.24
PV0213M.DAT	.961	.979	1.57	1.33
PV0218A.DAT	.955	.712	.29	-.023
PV0218B.DAT	1.184	1.140	1.53	.95
PV0218C.DAT	1.149	1.130	.938	.47
PV0218D.DAT	1.154	1.110	.490	-.10
PV0218E.DAT	1.139	1.090	.193	-.16
PV0218F.DAT	1.156	1.080	.231	-.26
PV0218G.DAT	1.109	1.072	1.782	1.20
PV0218H.DAT	1.074	1.030	1.038	0.58
PV0218I.DAT	1.065	1.020	.673	0.20
PV0218J.DAT	1.060	1.000	.513	0.00
PV0218K.DAT	1.029	.845	.579	-.15
PV0218L.DAT	.938	.624	.526	-.12
PV0218M.DAT	1.073	1.000	.594	-.02
PV0218N.DAT	1.029	.891	.520	.04

Table B.3b (cont.) Liquid helium temperature experiment data.

File #	P_1^*	P_2^*	θ_1^* (degrees)	θ_2^* (degrees)
PV0218O.DAT	1.107	1.071	1.70	1.79
PV0218P.DAT	1.099	1.083	1.32	1.07
PV0218Q.DAT	1.119	1.090	1.28	.77
PV0218R.DAT	1.135	1.090	1.19	.61
PV0218S.DAT	1.159	1.090	1.19	.55
PV0218T.DAT	1.124	1.060	1.01	.26
PV0218U.DAT	1.030	1.010	1.21	1.53
PV0218V.DAT	1.059	1.070	1.61	1.14
PV0218W.DAT	1.103	1.100	1.77	.98
PV0218X.DAT	1.137	1.100	1.78	.48
PV0218Y.DAT	1.197	1.140	1.55	.53
PV0218Z.DAT	1.129	1.080	1.12	.54
PV0218AA.DA	.884	.911	2.02	1.41
PV0218BA.DA	.900	.958	2.34	1.80
PV0218CA.DA	.963	1.032	2.49	1.48
PV0218DA.DA	1.045	1.110	2.26	.94
PV0218EA.DA	1.107	1.115	1.87	1.01
PV0218FA.DA	1.132	1.130	1.40	.56

Table B.3c Liquid helium temperature experiment data.

File #	V_0 (in³)	V_m (in³)	P_0 (psia)	P_1 (psia)	P_2 (psia)
PV0213A.DAT	11.44	3.21	28.6	12.9	2.16
PV0213B.DAT	11.44	3.21	29.7	13.6	2.42
PV0213C.DAT	11.44	3.21	30.6	14.2	2.54
PV0213D.DAT	11.44	3.21	32.2	15.8	2.83
PV0213E.DAT	11.44	3.21	29.9	15.5	2.81
PV0213F.DAT	11.44	3.21	58.3	27.5	4.03
PV0213G.DAT	11.44	3.21	26.0	13.2	2.41
PV0213H.DAT	11.44	3.21	17.6	6.8	1.17
PV0213I.DAT	11.44	3.21	18.3	7.1	1.29
PV0213J.DAT	11.44	3.21	18.9	8.0	1.48
PV0213K.DAT	11.44	3.21	17.8	7.7	1.41
PV0213L.DAT	11.44	3.21	17.1	7.6	1.38
PV0213M.DAT	11.44	3.21	15.4	7.1	1.31
PV0218A.DAT	11.44	3.21	283.8	155.3	27.63
PV0218B.DAT	11.44	3.21	136.9	81.8	18.41
PV0218C.DAT	11.44	3.21	138.1	82.9	18.81
PV0218D.DAT	11.44	3.21	141.4	85.5	19.11
PV0218E.DAT	11.44	3.21	151.2	90.8	19.98
PV0218F.DAT	11.44	3.21	189.1	111.9	23.61
PV0218G.DAT	11.44	3.21	94.0	52.1	11.16
PV0218H.DAT	11.44	3.21	100.8	56.3	12.02
PV0218I.DAT	11.44	3.21	107.1	56.0	12.75
PV0218J.DAT	11.44	3.21	118.9	66.5	13 96
PV0218K.DAT	11.44	3.21	148.6	81.0	15.00
PV0218L.DAT	11.44	3.21	189.0	100.4	16.21
PV0218M.DAT	11.44	3.21	150.8	83.7	16.99
PV0218N.DAT	11.44	3.21	192.6	105.0	20.11

Table B.3c (cont.) Liquid helium temperature experiment data.

File #	V_0 (in³)	V_m (in³)	P_0 (psia)	P_1 (psia)	P_2 (psia)
PV0218O.DAT	11.44	3.21	47.1	23.7	4.51
PV0218P.DAT	11.44	3.21	46.6	23.7	4.57
PV0218Q.DAT	11.44	3.21	47.5	24.5	4.73
PV0218R.DAT	11.44	3.21	52.1	27.0	5.20
PV0218S.DAT	11.44	3.21	64.8	33.7	6.40
PV0218T.DAT	11.44	3.21	80.5	41.8	7.90
PV0218U.DAT	11.44	3.21	28.5	13.9	2.57
PV0218V.DAT	11.44	3.21	29.7	14.5	2.76
PV0218W.DAT	11.44	3.21	30.1	15.0	2.83
PV0218X.DAT	11.44	3.21	32.4	16.3	3.07
PV0218Y.DAT	11.44	3.21	37.0	18.9	3.52
PV0218Z.DAT	11.44	3.21	46.3	23.4	4.35
PV0218AA.DA	11.44	3.21	18.0	8.0	1.46
PV0218BA.DA	11.44	3.21	19.6	8.8	1.66
PV0218CA.DA	11.44	3.21	20.3	9.4	1.79
PV0218DA.DA	11.44	3.21	21.4	10.2	1.98
PV0218EA.DA	11.44	3.21	23.2	11.3	2.13
PV0218FA.DA	11.44	3.21	25.7	12.8	2.42

Appendix C
HEAT TRANSFER CALCULATION

In this appendix we solve for the gas spring heat transfer for a specific experiment run. The general method used in this solution is presented in Chapter 5. We now apply it to the specific case of experiment run # PV0215C.DAT. This is a room temperature run with a cycle period of 8 seconds. The parameters necessary to calculate the heat transfer for this run and all other experiment runs is presented in Appendix B.

Since we have used a two harmonic Fourier series to represent the pressure and a one harmonic Fourier series to represent the volume, we retain only these harmonics in the equations 5.4 through 5.7. We then substitute these equations into equation 5.3 to get the following equation:

$$\dot{Q}=\frac{-\omega\gamma}{\gamma-1}V_m\sin(\omega t)\left\{P_0+P_1\cos\left(\omega t+\frac{\theta_1^*\pi}{180}+\pi\right)\right\}+\frac{-\omega}{\gamma-1}\left\{V_0P_1\sin\left(\omega t+\frac{\theta_1^*\pi}{180}+\pi\right)+2V_0P_2\sin\left(2\left(\omega t+\frac{\theta_2^*\pi}{180}+\pi\right)\right)+V_mP_1\cos(\omega t)\sin\left(\omega t+\frac{\theta_1^*\pi}{180}+\pi\right)\right\} \qquad \text{(c.1)}$$

In this equation we have removed the highest order terms (i.e. the terms involving the product of the first harmonic volume and second harmonic pressure functions). This actually increases the accuracy of the calculation.

Figure C.1 shows the heat transfer predicted by equation C.1 for the experiment run # PV0215C.DAT. Using the parameters directly from Table B.1 gives results in the units lbf-in/sec. These units are somewhat obscure so the data for Figure C.1 has been converted to Watts. We point out that we have defined heat flux *to* the gas as being *positive*. Equation C.1 may used to calculate gas spring heat transfer for any of the experiment runs by simply substituting the correct parameters from Appendix B.

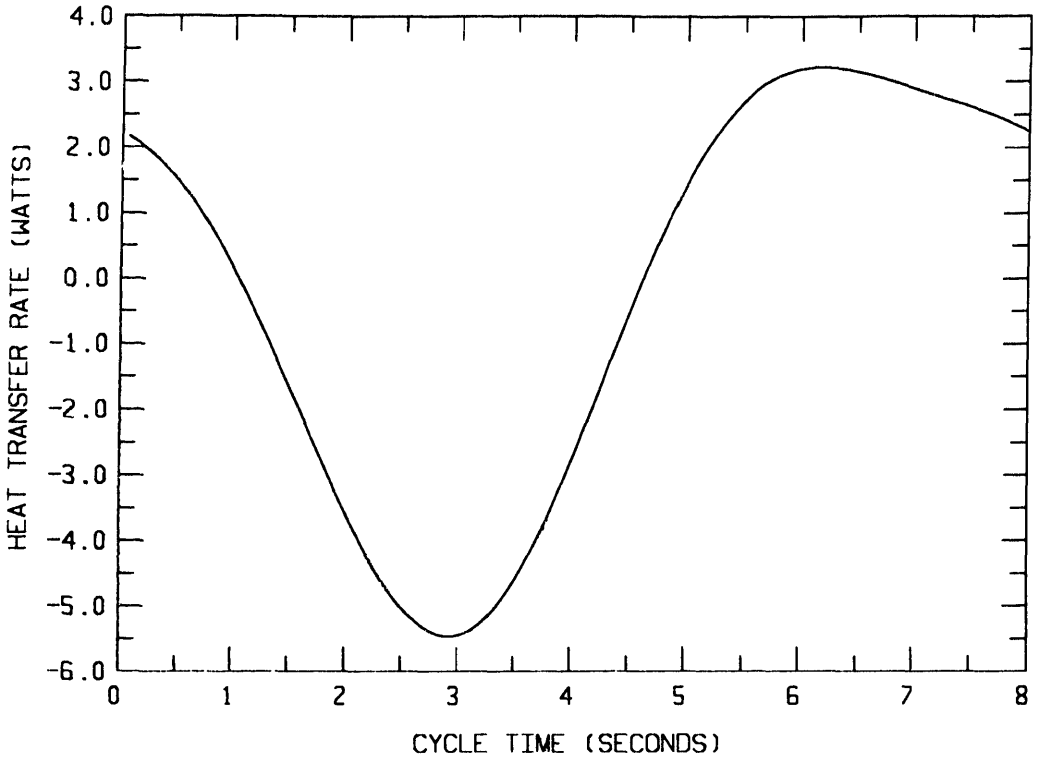

Figure C.1 Heat transfer for run # PV0215C.DAT

Appendix D
COMPUTER FACILITIES & SOFTWARE

The experiment data collection and data reduction were all performed on a *Compaq® Portable II* personal computer. The data collection was accomplished with the aid of two *Data Translation®* model 2801A data acquisition boards installed in the computer and an *ACRO SYSTEMS®* model 900 data acquisition system.

The data acquisition program which controls the two data acquisition systems is written in Microsoft® QuickBasic ver. 4.0 in conjunction with Data Translation® PCLAB software used for communication with the *Data Translation®* model 2801A data acquisition boards. Communication with the *ACRO SYSTEMS®* model 900 data acquisition system is accomplished by using its own command language over an asynchronous RS-232 (serial) port.

The data reduction program is written in Microsoft® FORTRAN ver. 4.0. It accesses graphic routines from Heartland Software's HGRAPH ver.4.1 graphics library.

D.1 Discussion Of Programs

The following is a brief description of the different computer programs used to collect and reduce the experiment data.

D.1.1 Cool Down

COOLDWN.BAS, written in Microsoft® QuickBasic ver. 4.0, is the program used to monitor and record the temperature of the cylinder during the cooldown for the liquid nitrogen and liquid helium temperature range experiment runs. The temperatures measured by the thermocouples and germanium resistors, when used, are recorded at 30 sec intervals.

D.1.2 Mass Fill

The program MFILL.BAS is primarily used to measure and record the pressure, temperature and volume of the gas in the cylinder just prior to a series of experiment runs. The data are used to calculate the mass of the gas in the cylinder. This program can also be used to purge all air from the cylinder and to monitor the pressure in the cylinder.

D.1.3 Experiment Run

The program ACQDAT.BAS is used to control the piston motion of the apparatus and to record all the data of an experiment run. This program is written so that several data runs, each with different cycle periods, will be performed automatically. For each experiment run, first the piezoelectric pressure transducer is calibrated, then the cylinder volume control signal is sent to the hydraulic actuator and after several cycles the volume control signal is adjusted to provide a sinusoidal volume wave. The pressure of the gas is monitored over several cycles until the pressure at the start of a cycle is nearly equal to the pressure at the end of the cycle. This is used as an indication as to whether the apparatus has reached cyclic steady state. Once steady cyclic state has been reached, both the pressure and volume signals are sampled, at a rate of 128 samples per cycle, and recorded for three complete cycles. After the data collection, the piezoelectric pressure transducer is again calibrated. All the data for each run, including the last measurement of the mass of the gas in the cylinder and the cycle period, is recorded into its own individual data file for subsequent data reduction.

D.1.4 Data Reduction

The reduction of the data collected for a run is quite complex and involved. This task is performed by using either the program DRIFT1.FOR or DRIFTH.FOR. They are for the most part identical except that the former uses ideal gas relations to evaluate the thermodynamic properties of the gas while the latter uses real gas properties. The real gas properties are evaluated by subroutines from HEPROP.FOR modified from its original form.[24]

In the reduction of the data the following steps are performed. The piezoelectric pressure transducer's calibrations are checked for consistency and then used to convert the raw pressure data to real pressures. The raw volume data is converted to real volume data using information from manually performed volume calibrations. A running average routine is used to remove any long term drift, due to instrumentation limitations, from the pressure waves. Both the pressure and volume waves are processed by a fast Fourier transform routine and then transformed into a Fourier

---

24 McCarty, R.D., "Interactive Fortran IV Computer Programs for the Thermodynamic and Transport Properties of Selected Cryogens (Fluids Pack)", NBS Technical Note 1025, U.S. Government Printing Office, Washington D.C., 1980.

series of the desired form. Once this processing is complete, the programs provide the ability to display the results in several different graphical forms and also allows for the logging of the results onto a data file which contains the accumulated results of all previously reduced experiment runs.

Many of the subroutines required for the program DRIFTH.FOR are completely identical to those required for DRIFT1.FOR and are therefore listed only once.

D.2 Program Listings

Complete listings of the source code for each of the above programs are presented below. These listings have not been retyped for presentation in this thesis but are the unaltered listings used to create executable code.

D.3 COOLDWN.BAS

```
REM $TITLE: 'PROGRAM COOLDWN.BAS'
REM $SUBTITLE: 'PROGRAM TO MONITOR AND RECORD TEMP. INST.'
REM $PAGESIZE:66
      DIM CEC(30),TEMP(8),EMF(8),VOUT%(4096),VOL%(1)
      KEY(10) ON
      KEY(9) ON
      ON KEY(10) GOSUB DONE
      ON KEY(9) GOSUB DONE
      OPEN "CLDWN.DAT" FOR OUTPUT AS #1
      OPEN "COM1:9600,N,8,2,CS,DS" AS #2
      PRINT "RESETING ACRO-900"
      PRINT #2, CHR$(24)
      TIME=TIMER+8
WAITT: IF TIME>=TIMER THEN GOTO WAITT
HERE: IF NOT EOF(2) THEN X$=INPUT$(1,#2): GOTO HERE
      TIME=TIMER+30
      DO WHILE TIMER < TIME
      PRINT #2, "#4,CJR1,2,3,5,6,7,8=0;CJR4=1"
      PRINT #2, "#4,TYPE=T;DEG=K;MVIN2,3;MVIN4;MVIN5,6,7,8"
      LOCATE 1,2
      INPUT #2, EMF(2),EMF(3),EMF(4)
      PRINT EMF(2),EMF(3),EMF(4)
      FOR I=5 TO 8
      INPUT #2, EMF(I)
      EMF(I) = (EMF(I)+EMF(4))*1000.0
      CALL THRMCPL(EMF(I),TEMP(I))
      PRINT I,TEMP(I)
      NEXT I
      LOOP
      PRINT #1, TIME,EMF(2),EMF(3),EMF(4),TEMP(5),TEMP(6),TEMP(7),TEMP(8)
      GOTO HERE
DONE:   END
REM $TITLE: 'THRMCPL.BAS  ** REQUIRES DATA FILE "THERM.DAT" **'
REM $SUBTITLE: 'SUBROUTINE TO CONVERT TYPE T THERMOCOUPLE EMFs TO DEGREES K'
REM $PAGESIZE: 66
SUB THRMCPL(EMF,T) STATIC
      IF A% = 1 GOTO SKIP
      OPEN "THERM.DAT" FOR RANDOM ACCESS READ AS #10 LEN=4
      A% = 1
      FIELD #10, 4 AS TEMP$
SKIP:   N%=-EMF+2501
      GET #10,N%
      T=CVS(TEMP$)
      END SUB
```

```
REM $TITLE: 'THERMOCOUPLE (TYPE T) SUBROUTINE'
REM $SUBTITLE: 'PROGRAM TO CREATE THERMOCOUPLE LOOKUP TABLE'
REM $PAGESIZE: 66
'SUB THRMCPL(EMF,T)
      KEY(10) ON
      ON KEY(10) GOSUB FIN
      OPEN "THERM.DAT" FOR RANDOM AS #1 LEN = 4
      FIELD #1, 4 AS TEMP$
      DIM A#(14),B#(8)
      DATA     3.8740773840D+01
      DATA     4.4123932482D-02
      DATA     1.1405238498D-04
      DATA     1.9974406568D-05
      DATA     9.0445401187D-07
      DATA     2.2766018504D-08
      DATA     3.6247409380D-10
      DATA     3.8648924201D-12
      DATA     2.8298678519D-14
      DATA     1.4281383349D-16
      DATA     4.8833254364D-19
      DATA     1.0803474683D-21
      DATA     1.3949291026D-24
      DATA     7.9795893156D-28
      FOR I=1 TO 14
           READ A#(I)
      NEXT I
      DATA     3.8740773840D+01
      DATA     3.3190198092D-02
      DATA     2.0714183645D-04
      DATA    -2.1945834823D-06
      DATA     1.1031900550D-08
      DATA    -3.0927581898D-11
      DATA     4.5653337165D-14
      DATA    -2.7616878040D-17
      FOR I= 1 TO 8
           READ B#(I)
      NEXT I
      FOR EMF=2500 TO 1 STEP -1
      DO
           E=0
           DEDT=0
           T=TT
           FOR I=1 TO 8
                E    = E + B#(I)*T^I
                DEDT = DEDT + B#(I)*T^(I-1)
           NEXT I
           DELTAE = EMF - E
           TT = T+DELTAE/DEDT
      LOOP UNTIL ABS(DELTAE)<0.1
      LOCATE 1,1
      N%=-EMF+2501
      PRINT EMF
      T = T +273.15
      LSET TEMP$=MKS$(T)
      PUT #1,N%
   NEXT EMF
      FOR EMF =0 TO -6257 STEP -1
      DO
           E=0
           DEDT=0
           T=TT
           FOR I=1 TO 14
                E    = E + A#(I)*T^I
                DEDT = DEDT + A#(I)*T^(I-1)
           NEXT I
           DELTAE=EMF-E
           TT = T + DELTAE/DEDT
      LOOP UNTIL ABS(DELTAE)<0.35
      LOCATE 1,1
      PRINT EMF
      T=T+273.15
      N%=-EMF+2501
      LSET TEMP$=MKS$(T)
      PUT #1,N%
   NEXT EMF
      GOTO FIN
ORANGE: T= 0.0
FIN:    END
```

D.4 MFILL.BAS

```
REM $TITLE: 'PROGRAM MFILL.BAS'
REM $SUBTITLE:'ADUST AND RECORD TOTAL MASS IN THE SYSTEM'
REM $PAGESIZE: 66
      DIM CEC(30),TEMP(8,2),EMF(8),VOUT%(4096),VOL%(1)
      KEY(10) ON
      KEY(9) ON
      ON KEY(10) GOSUB FINISH
      ON KEY(9) GOSUB DONE
      OPEN "MASS.DAT" FOR APPEND AS #1:' STORE MASS INFORMATION
      OPEN "COM1:9600,N,8,2,CS,DS" AS #2
      OPEN "CAL.DAT" FOR INPUT AS #3:'PRESSURE CALIBRATION COEFFICIENTS
      CALL XSB(1)
      CALL XSCD(900)
      CALL XSD(0,1)
PINPUT: DAAT$=INPUT$(12,#3)
      INPUT #3, CSENS,OFFSET,CECSTDERR
      IF NOT EOF(3) THEN GOTO PINPUT
      PRINT "RESETING ACRO-900"
      PRINT #2, CHR$(24)
      TIME=TIMER+8
      CALL XDV(1,2047)
      INPUT "OPEN HYDRAULIC VALVE SLOWLY, THEN HIT ENTER",DUMY
WAITT: IF TIME>=TIMER THEN GOTO WAITT
HERE: IF NOT EOF(2) THEN X$=INPUT$(1,#2): GOTO HERE
      PRINT #2, "#4,CJR1,2,3,5,6,7,8=0;CJR4=1"
      PRINT #2, "#4,TYPE=T;DEG=K;MVIN2,3,4,5,6,7,8"
      LOCATE 2,2
      INPUT #2, EMF2,EMF3,EMF(4)
      FOR I=5 TO 8
      INPUT #2, EMF(I)
      EMF(I)=(EMF(I)+EMF(4))*1000.0
      CALL THRMCPL (EMF(I),TEMP(I,1))
      PRINT I,TEMP(I,1)
      NEXT I
RETRY:  PRINT "TYPE P FOR COMPLETE PURGE"
      PRINT "TYPE A FOR ADUSTMENT OF PRESSURE ONLY"
      INPUT "TYPE R FOR PRESSURE READING ONLY (NO ADDITION TO DATA FILE)", CONTROL$
      IF CONTROL$ = "A" THEN GOTO ADJUST
      IF CONTROL$ = "R" THEN GOTO READING
      IF NOT CONTROL$ = "P" THEN PRINT "ERROR,TRY AGAIN":GOTO RETRY
      INPUT "OPEN VALVE A, THEN HIT ENTER",DUMY
      FOR I=1 TO 2047
      VOUT%(I)=2047-I
      NEXT I
      CALL XDS(2047,VOUT%(1))
      FOR J=1 TO 5
      CALL SETTLE
      INPUT "CLOSE VALVE A, OPEN VALVE G, THEN HIT ENTER", DUMY
      FOR I=1 TO 4095
      VOUT%(I)=I
      NEXT I
      CALL XDS(4095,VOUT%(1))
      CALL SETTLE
      INPUT "CLOSE VALVE G, OPEN VALVE A, THEN HIT ENTER", DUMY
      FOR I=1 TO 4095
      VOUT%(I)=4096-I
      NEXT I
      CALL XDS(4095,VOUT%(1))
      NEXT J
      CALL SETTLE
      INPUT "CLOSE VALVE A, OPEN VALVE G, THEN HIT ENTER", DUMY
      FOR I=1 TO 2047
      VOUT%(I)=I
      NEXT I
    CALL XDS(2047,VOUT%(1))
      CALL SETTLE
ADJUST:
      '***************** FILL CYLINDER TO PROPER PRESSURE LEVEL ***************
      '
      CLS:PRINT TEMP(2,1);TEMP(3,1);TEMP(4,1);TEMP(5,1);TEMP(6,1);TEMP(7,1);TEMP(8,1)
TEMPB:PRINT #2, "#4,MVIN2,3,4,5,6,7,8"
      INPUT #2, EMF(2),EMF(3),EMF(4)
      FOR I=5 TO 8
      INPUT #2, EMF(I)
      EMF(I)= (EMF(I)+EMF(4))*1000.0
      CALL THRMCPL(EMF(I),TEMP(I,2))
```

```
      NEXT I
      PRINT EMF(2);EMF(3);TEMP(4,2);TEMP(5,2);TEMP(6,2);TEMP(7,2);TEMP(8,2)
      TEMPD=(TEMP(7,1)-TEMP(7,2))^2
      IF TEMPD >= 20.0 GOTO DONE
DEB:IF NOT EOF(2) THEN X$=INPUT$(1,#1):PRINT X$: GOTO DEB
      PRINT "PRESS F10 WHEN PRESSURE IS ADJUSTED"
      LOCATE 10,1:PRINT "PRESSURE:"
FINE:IF NOT EOF(2) THEN X$=INPUT$(1,#2):GOTO FINE
      II=II+1
      PRINT #2,"#4,MVIN1"
      INPUT #2, PVOLT
      CALL XSB(2)
      CALL XAV(0,1,VOL%(1))
      CALL XSB(1)
      PRES=(PVOLT-OFFSET)/CSENS + 14.7
      LOCATE 10,12
      PRINT PRES,II,PVOLT
      TTIME=TIMER+1
      GOTO FINE
FINISH:PRINT #1, DATE$,TIME$,PRES,TEMP(7,2),VOL%(1),EMF(2),EMF(3),EMF2,EMF3
      END
READING:CLS:PRINT "PRESS F9 WHEN PRESSURE IS ADJUSTED"
      LOCATE 10,1:PRINT "PRESSURE:"
DING:   IF NOT EOF(2) THEN X$=INPUT$(1,#2):GOTO DING
      II=II+1
      PRINT #2,"#4,MVIN1,2"
      INPUT #2, PVOLT,PVOLTA
      PRES=(PVOLT-OFFSET)/CSENS + 14.7
      LOCATE 10,12
      PRINT PRES,II,PVOLT,PVOLTA
      TTIME=TIMER+1
      GOTO DING
DONE:   END
      SUB SETTLE STATIC
      DIM CEC(30)
2640 'SUBROUTINE SETTLE
2660 'WAITS FOR CEC PRESSURE TO REACH EQUILIBRIUM
      TOL1=0.01
2680  PRINT "SUBROUTINE SETTLE"
2690 IF NOT EOF(2) THEN X$=INPUT$(1,#2):GOTO 2690
2700 PRINT #2,"@REPEAT 30,100;#4,MVIN1;@UNTIL"
2720 FOR I=1 TO 30
2740       INPUT #2,CEC(I)
2760      PRINT "CEC PRESSURE (UNCORRECTED mVOLTS):",CEC(I)
2780      IF I=1 THEN GOTO 2820
2800      IF ABS(CEC(I-1)-CEC(I)) < TOL1 THEN GOTO 2880
2820 NEXT I
2840 PRINT "SETTLE ERROR"
      END
2860 GOTO 3000
2880 PRINT #2,CHR$(20)
2900 TIME=TIMER+2
2920 NTIME=TIMER
2940 IF TIME>=NTIME THEN GOTO 2920
2960 IF NOT EOF(2) THEN X$=INPUT$(1,#2):GOTO 2960
3000 END SUB
REM $TITLE: 'THRMCPL.BAS  ** REQUIRES DATA FILE "THERM.DAT" **'
REM $SUBTITLE: 'SUBROUTINE TO CONVERT TYPE T THERMOCOUPLE EMFs TO DEGREES K'
REM $PAGESIZE: 66
SUB THRMCPL(EMF,T) STATIC
      IF A% = 1 GOTO SKIP
      OPEN "THERM.DAT" FOR RANDOM ACCESS READ AS #10 LEN=4
      A% = 1
      FIELD #10, 4 AS TEMP$
SKIP:   IF EMF > 2500 OR EMF< -6257 THEN
      T=0.0
      GOTO RET
      END IF
      N%=-EMF+2501
      GET #10,N%
      T=CVS(TEMP$)
RET:    END SUB
```

D.5 ACQDAT.BAS

```
REM $TITLE:'ACQDAT.BAS'
REM $PAGESIZE:66
'       ****************** PROGRAM ACQDAT.BAS **********************
'
'
      OPTION BASE 0
      DEFINT I-K
      CONST PI=3.14159265,SRO%=1024,SRI%=128,SHARP%=125.,TOL=8.
      CONST STROKE=3.8,OFFSET=0.0
      DIM SHARED CEC(30),CECCAL(40),DYTCAL%(40),COMAND%(SRO%),FILE$,_
      ACTUAL%(SRO%),DESIRE%(SRO%),C(SHARP%),X(10),Y(10),XA(10),YA(10),DIFF%(SRO%+2*SHARP%)
      DIM RCSYNCH(30),RDSYNCH%(30),TEMP(300,8),DSYNCH%(10),_
           PVDAT%(768),CSYNCH(10)
      DIM SHARED TOL1,DYTDFT,CECHYS,PMAX,PMIN,PERIOD,PDRIFT,ST1TIME,_
           ST2TIME,ST3TIME,ST4TIME,ENDTIME,A1,A2,LVSLP,LVYIN,FLAG%,_
           LVSLPA,LVYINA
'
      PERIOD = 32 :'INITIALIZE PERIOD TO 2 TIME ACTUAL 1ST PERIOD
'
      KEY(10) ON
      ON KEY(10) GOSUB TERMINATE
'
'
      OPEN "COM1:9600,N,8,2,CS,DS" AS #1
      OPEN "MASS.DAT" FOR INPUT AS #2
      OPEN "CAL.DAT"  FOR INPUT AS #3
      OPEN "LVDT.DAT" FOR INPUT AS #5
'
'     RESET ACRO
'
      CLS
      PRINT "RESETING ACRO-900"
40    IF NOT EOF(1) THEN X$=INPUT$(1,#1):GOTO 40
      PRINT #1, CHR$(24)
      TIME=TIMER+8
'
'     INPUT EXPERIMENT PARAMETERS
'
10      DAAT$=INPUT$ (12,#3)
      INPUT #3,A1,A2,STDERR
      IF NOT EOF(3) THEN GOTO 10
20      LINE INPUT #2, FULLLINE$           :'DAAT$=INPUT$ (28,#2)
      PRESS$=MID$(FULLLINE$,29,42)         :'INPUT #2, PRESS,TEMP,VOL%
      'THE ABOVE 2 LINES HAVE BEEN CHANGED TO ALLOW FOR VARIABLE LENGTH
      'MASS FILL DATA INFO (FOR GERMANIUM RESISTOR DATA)
      IF NOT EOF(2) THEN GOTO 20
'
'     SET TOLERENCES
'
      TOL1=0.10
      TOL2=8.
      CALL FILTER :' CALCULATE FILTER COEFFICIENTS.
      CALL POSITION
70      NTIME=TIMER
      IF TIME>=NTIME THEN GOTO 70
'
'     INITIALIZE TRIGGER & BRIDGE OUTPUTS
'
      PRINT #1,"#4,CJR1,2,3,5,6,7,8=0;CJR4=1;#1,VOUT1=10.0;VOUT2=-10.0"
'                                VOUT CHANNELS ARE USED TO RESET PIEZO AMP
      CALL XSB(2)
      CALL XRD(ID%):'                    RESET BOARD (NOT REALLY NECASSARY
      CALL XSB(1)
      CALL XDV(0,3072):'          SET DAC 0 TO 7.5 VOLTS (NO LONGER USED?)
      CALL XDV(1,2047):'         SET DAC 1 TO 5 VOLTS (MID VOLUME)
      BEEP
      INPUT "ENTER TWICE STARTING PERIOD (USUALLY 32)",PERIOD
      INPUT "OPEN HYDRAULIC VALVE, THEN HIT ENTER",DUMY
'
DO WHILE PERIOD >= 1.0
      PERIOD = PERIOD/2
'
'   WRITE EXPERIMENT PARAMETERS TO DATA FILE
      CALL FNDFILE
      PRINT #4,DATE$,TIME$,A1,A2,PRESS$          :',TEMP,VOL%  SEE CHANGE ABOVE
      PRINT #4,PERIOD,STROKE,OFFSET,LVSLP,LVYIN,LVSLPA,LVYINA
```

```
      FOR I=1 TO 10
           PRINT #4, X(I),Y(I),XA(I),YA(I)
      NEXT I
'
'     CALIBRATE PRESSURE SCALES
'
      PRINT "CALIBRATING PRESSURE SCALES"
      PRINT #1,"#1,VOUT2=5.0;VOUT2=-10.0"
      CALL CALPRS(SLOPE1,YINTRC1)
      PRINT #4,ST1TIME,ST2TIME,ST3TIME,ST4TIME,ENDTIME
'
'     CALCULATE CLOCK SPEEDS
'
      NTTS=(PERIOD/(2.*SRI%-1)/1.25E-6)
      NTTF%=(PERIOD/(SRO%-1)/1.25E-6)
      ITO = PERIOD*13.0
      COUNT%=0
      CALL XSB(1)
      CALL XST(ITO)
      CALL XSCD(NTTF%)
      CALL XSB(2)
      CALL XST(ITO)
      CALL XSCD(NTTF%)
'
'     CORRECT WAVEFORM
'
      PRINT "CORRECTING POSITION DATA"
      FOR I=1 TO SRO%
           COMAND%(I)=DESIRE%(I)
      NEXT I
'
'     OUTPUT COMMAND SIGNAL FOR 10 CYCLES
'
      CALL XSB(1)
      CALL XSD(0,1)
      CALL XCDD(SRO%,COMAND%(1))
      FOR I=1 TO 10
           CALL XWDD(COMAND%(SRO%))
      NEXT I
'
'
      CALL XSB(2)
      CALL XSA(0,0,0,1)
      CALL CORRECT
      CALL XSB(1)
      CALL XWDD(COMAND%(SRO%))
      CALL XSDD
'
'     SYNCHRONIZE PRESSURE SIGNALS
'
      CALL XSB(2)
      CALL XSA(0,1,1,1)
      PRINT "SYNCHRONIZING PRESSURE SIGNALS"
      CALL SETTLE
'
'     RESET PIEZO AMPLIFIER
'
      PRINT #1, "#1,VOUT2=5.0;VOUT2=-10.0"
'
'     COLLECT 5 SECONDS OF PRESSURE SIGNALS
'
      ST1SYNC=TIMER
      PRINT #1, "#4,@REPEAT 10,50;MVIN1;@UNTIL"
      FOR I=1 TO 10
           INPUT #1,CSYNCH(I)
           CALL XAV(1,1,DSYNCH%(I))
      NEXT I
      END1SYNC=TIMER
'
'     START DATA COLLECTION
'
      PRINT "COLLECTING PV DATA"
      CALL XSB(1)
      CALL XSD(0,1)
      CALL XSB(2)
      CALL XSA(0,0,1,1)
      IF NTTS > 32767 THEN
           NTTS%=NTTS-32767
           CALL XSSC(NTTS%)
      ELSE
           NTTS%=NTTS
```

```
          CALL XSCD(NTTS%)
        END IF
'
'       BEGIN BACKGROUND CONVERSION OF THERMOCOUPLES
'
        PRINT #1, "#4,TYPE=T;DEG=K;BUF2,3,4,5,6,7,8=DIM37:ON;CONV2,3,4,5,6,7,8=65535"
'
'       BEGIN CONTINUOUS OUTPUT OF COMMAND SIGNAL
'
        SSRI%=6*SRI%
        FSRI%=4*SRI%
        CALL XSB(1)
        PRINT "OUTPUTING CORRECTED SIGNAL NOW"
        CALL XCDD(SRO%,COMAND%(1))
'
'       BEGIN MONITORING PRESSURE AND POSITION
'
        CALL XSB(2)
        CALL XCAD(SSRI%,PVDAT%(1))
'
'       WAIT FOR CONVERGENCE OF CYCLE
'
        PRINT "TOLERENCE =",TOL2
        CALL XWAD(PVDAT%(SSRI%))
        FOR ICOUNT=1 TO 4
             CALL XWAD(PVDAT%(FSRI%))
             CALL XWAD(PVDAT%(SSRI%))
             PVDIFF=ABS(PVDAT%(FSRI%)-PVDAT%(SSRI%))
             PRINT "PVDIFF=",PVDIFF
             IF PVDIFF < TOL2 THEN GOTO 820
        NEXT ICOUN
        PRINT "PRESSURE HAS NOT CONVERGED AFTER 12 CYCLES, ADJUST TOL ?"
'         STOP
          '
820       'COLLECT THREE FULL CYCLES OF PV DATA
          '
        STDATTIME=TIMER
        CALL XWAD(PVDAT%(SSRI%))
        CALL XSAD
        CALL XSB(1)
        CALL XWDD(COMAND%(SRO%))
        CALL XSDD
        PRINT #1, "#4,CONV2,3,4,5,6,7,8=0;BUF2,3,4,5,6,7,8=OFF;BUF2,3,4,5,6,7,8=VIN"
        ENDDATTIME=TIMER
'
'       RESYNCHRONIZE PRESSURE SIGNALS
'
        CALL XSB(2)
        PRINT "RESYNCHRONIZING PRESSURE SIGNALS"
        CALL SETTLE
        ST2SYNC=TIMER
        PRINT #1,"#4,@REPEAT 30,50;MVIN1;@UNTIL"
        FOR I=1 TO 30
             INPUT #1,RCSYNCH(I)
             CALL XAV(1,1,RDSYNCH%(I))
        NEXT I
        END2SYNC=TIMER
'
'       RECALIBRATE PRESSURE SCALES
'
        CALL CALPRS (SLOPE2,YINTRC2)
'
'       RECORD SYNCH. AND PV DATA
'
        PRINT #4,ST1TIME,ST2TIME,ST3TIME,ST4TIME,ENDTIME,STDATTIME,ENDDATTIME,_
        ST1SYNC,END1SYNC,ST2SYNC,END2SYNC
        FOR I=1 TO 10
             PRINT #4,CSYNCH(I),DSYNCH%(I)
        NEXT I
        FOR I=1 TO 30
             PRINT #4,RCSYNCH(I),RDSYNCH%(I)
        NEXT I
        FOR I=1 TO 768 STEP 2
             PRINT #4, PVDAT%(I),PVDAT%(I+1)
        NEXT I
'
'       UNLOAD TEMPERATURE DATA FROM ACRO-900
'
        PRINT #1,"#4,BUF8:N"
        INPUT #1,INUM
        FOR N%=1 TO INUM
```

```
            PRINT #1,"#4,BUF2,3,4,5,6,7,8"
            FOR I=2 TO 8
                  INPUT #1, TEMP(N%,I)
                  IF I > 4 THEN
                        TEMP=(TEMP(N%,I)+TEMP(N%,4))*10.0^6
                        CALL THRMCPL(TEMP,TEMP(N%,I))
                  END IF
            NEXT I
      NEXT N%
'
'     RECORD TEMPERATURE DATA
'
      FOR I=1 TO INUM
            PRINT #4,TEMP(I,2),TEMP(I,3),TEMP(I,4),TEMP(I,5),TEMP(I,6),TEMP(I,7),TEMP(I,8)
      NEXT I
      close #4
LOOP
TERMINATE: CALL XSB(1)
      CALL XSDD
      CALL XSB(2)
      CALL XSAD
'
      BEEP
      END
'
'
'ERROR TRAP, IF NON-EXISTANT FILE CAUSED ERROR THEN OPEN THE FILE
'   IF ERROR IS DIFFERENT THEN END
'
'
TROUBLE:IF ERR=53 AND ERL= 3 THEN
            CLOSE #4
            PRINT "OPENING",FILE$
            OPEN FILE$ FOR OUTPUT AS #4
            FLAG%=1
            RESUME NEXT
      ELSE
           PRINT "ERROR OCCURED: ERROR CODE:",ERR,"NEAREST LINE NO:",ERL
      END
      END IF
'
'$SUBTITLE:'SUBROUTINE CALPRS; CALIBRATE PRESSURE TRANSDUCERS'
'$PAGE
'
      SUB CALPRS(SLOPE,YINTRC) STATIC
1000 ' SUBROUTINE CALPRS
1020 ' MATCHES CALIBRATION OF DYTRAN PEIZO TRANS. TO CEC TRANS.
1040 '
1060 'SET PISTON TO MID STROKE & SETUP DAC PARAMETERS
1080 DEFINT I-K
1100 DIM IVOUT(410)
      IF SLOPE = 3.5 THEN SLOPE%=1
1120 CALL XSB(1)
1140 CALL XDV(1,2047)
1160 CALL XSD(0,1)
1180 ITIME=16383
1200 CALL XSCD(ITIME)
1220 '
1240 ' WAIT FOR CEC TRANS. TO SETTLE OUT
1260 CALL SETTLE
1300 J=1
      ST1TIME=TIMER
1320 GOSUB 2440
1340 'RAMP TO MAX VOLUME
1360 FOR I=1 TO 204
1380    IVOUT(I)=2046+I*10
1400 NEXT I
1420 CALL XSB(1)
1440 CALL XDS(204,IVOUT(1))
1460 'WAIT FOR CEC TO SETTLE OUT
1480 CALL SETTLE
1500 J=2
      ST2TIME=TIMER
1520 GOSUB 2440
1540 'RAMP TO MIN VOLUME
1560 FOR I=1 TO 409
1580     IVOUT(I)=4096-I*10
1600 NEXT I
1620 CALL XSB(1)
1640 CALL XDS(409,IVOUT(1))
      CALL SETTLE
```

```
1660 J=3
      ST3TIME=TIMER
1680 GOSUB 2440
1700 FOR I=1 TO 205
1720      IVOUT(I)=I*10
1740 NEXT I
1760 CALL XSB(1)
1780 CALL XDS(205,IVOUT(1))
1800 CALL XDV(1,2047)
1820 CALL SETTLE
1840 J=4
      ST4TIME=TIMER
1860 GOSUB 2440
      ENDTIME=TIMER
1920 PMAX=0
1940 PMIN=0
2040 FOR I=1 TO 10
2080      PMAX =((PMAX+CECCAL(10+I))/10-A2)*A1
2100      PMIN =((PMIN+CECCAL(20+I))/10-A2)*A1
2220 NEXT I
      IF SLOPE% = 1 THEN GOTO 2615
2360 FOR I=1 TO 40
2380      PRINT #4, CECCAL(I),DYTCAL%(I)
2400 NEXT I
2420 GOTO 2620
2440 'ROUTINE TO COLLECT 5 SEC OF PRESS. DATA
2460 PRINT #1,"#4,@REPEAT 10,50;MVIN1;@UNTIL"
2480 CALL XSB(2)
2500 K=(J-1)*10
2520 FOR I=1 TO 10
2540      INPUT #1, CECCAL(K+I)
      CECPRES=(CECCAL(K+I)-A2)/A1+14.7
2560      PRINT "PRESSURE IN CYLINDER =",CECPRES
2580      CALL XAV(1,1,DYTCAL%(K+I))
2600 NEXT I
2610 RETURN
2615 SLOPE%=0
2620 END SUB
'
'
'$SUBTITLE:'SUBROUTINE SETTLE; WAIT FOR PRESSURE TO SETTLE'
'$PAGE
'
      SUB SETTLE STATIC
2640 'SUBROUTINE SETTLE
2660 'WAITS FOR CEC PRESSURE TO REACH EQUILIBRIUM
2680  PRINT "SUBROUTINE SETTLE"
2690 IF NOT EOF(1) THEN X$=INPUT$(1,#1):GOTO 2690
2700 PRINT #1,"@REPEAT 30,100;#4,MVIN1;@UNTIL"
2720 FOR I=1 TO 30
2740       INPUT #1,CEC(I)
2760      PRINT "CEC PRESSURE (UNCORRECTED mVOLTS):",CEC(I)
2780      IF I=1 THEN GOTO 2820
2800      IF ABS(CEC(I-1)-CEC(I)) < TOL1 THEN GOTO 2880
2820 NEXT I
2840 PRINT "SETTLE ERROR"
2880 PRINT #1,CHR$(20)
2900 TIME=TIMER+2
2920 NTIME=TIMER
2940 IF TIME>=NTIME THEN GOTO 2920
2960 IF NOT EOF(1) THEN X$=INPUT$(1,#1):GOTO 2960
3000 END SUB
'
'$SUBTITLE:'SUBROUTINE FILTER; CALCULATES THE FILTER COEFFICIENTS'
'$PAGE
'
'
      SUB FILTER STATIC
'       ************** SUBROUTINE FILTER *********************
'       CALCULATES THE FILTER COEFICIENTS
'
      FS=1./70.
      C(0)=2*FS
      FOR K=1 TO SHARP%
           C(K)=SIN(PI*K/SHARP%)*SIN(2*PI*K*FS)/((PI^2)*(K^2)/SHARP%)
      NEXT K
      END SUB
'$SUBTITLE:'SUBROUTINT CORRECT; CORRECT VOLUME WAVEFORM'
'$PAGE
'
'
```

```
      SUB CORRECT STATIC
'
'
'       ***************** SUBROUTINE TO CORRECT VOLUME WAVEFORM *************
'
      CALL XSB(1)
      HSRO%=SRO%/2
      COUNT%=0
BEGIN:  COUNT%=COUNT%+1
      CALL XWDD(COMAND%(1))
      CALL XSB(2)
      CALL XBAD(SRO%,ACTUAL%(1))
      CALL XWAD(ACTUAL%(SRO%))
      FOR I=1 TO SRO%
           DIFF%(I+SHARP%)=DESIRE%(I)-ACTUAL%(I)
      NEXT I
      FOR I=1 TO SHARP%
           DIFF%(I)=DESIRE%(SRO%-SHARP%+I)-ACTUAL%(SRO%-SHARP%+I)
           DIFF%(I+SRO%+SHARP%)=DESIRE%(I)-ACTUAL%(I)
      NEXT I
      FLAG%=0
      FOR I=1 TO SRO%
           SMOOTH=0
           FOR K=-SHARP% TO SHARP%
                 J=ABS(K)
                 SMOOTH=SMOOTH+C(J)*DIFF%(I+SHARP%+K)
           NEXT K
           IF ABS(SMOOTH) > TOL THEN FLAG%=1
           ACTUAL%(I)=CINT(SMOOTH)
      NEXT I
      CALL XSB(1)
      CALL XWDD(COMAND%(HSRO%))
      IF FLAG%=0 THEN GOTO FINISH
      FOR I=1 TO HSRO%
           COMAND%(I)=COMAND%(I)+ACTUAL%(I)
           IF COMAND%(I) >4095 THEN
                 COMAND%(I)=4095
           ELSEIF COMAND%(I) < 0 THEN
                 COMAND%(I)=0
           END IF
      NEXT I
      CALL XWDD (COMAND%(SRO%))
      FOR I=HSRO%+1 TO SRO%
           COMAND%(I)=COMAND%(I)+ACTUAL%(I)
           IF COMAND%(I) >4095 THEN
                 COMAND%(I)=4095
           ELSEIF COMAND%(I) < 0 THEN
                 COMAND%(I)=0
           END IF
      NEXT I
      PRINT "NUMBER OF TIMES THROUGH FILTER:",COUNT%
      IF COUNT%=3 GOTO FINISH
      FOR I=1 TO 3
           CALL XWDD(COMAND%(SRO%))
      NEXT I
      GOTO BEGIN
FINISH: PRINT "WAVEFORM HAS BEEN CORRECTED, LATEST DIFF=",ACTUAL%(I-1)
      END SUB

'$SUBTITLE:' SUBROUTINE POSITION; CALCULATE DESIRED VOL WAVE'
'$PAGE
      SUB POSITION STATIC
      INPUT #5, LVSLP,LVYIN,LVSLPA,LVYINA
      FOR I=1 TO 10
            INPUT #5,X(I),Y(I),XA(I),YA(I)
            SUMX=SUMX+X(I)*2.0/33.0
      NEXT I
      RIOFFSET=(-LVYIN-SUMX)/LVSLP
'     INPUT " ENTER PERIOD OF A SINGLE CYCLE (SEC):",PERIOD
'80      INPUT " ENTER TOTAL STROKE IN INCHES (<4.0): ",STROKE
'     INPUT " ENTER STROKE OFFSET IN INCHES:       ",OFFSET
'     IF STROKE > 4! THEN GOTO 80
'     IF PERIOD = 0! THEN PERIOD =1!
''    IF STROKE = 0! THEN STROKE=2!
'     EXTLEN=STROKE\2 + OFFSET
'IF EXTLEN > 4! GOTO 90
'GOTO 100
'90      PRINT " OFFSET + STROKE/2 MUST BE LESS THE 4.0 in."
'GOTO 80
```

```
100     PRINT " CALCULATING POSITION DATA"
FOR I=1 TO SRO%
RINCHES=STROKE*(3.0/3.8*SIN(2*PI*I/SRO%)+1.2/STROKE*SIN(4*PI*I/SRO%)+1)/2+OFFSET+RIOFFSET
DESIRE%(I)=(RINCHES*LVSLP)+LVYIN
          FOR J=1 TO 10
                DESIRE%(I)= DESIRE%(I) +
CINT(X(J)*2.0/33.0*COS(J*PI*(I-1)/SRO%)+Y(J)*2.0/33.0*SIN(J*PI*(I-1)/SRO%))
          NEXT J
      NEXT I
      END SUB
```

```
REM $TITLE: 'THRMCPL.BAS  ** REQUIRES DATA FILE "THERM.DAT" **'
REM $SUBTITLE: 'SUBROUTINE TO CONVERT TYPE T THERMOCOUPLE EMFs TO DEGREES K'
REM $PAGE
'
'
'
SUB THRMCPL(EMF,T) STATIC
      IF A% = 1 GOTO SKIP
      OPEN "THERM.DAT" FOR RANDOM ACCESS READ AS #10 LEN=4
      A% = 1
      FIELD #10, 4 AS TEMP$
SKIP:   IF EMF > 2500 OR EMF< -6257 THEN
          T=0.0
          GOTO RET
      END IF
      N%=-EMF+2501
      GET #10,N%
      T=CVS(TEMP$)
RET: END SUB
'$SUBTITLE:'SUBROUTINE FNDFILE; FIND AN UNUSED FILE NAME'
'$PAGE
SUB FNDFILE STATIC
'INITIALIZATION: SHOULD ONLY BE DONE ONCE
'
'
ON ERROR GOTO TROUBLE
STATIC NO$,NAME$ :' THESE VARIABLE WILL RETAIN THEIR VALUES AT RE-ENTRY, BUT CANNOT BE
SEEN ELSEWHERE
'
'FILE$ DECLARED GLOBAL BECAUSE ERROR TRAP MUST BE AT MAIN LEVEL
PRINT NAME$
IF NO$<CHR$(65) THEN
      NO$=CHR$(65)
      NAME$ = "PV"+LEFT$(DATE$,2)+MID$(DATE$,4,2)
ELSE
      NO$=CHR$(ASC(NO$)+1)
END IF
'
'CHECK TO SEE IF FILE EXISTS, NOTE - IF IT DOES NOT EXIST ERR-53 OCCURS
'   THE ONLY WAY TO EXIT THIS LOOP IS TO FIND A FILE THAT DOES NOT EXIST
'   AND THEN THE ERROR TRAP WILL CAUSE AN EXIT
'
'
FLAG%=0
DO WHILE 1
      FILE$=NAME$+NO$+".DAT"
3       OPEN FILE$ FOR INPUT AS #4
      IF FLAG%=1 THEN EXIT DO
      CLOSE #4
      NO$=CHR$(ASC(NO$)+1)
LOOP
'
'
ON ERROR GOTO 0
END SUB
```

D.6 DRIFT1.FOR

```
$TITLE: 'DRIFT1.FOR'
$SUBTITLE:'ROOM TEMPERATURE REDUCTION PROGRAM'
**************PROGRAM DRIFT**********************
$NOTRUNCATE
$INCLUDE: 'HGRAPHALIAS.FOR'
      IMPLICIT REAL (A-H,L-Z)
C
      REAL A1,A2,PRESS,TEMP,PERIOD,STROKE,OFFSET,LVSP,
     1 LVYIN,VMAG(0:6),INCHES,PPLOTDAT(384),AVGG(256)
C
      CHARACTER DATE*14,CTIME*14,RESPONSE*1,VS1*10,VS2*10,
     1 TS1*10,TS2*10,TS3*10,TS4*10,CREYNOLD*14,JUNK*14
C
      CHARACTER FNAME*30,CMASS*11,CVR*7,CPER*9,TITLE*90
C
      DIMENSION CECCAL1(40),CECCAL2(40),DYTCAL1(40),DYTCAL2(40),
     1 CECSYNC1(80),CECSYNC2(30),DYTSYNC1(80),DYTSYNC2(30),
     2 CAL1TIME(40),CAL2TIME(40),SYNC1TIME(10),SYNC2TIME(30),YCALC(400)
C
      DIMENSION PCALC(40),PCALC2(40),PCALC3(40),CAL3TIME(80),
     1 VOLDAT(385),PRESDAT(384),TIME(384),PMAG(0:6),
     2 PPHASE(6),X(10),Y(10)
C
      EQUIVALENCE (DYTSYNC1(11),DYTSYNC2(1)),(CECSYNC1(11),CECSYNC2(1))
     1 ,(DYTSYNC1(41),DYTCAL2(1)),(CECSYNC1(41),CECCAL2(1))
C
      PARAMETER (PI=3.14159265)
C     OPEN (UNIT=4,FILE='BATCH.DAT',STATUS='OLD')
C
C     ***************** OPEN FILE TO BE PROCESSED ******************
 4011      WRITE(*,*) 'CURRENT FILE:',FNAME
      WRITE(*,'(A\)') ' ENTER FILE NAME:'
                           READ (*,'(A30)') FNAME
                           OPEN (UNIT=1,FILE=FNAME,STATUS='OLD')
                           REWIND 1
C
C
C      ************* READ IN PARAMETERS OF DATA RUN *****************
                           READ (1,1000)DATE,CTIME,A1,A2,PRESS,TEMP,VOLUME
                           READ (1,'(7F14.0)') PERIOD,STROKE,OFFSET,LVSP,LVYIN,LVSP,LVYIN
                           READ (1,'(4F14.0)')(X(I),Y(I),X(I),Y(I), I=1,10)
C
C      READ IN FIRST PRESSURE CALIBRATION DATA SET
      DO 10, I=1,40
           READ (1,1010)CECCAL1(I),DYTCAL1(I)
   10  CONTINUE
      READ (1,1020)ST1TIME1,ST2TIME1,ST3TIME1,ST4TIME1,ENDTIME1
C
C      READ IN SECOND PRESSURE CALIBRATION DATA SET
      DO 20, I=1,40
           READ (1,1010) CECCAL2(I),DYTCAL2(I)
   20  CONTINUE
      READ (1,1030)ST1TIME2,ST2TIME2,ST3TIME2,ST4TIME2,STDATTIME,
     1 ENDDATTIME,ST1SYNC,END1SYNC,ST2SYNC,END2SYNC
 1030  FORMAT(10F14.0)
C
C     READ IN FIRST PRESSURE SYNCHRONIZATION DATA SET
      DO 30, I=1,10
           READ (1,1010)CECSYNC1(I),DYTSYNC1(I)
   30  CONTINUE
C
C      READ IN SECOND PRESSURE SYNCHRONIZATION DATA SET
      DO 40, I=1,30
           READ (1,1010)CECSYNC2(I),DYTSYNC2(I)
   40  CONTINUE
C
C      READ IN P-V DATA
      DO 50, I=1,384
           READ (1,1010) VOLDAT(I),PRESDAT(I)
C     WRITE(*,*) VOLDAT(I),PRESDAT(I)
   50  CONTINUE
C      ******************************************************************
C
C
      AVGCSYNC1=0
      AVGCSYNC2=0
```

```
C
C      ****************** CREATE TIME DATA ARRAYS *********************
       DO 100, I=1,10
            CAL1TIME(I)    =(I-1)*0.5
            CAL1TIME(I+10)=(I+9)*0.5
            CAL1TIME(I+20)=(I+19)*0.5
            CAL1TIME(I+30)=(I+29)*0.5
C
            CAL2TIME(I)    =(I-1)*0.5
            CAL2TIME(I+10)=(I-1)*0.5 + ST2TIME-ST1TIME
            CAL2TIME(I+20)=(I-1)*0.5 + ST3TIME-ST1TIME
            CAL2TIME(I+30)=(I-1)*0.5 + ST4TIME-ST1TIME
C
            CAL3TIME(I)    =(I-1)*0.5
            CAL3TIME(I+40)=(I-1)*0.5 + ST1TIME2-ST1SYNC
            CAL3TIME(I+50)=(I-1)*0.5 + ST2TIME2-ST1SYNC
            CAL3TIME(I+60)=(I-1)*0.5 + ST3TIME2-ST1SYNC
            CAL3TIME(I+70)=(I-1)*0.5 + ST4TIME2-ST1SYNC
C
       SYNC1TIME(I)=(I-1)*0.5
C       CONVERT RAW CEC SYNCH. OUTPUT TO PRESSURES AND AVERAGE *****
       CECSYNC1(I) =(CECSYNC1(I)-A2)/A1+14.7
       AVGSYNC1=AVGSYNC1+CECSYNC1(I)
C
  100   CONTINUE
       AVGSYNC1=AVGSYNC1/10
       DO 110, I=1,30
            SYNC2TIME(I)=(I-1)*0.5
            CAL3TIME(I+10)=(I-1)*0.5 + ST2SYNC-ST1SYNC
C       CONVERT RAW CEC SYNCH. OUTPUT TO PRESSURES AND AVERAGE *****
            CECSYNC2(I) =(CECSYNC2(I)-A2)/A1+14.7
            AVGSYNC2=AVGSYNC2+CECSYNC2(I)
  110   CONTINUE
       AVGSYNC2=AVGSYNC2/30
C
C
C       CONVERT RAW CEC CALIBRATION OUTPUT TO PRESSURES **************
       DO 120, I=1,40
            CECCAL1(I)=(CECCAL1(I)-A2)/A1+14.7
            CECCAL2(I)=(CECCAL2(I)-A2)/A1+14.7
  120   CONTINUE
C
C       FIND BEST FIT FOR DYTRAN OUTPUT DATA TO CEC OUTPUT DATA
C       ********* FIRST PRESSURE CALIBRATION *************
C       USING LEAST SQUARES FIT IN BOTH PRESSURE AND TIME TO COMPENSATE
C       FOR TIME DRIFT.
C
       CALL LSSQ3D(CECCAL1,DYTCAL1,CAL2TIME,40,A10,A11,A12)
C       STORE PRESSURE AND TIME SENSITIVITIES AS STRINGS FOR GRAPHICS
       WRITE (VS1,5000)A11
       WRITE (TS1,5000)A12
C       USE CURVE FIT PARAMS TO CONVERT DYTRAN OUTPUT TO PRESSURE
       DO 121, I=1,40
            PCALC(I)=A10+A11*DYTCAL1(I)+A12*CAL2TIME(I)
  121   CONTINUE
C
C
C
C      ********************
C
C
C
C      LET A3 EQUAL THE VOLTAGE-PRESSURE SENSITIVITY OF THE FIRST CAL.
       A3=A11
C
C
C      LET OFFSET A4 EQUAL THE MEAN OF THE TWO SYNCH. RUNS
       A4=(CECSYNC1(10)+CECSYNC1(11))/2-(DYTSYNC1(10)+DYTSYNC1(11))/2*A3
C
C
C**********************************************************************
C
C
C
C
C      ***** CALCULATE PRESSURE TIME SLOPES FOR CECSYNC RUNS **********
       CALL LSSQLN(SYNC2TIME,CECSYNC2,30,CTSLP2,YINTRC,YCALC,STDERR)
       CALL LSSQLN(SYNC1TIME,CECSYNC1,10,CTSLP1,YINTRC,YCALC,STDERR)
C
C      **** CALCULATE PRESSURE TIME SLOPES FOR DYTSYNC RUNS USING
C           PRESSURE SLOPES FROM FIRST CALIBRATION ******************
```

```
      DO 123, I=1,40
           DYTSYNC1(I)=DYTSYNC1(I)*A1
  123      CONTINUE
      CALL LSSQLN(SYNC2TIME,DYTSYNC2,30,DTSLP2,YINTRC,YCALC,STDERR)
      CALL LSSQLN(SYNC1TIME,DYTSYNC1,10,DTSLP1,YINTRC,YCALC,STDERR)
C*********************************************************************
C
C
C
C
C
CC
C
C
C*********************************************************************
C     CORRECT 2ND P CALIBRATION FOR TIME DRIFT BASED ON 2ND SYNCH. RUN
C     DO 200, I=1,40
C          DYTCAL2(I)=DYTCAL2(I)-TSLOPE3 * CAL3TIME(I+40)
C  200  CONTINUE
C     CORRECT 2ND P CALIBRATION FOR TIME DRIFT BASED ON 2ND SYNCH. RUN
C*********************************************************************
C
C
C
C
C*********************************************************************
C     ************** SECOND PRESSURE CALIBRATION ****************
C     FIND BEST FIT FOR DYTRAN OUTPUT DATA TO CEC OUTPUT DATA
C     USING LEAST SQUARES FIT IN BOTH PRESSURE AND TIME
      CALL LSSQ3D(CECCAL2,DYTCAL2,CAL2TIME,40,A0,A21,A22)
C*********************************************************************
C
C
C
C     RECORD VARIOUS SENSITIVITIES AS STRINGS FOR GRAPHICS
      TSLP2=CTSLP1-DTSLP1
      TSLP4=CTSLP2-DTSLP2
      WRITE (VS2,5000)A21
      WRITE (TS2,5000)TSLP2
      WRITE (TS3,5000)A22
      WRITE (TS4,5000)TSLP4
C
C     CONVERT SECOND DYTRAN CALIBRATION AND BOTH SYNCH. RUNS
C     TO PRESSURE USING FIRST CAL PRESS VS. VOLT SENS. AND
C     ALSO ORIGINAL OFFSET.
      DO 122, I=1,40
           PCALC2(I)=A0+A21*DYTCAL2(I)
           PCALC3(I)=A0+A21*DYTSYNC1(I)+A2*CAL3TIME(I)
  122  CONTINUE
      CAL1MAX=(AINT(CAL1TIME(40)/5)+1.0)*5
C
C
C
C      *************** CALCULATE PERTINANT PV DATA ******************
C     ***************************************************************
C     NOTE: PRESDAT IS CONVERTED USING THE FOLLOWING:
C          PRESS VS. VOLT. SENS. FROM FIRST CAL. RUN  (A3)
C          OFFSET AS AVERAGE OF THE LAST OF THE FIRST SYNCH. AND
C          THE FIRST OF THE SECOND SYNCH.             (A4)
C
C
      VMAX=0.0
      VMIN=100
      PREF = -100
      CLVOL=1.12
      DO 210, I=1,384
           TIME(I)=(I-1)*PERIOD/128
           INCHES=0.0
           DO 211, J=1,10
                INCHES=INCHES +
     1                X(J)*2.0/33.0*COS(J*PI*VOLDAT(I)/4096.0)+
     2                Y(J)*2.0/33.0*SIN(J*PI*VOLDAT(I)/4096.0)
  211      CONTINUE
           INCHES=INCHES+VOLDAT(I)*LVSP+LVYIN
           VOLDAT(I)=clvol+(PI*(1.469**2))/4*INCHES
           VMAX=AMAX1(VOLDAT(I),VMAX)
           VMIN=AMIN1(VOLDAT(I),VMIN)
           PRESDAT(I)=A4+A3*PRESDAT(I)
C          PREF = AMAX1(PRESDAT(I),PREF)
  210  CONTINUE
C     ***** TIME DRIFT IS COMPENSATED FOR BY RUNNING AVERAGE ******
```

```
C
C
C
      CALL AVERAGE(PRESDAT(1),AVGG(1))
C
C
C     ***************** THE ABOVE ROUTINE ALTERS PRESDAT ************
C
      VOLDAT(385)=0.0
      VREF = (VMIN+VMAX)/2.
      VOLRAT = VMAX/VMIN
C
C
      VOLD=VOLDAT(129)
      DO 220, I=130,256
           VNEW=VOLDAT(I)
           IF (VNEW.GT.VOLDAT(I-5).AND.VNEW.LT.VOLDAT(I+5)) THEN
           IF (VNEW.GT.VOLD.AND.VNEW.GT.VREF.AND.VOLD.LT.VREF) THEN
                GOTO 221
           END IF
           END IF
           VOLD=VNEW
  220      CONTINUE
      WRITE (*,*) 'ERROR FINDING VREF'
      STOP
  221 CONTINUE
      PREF= 0
      DO 222, J=I-10,I+10
           PREF=PREF+PRESDAT(J)
  222      CONTINUE
      PREF=PREF/21.
C
C     FIND DRIFT IN P-V DATA
C
      CALL SCALES (VOLDAT(1),385,VMIN,VMAX,VTICS)
      CALL SCALES (PRESDAT(1),384,PMIN,PMAX,PTICS)
      CALL SCALES (TIME(1),384,TMIN,TMAX,TTICS)
      MASS= (PRESS*(CLVOL+(PI*(1.469**2))/4*
     1     (VOLUME*LVSP+LVYIN)))/(18.381*TEMP)
      PER3=3.0*PERIOD
      REYNOLD=(2*MASS/(VMAX+VMIN))*(2*STROKE/PERIOD)*1.460/0.5177E-3
      PEC=REYNOLD*0.69
C
C*********************************************
C     FOR ROOM TEMP RUNS
C     K(HE) = 0.155
C     CP(HE) = 5190
C     RRCKS  =7040
C     FOR LN2 TEMP RUNS
C     K(HE) = 0.061
C     CP(HE)= 5225.
C     RRCKS = 3150
C*********************************************
C
      RHOREF= MASS/(VREF*(0.0254**3)*1000)
      RRCKH =(0.155*5193*RHOREF)**0.5
      RRCKS = 7070.
      WRITE (CREYNOLD,'(F14.2)')REYNOLD
      WRITE(CMASS,'(F10.8)')MASS
      WRITE(CVR,'(F6.3)')VOLRAT
      WRITE(CPER,'(F9.2)')PERIOD
C
C
C      ********** DECOMPOSE PRESSURE WAVE **************
C
     CALL DECOMP(VOLDAT(1),PRESDAT(1),PREF,
    1 VREF,PMAG(1),PPHASE(1),VMAG(1),PERIOD,
    2 PPLOTDAT(1))
C
C
C
C*********************************************************************
C
C
      CALL SCALES(PPHASE(1),3,PPHASEMIN,PPHASEMAX,PPHASETICS)
      CALL SCALES(PMAG(1),4,PMAGMIN,PMAGMAX,PMAGTICS)
      CALL SCALES(VMAG(1),6,VMAGMIN,VMAGMAX,VMAGTICS)
      CALL SCALES(CECCAL1(1),40,CEC1MIN,CEC1MAX,YTICS1)
      CALL SCALES(DYTCAL1(1),40,DYT1MIN,DYT1MAX,YTICS2)
      CALL SCALES(CECCAL2(1),40,CEC2MIN,CEC2MAX,YTICS3)
      CALL SCALES(DYTCAL2(1),40,DYT2MIN,DYT2MAX,YTICS4)
```

```
      CALL SCALES(CECSYNC1(1),40,CSYNCMIN,CSYNCMAX,CSYNTICS)
C     CALL SCALES(PCALC3(1),40,CSYNCMIN,CSYNCMAX,CSYNTICS)
      CALL SCALES(DYTSYNC1(1),40,DSYNCMIN,DSYNCMAX,DSYNTICS)
C     CALL SCALES(YCALC(1),35,CORRMIN,CORRMAX,CORRTICS)
C*********************************************************************
C
C
C
 4000     CONTINUE
      WRITE (*,'(A)') '1'
      WRITE (*,'(A)') '1'
      WRITE (*,'(A)') '1'
      WRITE (*,'(A)') '1'
      WRITE (*,*) ' CURRENT FILE:',FNAME
      WRITE (*,'(A)') '1'
      WRITE (*,*) ' ************************************************'
      WRITE (*,*) ' 1.  DISPLAY PRESSURE TRANSDUCER SENSITIVITIES'
      WRITE (*,*) ' 2.  WRITE TO CORE.DAT'
      WRITE (*,*) ' 3.  DISPLAY PLOT P-V'
      WRITE (*,*) ' 4.  DISPLAY PLOT PRESSURE MAGNITUDE SPECTRUM'
      WRITE (*,*) ' 5.  DISPLAY PLOT PRESSURE PHASE SPECTRUM'
      WRITE (*,*) ' 6.  DISPLAY PLOT PRESSURE CALIBRATION #1'
      WRITE (*,*) ' 7.  DISPLAY PLOT PRESSURE CALIBRATION #2'
      WRITE (*,*) ' 8.  DISPLAY PLOT PRESSURE SYNCHRONIZATION'
      WRITE (*,*) ' 9.  DISPLAY PLOT PRESSURE WAVES'
      WRITE (*,*) ' 10. DISPLAY PLOT PRESSURE TRANSDUCER DRIFT'
      WRITE (*,*) ' 11. PROCESS NEW FILE'
      WRITE (*,*) ' 12. EXIT PROGRAM'
      WRITE (*,*) ' ************************************************'
      WRITE (*,'(A)') '1'
      WRITE (*,'(A\)')' ENTER NUMBER OF DESIRED ACTION FROM ABOVE LIST:'
READ (*,'(I2)') IAC
GOTO (4001,4002,4003,4004,4005,4006,4007,4008,4009,4010,4011,4012)
     1     IAC
CGOTO 4002
C
C
C
C*********************************************************************
C
C
CPRINT OUT VARIOUS PRESSURE TRANSDUCER SENSITIVITIES
C
 4001     CONTINUE
      WRITE (*,*) ' DATA FILE:',FNAME
      WRITE (*,*) ' ',DATE,'    ',CTIME
      WRITE (*,*) ' MASS (GRAMS):.............................',CMASS
      WRITE (*,*) ' VOLUME RATIO:.............................',CVR
      WRITE (*,*) ' PERIOD (SECS):............................',CPER
      WRITE (*,*) ' TEMPERATURE RANGE:........................ LN2'
      WRITE (*,*) ' REYNOLDS NO.:.............................',
     1     CREYNOLD
      WRITE (*,*) ' PECLET NO.:...............................',PEC
      WRITE (*,*) ' FIRST HARMONIC PRESSURE MAGNITUDE:........',
     1     PMAG(1)
      WRITE (*,*) ' SECOND HARMONIC PRESSURE MAGNITUDE:.......',
     1     PMAG(2)
      WRITE (*,*) ' FIRST HARMONIC PRESSURE PHASE:............',
     1     PPHASE(1)
      WRITE (*,*) ' SECOND HARMONIC PRESSURE PHASE:...........',
     1     PPHASE(2)
      WRITE (*,*) ' *********** FIRST PRESSURE CALIBRATION *******'
      WRITE (*,*) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A11
      WRITE (*,*) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A12
      WRITE (*,*) ' *********** SECOND PRESSURE CALIBRATION ******'
      WRITE (*,*) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A21
      WRITE (*,*) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A22
      WRITE (*,*) ' *********** FIRST SYNCHRONIZATION ************'
      WRITE (*,*) ' CEC PRESSURE TIME SLOPE       (PSI/SEC):.....',
     1     CTSLP1
      WRITE (*,*) ' DYTRAN PRESSURE TIME SLOPE    (PSI/SEC):.....',
     1     DTSLP1
      WRITE (*,*) ' *********** SECOND SYNCHRONIZATION ***********'
      WRITE (*,*) ' CEC PRESSURE TIME SLOPE       (PSI/SEC):     ',
     1     CTSLP2
      WRITE (*,*) ' DYTRAN PRESSURE TIME SLOPE    (PSI/SEC):.....',
     1     DTSLP2
      WRITE (*,*) ' 1ST CYCLE START PRESSURE:.................',
     1     PRESDAT(1)
      WRITE (*,*) ' 1ST CYCLE END PRESSURE:...................',
     1     PRESDAT(128)
```

```
      WRITE (*,*) ' 2ND CYCLE START PRESSURE:.....',
     1     PRESDAT(129)
      WRITE (*,*) ' 2ND CYCLE END PRESSURE:.....',
     1     PRESDAT(256)
      WRITE (*,*) ' 3RD CYCLE START PRESSURE:.....',
     1     PRESDAT(257)
      WRITE (*,*) ' 3RD CYCLE END PRESSURE:.....',
     1     PRESDAT(384)
C
C
C     WRITE (*,'(A\)') ' ENTER Y FOR PRINTOUT:'
CREAD  (*,'(A1)') RESPONSE
CIF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 4000
C
C
OPEN (UNIT=9,FILE='PRN')
WRITE (9,*) ' DATA FILE:',FNAME
WRITE (9,*) ' ',DATE,'    ',CTIME
WRITE (9,'(A)') '0'
WRITE (9,'(A)') '0'
WRITE (9,*) ' MASS (GRAMS):.....',CMASS
      WRITE (9,*) ' VOLUME RATIO:.....',CVR
      WRITE (9,*) ' PERIOD (SECS):.....',CPER
      WRITE (9,*) ' TEMPERATURE RANGE:..... LN2'
      WRITE (9,*) ' REYNOLDS NO.:.....',
     1     CREYNOLD
      WRITE (9,*) ' PECLET NO.:.....',PEC
      WRITE (9,*) ' FIRST HARMONIC PRESSURE MAGNITUDE:.....',
     1     PMAG(1)
      WRITE (9,*) ' SECOND HARMONIC PRESSURE MAGNITUDE:.....',
     1     PMAG(2)
      WRITE (9,*) ' FIRST HARMONIC PRESSURE PHASE:.....',
     1     PPHASE(1)
      WRITE (9,*) ' SECOND HARMONIC PRESSURE PHASE:.....',
     1     PPHASE(2)
      WRITE (9,'(A)') '0'
      WRITE (9,'(A)') '0'
      WRITE (9,*) ' ********** FIRST PRESSURE CALIBRATION *******'
      WRITE (9,*) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A11
      WRITE (9,*) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A12
      WRITE (9,*) ' ********** SECOND PRESSURE CALIBRATION ******'
      WRITE (9,*) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A21
      WRITE (9,*) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A22
      WRITE (9,*) ' ********** FIRST SYNCHRONIZATION ***********'
      WRITE (9,*)' CEC PRESSURE TIME SLOPE       (PSI/SEC):.....',CTSLP1
      WRITE (9,*)' DYTRAN PRESSURE TIME SLOPE    (PSI/SEC):.....',DTSLP1
      WRITE (9,*)' ********** SECOND SYNCHRONIZATION **********'
      WRITE (9,*)' CEC PRESSURE TIME SLOPE       (PSI/SEC):.....',CTSLP2
      WRITE (9,*)' DYTRAN PRESSURE TIME SLOPE    (PSI/SEC):.....',DTSLP2
      WRITE (9,'(A)') '0'
      WRITE (9,'(A)') '0'
      WRITE (9,*) ' 1ST CYCLE START PRESSURE:.....',
     1     PRESDAT(1)
      WRITE (9,*) ' 1ST CYCLE END PRESSURE:.....',
     1     PRESDAT(128)
      WRITE (9,*) ' 2ND CYCLE START PRESSURE:.....',
     1     PRESDAT(129)
      WRITE (9,*) ' 2ND CYCLE END PRESSURE:.....',
     1     PRESDAT(256)
      WRITE (9,*) ' 3RD CYCLE START PRESSURE:.....',
     1     PRESDAT(257)
      WRITE (9,*) ' 3RD CYCLE END PRESSURE:.....',
     1     PRESDAT(384)
      WRITE (9,'(A)') '1'
C
      GOTO 4000
C
C
C
C**********************************************************************
C
C      ********** WRITE DATA TO FILE ***************
C
C     OPEN FILE FOR STORAGE OF PARAMETERS AND RESULTS
 4002      CONTINUE
      OPEN (UNIT=2,FILE='NCORE.DAT',STATUS='OLD')
C
C      MOVE TO END OF FILE AND APPEND NEW DATA THERE
    8  READ (2,'(A14)',END=9)JUNK
      GOTO 8
    9  CONTINUE
```

```
      BACKSPACE 2
C
      WRITE(2,3000)FNAME,PERIOD,MASS,VOLRAT,REYNOLD,PEC,RRCKH,RRCKS,
     1 VMAG(0),VMAG(1),PMAG(0),PMAG(5),PMAG(6),PMAG(3),PMAG(1),PMAG(2),
     2 PPHASE(1),PPHASE(2),PPHASE(3)
 3000      FORMAT(A14,18F14.4)
      CLOSE (UNIT=2)
      GOTO 4000
C
C
C**********************************************************************
C
C
C
 4003      CONTINUE
      CALL PLOTPV
     1 (PER3,VMIN,VMAX,VTICS,PERIOD,TIME(1),VOLDAT(1),PMIN,PMAX,PTICS,
     2 PRESDAT(1),CMASS,CVR,CPER,CREYNOLD,DATE,CTIME)
      GOTO 4000
C
C
C      CALL PLOTVMSPEC (VMAGMIN,VMAGMAX,VMAGTICS,VMAG(1),DATE,CTIME)
C
C
 4004      CONTINUE
      CALL PLOTMSPEC (PMAGMIN,PMAGMAX,PMAGTICS,PMAG(1),DATE,CTIME)
      GOTO 4000
C
C
 4005      CONTINUE
      CALL PLOTPSPEC
     1 (PPHASEMIN,PPHASEMAX,PPHASETICS,PPHASE(1),DATE,CTIME)
      GOTO 4000
C
C
 4006      CONTINUE
      CALL PLOTPCAL1
     1 (DYT1MIN,DYT1MAX,YTICS2,DYTCAL1(1),CEC1MIN,CEC1MAX,YTICS1,
     2 CECCAL1(1),PCALC(1),CAL1MAX,CAL1TIME(1),DATE,CTIME,TS1,VS1)
      GOTO 4000
C
C
 4007      CONTINUE
      CALL PLOTPCAL2
     1 (DYT2MIN,DYT2MAX,YTICS4,DYTCAL2(1),CEC2MIN,CEC2MAX,YTICS3,
     2 CECCAL2(1),PCALC2(1),CAL1MAX,CAL1TIME(1),DATE,CTIME,TS3,VS2)
      GOTO 4000
C
C
 4008      CONTINUE
      CALL PLOTPSYNC
     1 (DSYNCMIN,DSYNCMAX,DSYNTICS,SYNC1TIME(1),DYTSYNC1(1),
     2 SYNC2TIME(1),DYTSYNC2(1),YCALC(1),CSYNCMIN,CSYNCMAX,
     3 CSYNTICS,CECSYNC1(1),CECSYNC2(1),DATE,CTIME,TS2,TS4)
      GOTO 4000
C
C
 4009 CONTINUE
      OPEN (UNIT=12,FILE='PDAT.DAT')
      OPEN (UNIT=13,FILE='PISO.DAT')
      OPEN (UNIT=14,FILE='PADI.DAT')
      DO 711, I=1,128
           WRITE(12,'(2F14.4)')TIME(I),PPLOTDAT(I)
           WRITE(13,'(2F14.4)')TIME(I),PPLOTDAT(I+128)
           WRITE(14,'(2F14.4)')TIME(I),PPLOTDAT(I+256)
  711      CONTINUE
      CLOSE (12)
      CLOSE (13)
      CLOSE (14)
      CALL PDATAPLOT (PERIOD,TIME(1),PPLOTDAT(1),DATE,CTIME)
      GOTO 4000
C
C
 4010      CONTINUE
      CALL PLOTPTDRIFT (TIME(1),AVGG(1),DATE,CTIME)
      GOTO 4000
 1000  FORMAT(2A14,9F14.0)
 1010  FORMAT(2F14.0)
 1020      FORMAT(5F14.0)
 2000  FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
```

```
 5000  FORMAT(F10.4)
 4012CONTINUE
END
$TITLE: 'SUBS.FOR'
$SUBTITLE: 'SUBROUTINE PLOTPV; PLOT P-V PLOTS'
$NOTRUNCATE
$INCLUDE:'HGRAPHALIAS.FOR'
      SUBROUTINE PLOTPV
     1 (PER3,VMIN,VMAX,VTICS,PERIOD,TIME,VOLDAT,PMIN,PMAX,PTICS,PRESDAT,
     2 CMASS,CVR,CPER,CREYNOLD,DATE,CTIME)
      REAL TIME(1),VOLDAT(1),PRESDAT(1),PER3,VMIN,VMAX,VTICS,
     1   PERIOD,PMIN,PMAX,PTICS
      CHARACTER DATE*14,CTIME*14,CMASS*11,CVR*7,CPER*9,CREYNOLD*14
$INCLUDE: 'HGRAPHCONSTS.FOR'
C
C
C     ********* PLOT PV DATA ***********
C
C
      CALL INIPLT(0,.FALSE.,1.0)
      CALL GRAPHBOUNDARY(1000,9000,1000,6000)
      CALL JUSTIFYSTRING(6000,6500,DATE//CTIME//CHAR(13)//CHAR(0),0,
     1 2,BASE,BASE)
      CALL WRITESTRING('MASS: '//CMASS//'grams'//CHAR(13)//CHAR(0),0,2)
      CALL WRITESTRING('VOLUME RATIO:'//CVR//CHAR(13)//CHAR(0),0,2)
      CALL WRITESTRING('PERIOD:'//CPER//' SECONDS'//CHAR(13)//CHAR(0),
     1 0,2)
      CALL VIEWPORT(0,5000,3500,7000)
      CALL GRAPHBOUNDARY(1000,4750,700,3000)
      CALL SCALE(0.0,PER3,VMIN,VMAX)
      CALL AXIS(PERIOD,'10.1'//CHAR(0),'TIME (SECONDS)'//CHAR(0),2,
     1 VTICS,'10.1'//CHAR(0),'VOLUME (cu. in.)'//CHAR(0),2)
      CALL POLYLINE(TIME,VOLDAT,384,0,0,0,0,0)
C
C
      CALL VIEWPORT(0,5000,0,3500)
      CALL SCALE(0.0,PER3,PMIN,PMAX)
      CALL AXIS(PERIOD,'10.1'//CHAR(0),'TIME (SECONDS)'//CHAR(0),2,
     1 PTICS,'10.0'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(TIME,PRESDAT,384,0,0,0,0,0)
C
C
      CALL VIEWPORT(5000,10000,0,7000)
      CALL GRAPHBOUNDARY(1000,4750,700,4500)
      CALL SCALE(VMIN,VMAX,PMIN,PMAX)
      CALL AXIS(VTICS,'10.0'//CHAR(0),'VOLUME (cu. in.)'//CHAR(0),2,
     1 PTICS,'10.0'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(VOLDAT,PRESDAT,380,0,0,0,0,0)
      CALL ENDPLT
C
C
      WRITE(*,2000)
      READ (*,'(A1)')RESPONSE
      IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 501
      CALL FILEUPDATE(ICOUNT)
C
C
      CALL INIPLT(ICOUNT,.FALSE.,1.0)
      CALL JUSTIFYSTRING(6000,6500,DATE//CTIME//CHAR(13)//CHAR(0),0,
     1 2,BASE,BASE)
      CALL WRITESTRING('MASS: '//CMASS//'grams'//CHAR(13)//CHAR(0),0,2)
      CALL WRITESTRING('VOLUME RATIO:'//CVR//CHAR(13)//CHAR(0),0,2)
      CALL WRITESTRING('PERIOD:'//CPER//' SECONDS'//CHAR(13)//CHAR(0),
     1 0,2)
      CALL VIEWPORT(0,5000,3500,7000)
      CALL GRAPHBOUNDARY(1000,4750,700,3000)
      CALL SCALE(0.0,PER3,VMIN,VMAX)
      CALL AXIS(PERIOD,'10.1'//CHAR(0),'TIME (SECONDS)'//CHAR(0),2,
     1 VTICS,'10.1'//CHAR(0),'VOLUME (cu. in.)'//CHAR(0),2)
      CALL POLYLINE(TIME,VOLDAT,384,0,0,0,0,0)
C
C
      CALL VIEWPORT(0,5000,0,3500)
      CALL SCALE(0.0,PER3,PMIN,PMAX)
      CALL AXIS(PERIOD,'10.1'//CHAR(0),'TIME (SECONDS)'//CHAR(0),2,
     1 PTICS,'10.0'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(TIME,PRESDAT,384,0,0,0,0,0)
C
C
      CALL VIEWPORT(5000,10000,0,7000)
```

```
      CALL GRAPHBOUNDARY(1000,4750,700,4500)
      CALL SCALE(VMIN,VMAX,PMIN,PMAX)
      CALL AXIS(VTICS,'10.0'//CHAR(0),'VOLUME (cu. in.)'//CHAR(0),2,
     1 PTICS,'10.0'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(VOLDAT,PRESDAT,380,0,0,0,0,0)
      CALL ENDPLT
  501 CONTINUE
 2000 FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
RETURN
END
$SUBTITLE: 'SUBROUTINE PLOTMSPEC;  PLOT MAGNITUDE SPECTRUM'
$PAGE
C
C
      SUBROUTINE PLOTMSPEC (PMAGMIN,PMAGMAX,PMAGTICS,PMAG,DATE,CTIME)
C
C
      REAL PMAGMIN,PMAGMAX,PMAGTICS,PMAG(1)
      CHARACTER DATE*14,CTIME*14
$INCLUDE: 'HGRAPHCONSTS.FOR'
C
C     ************* SPECTRUM MAGNITUDE PLOT ****************
C
      CALL INIPLT(0,.FALSE.,1.0)
      CALL SCALE (0.0,7.0,PMAGMIN,PMAGMAX)
      CALL AXIS (1.0,'5.0'//CHAR(0),'FREQUENCY'//CHAR(0),2,
     1 PMAGTICS,'10.2'//CHAR(0),'PRESSURE MAGINITUDES'//CHAR(0),2)
      CALL ARROW (1.0,PMAGMIN,1.0,PMAG(1),0,1,.TRUE.)
      CALL ARROW (2.0,PMAGMIN,2.0,PMAG(2),0,1,.TRUE.)
      CALL ARROW (3.0,PMAGMIN,3.0,PMAG(3),0,1,.TRUE.)
      CALL ARROW (4.0,PMAGMIN,4.0,PMAG(4),0,1,.TRUE.)
      CALL ARROW (5.0,PMAGMIN,5.0,PMAG(5),0,1,.TRUE.)
      CALL ARROW (6.0,PMAGMIN,6.0,PMAG(6),0,1,.TRUE.)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  HARMONIC PRESSURE MAGINITUDES'//CHAR(0),0,2)
      CALL ENDPLT
C
C
      WRITE(*,2000)
      READ (*,'(A1)')RESPONSE
      IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 600
      CALL FILEUPDATE(ICOUNT)
C
C
      CALL INIPLT(ICOUNT,.FALSE.,1.0)
      CALL SCALE (0.0,7.0,PMAGMIN,PMAGMAX)
      CALL AXIS (1.0,'5.0'//CHAR(0),'FREQUENCY'//CHAR(0),2,
     1 PMAGTICS,'10.2'//CHAR(0),'PRESSURE MAGINITUDES'//CHAR(0),2)
      CALL ARROW (1.0,PMAGMIN,1.0,PMAG(1),0,1,.FALSE.)
      CALL ARROW (2.0,PMAGMIN,2.0,PMAG(2),0,1,.FALSE.)
      CALL ARROW (3.0,PMAGMIN,3.0,PMAG(3),0,1,.FALSE.)
      CALL ARROW (4.0,PMAGMIN,4.0,PMAG(4),0,1,.FALSE.)
      CALL ARROW (5.0,PMAGMIN,5.0,PMAG(5),0,1,.FALSE.)
      CALL ARROW (6.0,PMAGMIN,6.0,PMAG(6),0,1,.FALSE.)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  HARMONIC PRESSURE MAGINITUDES'//CHAR(0),0,2)
      CALL ENDPLT
  600 CONTINUE
 2000 FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
RETURN
END
$SUBTITLE: 'SUBROUTINE PLOTPSPEC;  PLOT PHASE SPECTRUM'
$PAGE
C
C
      SUBROUTINE PLOTPSPEC
     1 (PPHASEMIN,PPHASEMAX,PPHASETICS,PPHASE,DATE,CTIME)
      CHARACTER DATE*14,CTIME*14
      REAL PPHASEMIN,PPHASEMAX,PPHASETICS,PPHASE(1)
$INCLUDE: 'HGRAPHCONSTS.FOR'
C     ********* PHASE PLOTS ***************
C
      CALL INIPLT (0,.FALSE.,1.0)
      CALL SCALE (0.0,7.0,PPHASEMIN,PPHASEMAX)
      CALL AXIS (1.0,'5.0'//CHAR(0),'FREQUENCY'//CHAR(0),2,
     1 PPHASETICS,'10.2'//CHAR(0),'PHASE SHIFT (DEGREES)'//CHAR(0),2)
      CALL ARROW (1.0,PPHASEMIN,1.0,PPHASE(1),0,1,.TRUE.)
      CALL ARROW (2.0,PPHASEMIN,2.0,PPHASE(2),0,1,.TRUE.)
```

```
      CALL ARROW (3.0,PPHASEMIN,3.0,PPHASE(3),0,1,.TRUE.)
C     CALL ARROW (4.0,PPHASEMIN,4.0,PPHASE(4),0,1,.TRUE.)
C     CALL ARROW (5.0,PPHASEMIN,5.0,PPHASE(5),0,1,.TRUE.)
C     CALL ARROW (6.0,PPHASEMIN,6.0,PPHASE(6),0,1,.TRUE.)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  HARMONIC PHASE SHIFTS'//CHAR(0),0,2)
      CALL ENDPLT
C
C
      WRITE(*,2000)
      READ (*,'(A1)')RESPONSE
      IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 601
      CALL FILEUPDATE(ICOUNT)
C
C
      CALL INIPLT (ICOUNT,.FALSE.,1.0)
      CALL SCALE (0.0,7.0,PPHASEMIN,PPHASEMAX)
      CALL AXIS (1.0,'5.0'//CHAR(0),'FREQUENCY'//CHAR(0),2,
     1 PPHASETICS,'10.2'//CHAR(0),'PHASE SHIFT (DEGREES)'//CHAR(0),2)
      CALL ARROW (1.0,PPHASEMIN,1.0,PPHASE(1),0,1,.FALSE.)
      CALL ARROW (2.0,PPHASEMIN,2.0,PPHASE(2),0,1,.FALSE.)
      CALL ARROW (3.0,PPHASEMIN,3.0,PPHASE(3),0,1,.FALSE.)
      CALL ARROW (4.0,PPHASEMIN,4.0,PPHASE(4),0,1,.FALSE.)
      CALL ARROW (5.0,PPHASEMIN,5.0,PPHASE(5),0,1,.FALSE.)
      CALL ARROW (6.0,PPHASEMIN,6.0,PPHASE(6),0,1,.FALSE.)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  HARMONIC PHASE SHIFTS'//CHAR(0),0,2)
      CALL ENDPLT
  601     CONTINUE
 2000     FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
RETURN
END
$SUBTITLE: 'SUBROUTINE PLOTPCAL1; PLOT 1ST PRESSURE CALIBRATION'
$PAGE
C
C
      SUBROUTINE PLOTPCAL1
     1 (DYT1MIN,DYT1MAX,YTICS2,DYTCAL1,CEC1MIN,CEC1MAX,YTICS1,CECCAL1,
     2 PCALC,CAL1MAX,CAL1TIME,DATE,CTIME,TS1,VS1)
      REAL DYTCAL1(1),CECCAL1(1),PCALC(1),CAL1TIME(1)
      CHARACTER DATE*14,CTIME*14,TS1*10,VS1*10
$INCLUDE: 'HGRAPHCONSTS.FOR'
C
C     *********PRESSURE CALIBRATION 1 ***********
C
      CALL INIPLT(0,.FALSE.,1.0)
      CALL SCALE(0.0,CAL1MAX,DYT1MIN,DYT1MAX)
      CALL SETYTICS(DYT1MIN,DYT1MAX,YTICS2,200,LEFT,2,1)
      CALL SETYLABEL( RIGHT, MAJOR, 2, '10.0'//CHAR(0))
      CALL SETYAXIS(CAL1MAX,DYT1MIN,DYT1MAX,LINEAR)
      CALL DRAWYAXIS( 'PIEZO PRESSURE (LSBs)'//CHAR(0), 2,
     1  RIGHT, CENTER)
      CALL POLYLINE(CAL1TIME,DYTCAL1,40,0,2,1,1,9)
      CALL WRITELEGEND('PIEZO TRANSDUCER'//CHAR(0),0,2,1,1,9)
C
      CALL BACKGRID(0,MAJOR)
      CALL SCALE (0.0,CAL1MAX,CEC1MIN,CEC1MAX)
      CALL AXIS (5.0,'6.1'//CHAR(0),'TIME (SEC)'//CHAR(0),2,
     .           YTICS1,'6.1'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(CAL1TIME,CECCAL1,40,0,0,0,0,3)
      CALL WRITELEGEND('CEC TRANSDUCER'//CHAR(0),0,0,0,0,3)
      CALL POLYLINE(CAL1TIME,PCALC,40,0,0,0,0,0)
      CALL WRITELEGEND('CORRECTED PIEZO PRESSURE       VOLTAGE SLOPE'
     1 //VS1//'   TIME SLOPE'//TS1//CHAR(0),0,0,0,0,0)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  PRESSURE CALIBRATION 1'//CHAR(0),0,2)
      CALL ENDPLT
C
      WRITE(*,2000)
      READ (*,'(A1)')RESPONSE
      IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 998
      CALL FILEUPDATE(ICOUNT)
C
      CALL INIPLT(ICOUNT,.FALSE.,1.0)
      CALL SCALE(0.0,CAL1MAX,DYT1MIN,DYT1MAX)
      CALL SETYTICS(DYT1MIN,DYT1MAX,YTICS2,200,LEFT,2,1)
      CALL SETYLABEL( RIGHT, MAJOR, 2, '10.0'//CHAR(0))
```

```
      CALL SETYAXIS(CAL1MAX,DYT1MIN,DYT1MAX,LINEAR)
      CALL DRAWYAXIS( 'PIEZO PRESSURE (LSBs)'//CHAR(0), 2,
     1  RIGHT, CENTER)
      CALL POLYLINE(CAL1TIME,DYTCAL1,40,0,2,1,1,9)
      CALL WRITELEGEND('PIEZO TRANSDUCER'//CHAR(0),0,2,1,1,9)
C
      CALL BACKGRID(0,MAJOR)
      CALL SCALE (0.0,CAL1MAX,CEC1MIN,CEC1MAX)
      CALL AXIS (5.0,'6.1'//CHAR(0),'TIME (SEC)'//CHAR(0),2,
     .           YTICS1,'6.1'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(CAL1TIME,CECCAL1,40,0,0,0,0,3)
      CALL WRITELEGEND('CEC TRANSDUCER'//CHAR(0),0,0,0,0,3)
      CALL POLYLINE(CAL1TIME,PCALC,40,0,0,0,0,0)
      CALL WRITELEGEND('CORRECTED PIEZO PRESSURE       VOLTAGE SLOPE'
     1 //VS1//'   TIME SLOPE'//TS1//CHAR(0),0,0,0,0,0)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  PRESSURE CALIBRATION 1'//CHAR(0),0,2)
      CALL ENDPLT
C
C
  998     CONTINUE
 2000     FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
RETURN
END
$SUBTITLE: 'SUBROUTINE PLOTPCAL2; PLOT 2ND PRESSURE CALIBRATION'
$PAGE
C
C
C
      SUBROUTINE PLOTPCAL2
     1 (DYT2MIN,DYT2MAX,YTICS4,DYTCAL2,CEC2MIN,CEC2MAX,YTICS3,CECCAL2,
     2 PCALC2,CAL1MAX,CAL1TIME,DATE,CTIME,TS3,VS2)
      REAL DYTCAL2(1),CECCAL2(1),PCALC2(1),CAL1TIME(1)
      CHARACTER DATE*14,CTIME*14,TS3*10,VS2*10
$INCLUDE: 'HGRAPHCONSTS.FOR'
C
C     ********** PRESSURE CALIBRATION 2 ***********
C
      CALL INIPLT(0,.FALSE.,1.0)
      CALL SCALE(0.0,CAL1MAX,DYT2MIN,DYT2MAX)
      CALL SETYTICS(DYT2MIN,DYT2MAX,YTICS4,200,LEFT,2,1)
      CALL SETYLABEL( RIGHT, MAJOR, 2, '10.0'//CHAR(0))
      CALL SETYAXIS(CAL1MAX,DYT2MIN,DYT2MAX,LINEAR)
      CALL DRAWYAXIS( 'PIEZO PRESSURE (LSBs)'//CHAR(0), 2,
     1  RIGHT, CENTER)
      CALL POLYLINE(CAL1TIME,DYTCAL2,40,0,2,1,1,9)
      CALL WRITELEGEND('PIEZO TRANSDUCER'//CHAR(0),0,2,1,1,9)
C
      CALL BACKGRID(NONE,MAJOR)
      CALL SCALE (0.0,CAL1MAX,CEC2MIN,CEC2MAX)
      CALL AXIS (5.0,'6.1'//CHAR(0),'TIME (SEC)'//CHAR(0),2,
     .           YTICS3,'6.1'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(CAL1TIME,CECCAL2,40,0,0,0,0,3)
      CALL WRITELEGEND('CEC TRANSDUCER'//CHAR(0),0,0,0,0,3)
      CALL POLYLINE(CAL1TIME,PCALC2,40,0,0,0,0,0)
      CALL WRITELEGEND('CORRECTED PIEZO PRESSURE       VOLTAGE SLOPE'
     1 //VS2//'   TIME SLOPE'//TS3//CHAR(0),0,0,0,0,0)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  PRESSURE CALIBRATION 2'//CHAR(0),0,2)
      CALL ENDPLT
C
      WRITE(*,2000)
      READ (*,'(A1)')RESPONSE
      IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 999
      CALL FILEUPDATE(ICOUNT)
C
      CALL INIPLT(ICOUNT,.FALSE.,1.0)
      CALL SCALE(0.0,CAL1MAX,DYT2MIN,DYT2MAX)
      CALL SETYTICS(DYT2MIN,DYT2MAX,YTICS4,200,LEFT,2,1)
      CALL SETYLABEL( RIGHT, MAJOR, 2, '10.0'//CHAR(0))
      CALL SETYAXIS(CAL1MAX,DYT2MIN,DYT2MAX,LINEAR)
      CALL DRAWYAXIS( 'PIEZO PRESSURE (LSBs)'//CHAR(0), 2,
     1  RIGHT, CENTER)
      CALL POLYLINE(CAL1TIME,DYTCAL2,40,0,2,1,1,9)
      CALL WRITELEGEND('PIEZO TRANSDUCER'//CHAR(0),0,2,1,1,9)
C
      CALL BACKGRID(NONE,MAJOR)
      CALL SCALE (0.0,CAL1MAX,CEC2MIN,CEC2MAX)
```

```
      CALL AXIS (5.0,'6.1'//CHAR(0),'TIME (SEC)'//CHAR(0),2,
     .           YTICS3,'6.1'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(CAL1TIME,CECCAL2,40,0,0,0,0,3)
      CALL WRITELEGEND('CEC TRANSDUCER'//CHAR(0),0,0,0,0,3)
      CALL POLYLINE(CAL1TIME,PCALC2,40,0,0,0,0,0)
      CALL WRITELEGEND('CORRECTED PIEZO PRESSURE       VOLTAGE SLOPE'
     1 //VS2//'   TIME SLOPE'//TS3//CHAR(0),0,0,0,0,0)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,
     1 1,BASE,BASE)
      CALL WRITESTRING('  PRESSURE CALIBRATION 2'//CHAR(0),0,2)
      CALL ENDPLT
  999 CONTINUE
 2000 FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
RETURN
END
$SUBTITLE: 'SUBROUTINE PLOTPSYNC; PLOT PRESSURE SYNCHRONIZATIONS'
$PAGE
C
C
      SUBROUTINE PLOTPSYNC
     1 (DSYNCMIN,DSYNCMAX,DSYNTICS,SYNC1TIME,DYTSYNC1,SYNC2TIME,
     2 DYTSYNC2,YCALC,CSYNCMIN,CSYNCMAX,CSYNTICS,CECSYNC1,CECSYNC2,
     3 DATE,CTIME,TS2,TS3)
      REAL SYNC1TIME(1),DYTCYNC1(1),SYNC2TIME(1),DYTSYNC2(1),YCALC(1),
     1 CECSYNC1(1),CECSYNC2(1)
      CHARACTER DATE*14,CTIME*14,TS2*10,TS3*10
$INCLUDE: 'HGRAPHCONSTS.FOR'
C
C     ********** PRESSURE SYNCHRONIZATION ************
C
      CALL INIPLT(0,.FALSE.,1.0)
      CALL SCALE (0.0,15.0,DSYNCMIN,DSYNCMAX)
      CALL AXIS (2.0,'4.0'//CHAR(0),'TIME (SEC)'//CHAR(0),2
     1 ,DSYNTICS,'10.0'//CHAR(0),'PIEZO PRESSURE (LSBs)'//CHAR(0),2)
      CALL POLYLINE (SYNC1TIME,DYTSYNC1,10,0,2,1,1,9)
      CALL POLYLINE (SYNC2TIME,DYTSYNC2,30,0,2,1,1,9)
      CALL WRITELEGEND('MEASURED PIEZO OUTPUT'//CHAR(0),0,2,1,1,9)
      CALL POLYLINE (SYNC1TIME,YCALC,10,0,0,0,0,0)
      CALL POLYLINE (SYNC2TIME,YCALC(11),30,0,0,0,0,0)
      CALL WRITELEGEND('CORRECTED PIEZO PRESSURE        TIME SLOPE 1:'
     1 //TS2//CHAR(0),0,0,0,0,0)
C
      CALL SCALE (0.0,15.0,CSYNCMIN,CSYNCMAX)
      CALL SETYTICS(CSYNCMIN,CSYNCMAX,CSYNTICS,200,LEFT,2,1)
      CALL SETYLABEL(RIGHT,MAJOR,2,'10.2'//CHAR(0))
      CALL SETYAXIS(15.0,CSYNCMIN,CSYNCMAX,LINEAR)
      CALL DRAWYAXIS('CEC PRESSURE (PSI)'//CHAR(0),2,RIGHT,CENTER)
      CALL POLYLINE(SYNC1TIME,CECSYNC1,10,0,0,0,0,3)
      CALL POLYLINE(SYNC2TIME,CECSYNC2,30,0,0,0,0,3)
      CALL WRITELEGEND('MEASURED CEC PRESSURE           TIME SLOPE 2:'
     1 //TS3//CHAR(0),0,0,0,0,3)
      CALL JUSTIFYSTRING(2000,6700,DATE//CTIME//CHAR(0),0,1,BASE,BASE)
      CALL WRITESTRING('  PRESSURE SYNCHRONIZATION'//CHAR(0),0,2)
      CALL ENDPLT
C
      WRITE(*,2000)
      READ (*,'(A1)')RESPONSE
      IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 997
      CALL FILEUPDATE(ICOUNT)
C
      CALL INIPLT(ICOUNT,.FALSE.,1.0)
      CALL SCALE (0.0,15.0,DSYNCMIN,DSYNCMAX)
      CALL AXIS (2.0,'4.0'//CHAR(0),'TIME (SEC)'//CHAR(0),2
     1 ,DSYNTICS,'10.0'//CHAR(0),'PIEZO PRESSURE (LSBs)'//CHAR(0),2)
      CALL POLYLINE (SYNC1TIME,DYTSYNC1,10,0,2,1,1,9)
      CALL POLYLINE (SYNC2TIME,DYTSYNC2,30,0,2,1,1,9)
      CALL WRITELEGEND('MEASURED PIEZO OUTPUT'//CHAR(0),0,2,1,1,9)
      CALL POLYLINE (SYNC1TIME,YCALC,10,0,0,0,0,0)
      CALL POLYLINE (SYNC2TIME,YCALC(11),30,0,0,0,0,0)
      CALL WRITELEGEND('CORRECTED PIEZO PRESSURE        TIME SLOPE 1:'
     1 //TS2//CHAR(0),0,0,0,0,0)
C
      CALL SCALE (0.0,15.0,CSYNCMIN,CSYNCMAX)
      CALL SETYTICS(CSYNCMIN,CSYNCMAX,CSYNTICS,200,LEFT,2,1)
      CALL SETYLABEL(RIGHT,MAJOR,2,'10.2'//CHAR(0))
      CALL SETYAXIS(15.0,CSYNCMIN,CSYNCMAX,LINEAR)
      CALL DRAWYAXIS('CEC PRESSURE (PSI)'//CHAR(0),2,RIGHT,CENTER)
      CALL POLYLINE(SYNC1TIME,CECSYNC1,10,0,0,0,0,3)
      CALL POLYLINE(SYNC2TIME,CECSYNC2,30,0,0,0,0,3)
      CALL WRITELEGEND('MEASURED CEC PRESSURE           TIME SLOPE 2:'
```

```
     1 //TS3//CHAR(0),0,0,0,0,3)
       CALL JUSTIFYSTRING(2000,6700,DATE//CTIME//CHAR(0),0,1,BASE,BASE)
       CALL WRITESTRING('  PRESSURE SYNCHRONIZATION'//CHAR(0),0,2)
       CALL ENDPLT
  997      CONTINUE
 2000      FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
RETURN
END
$SUBTITLE:'SUBROUTINE SCALES; FIND CORRECT SCALE FACTORS FOR PLOTTING'
$PAGE
C
C
C
      SUBROUTINE SCALES(DATA,N,MINVALUE,MAXVALUE,TICS)
C
C
      REAL MINVALUE,MAXVALUE,TICS,DATA(1)
      MAXVALUE=DATA(1)
      MINVALUE=DATA(1)
      DO 10, I=1,N
           MAXVALUE=AMAX1(MAXVALUE,DATA(I))
           MINVALUE=AMIN1(MINVALUE,DATA(I))
   10      CONTINUE
      DIFF=MAXVALUE-MINVALUE
C     WRITE (*,*) MINVALUE,MAXVALUE,DIFF
      DEC=10.0**ANINT(ALOG10(DIFF)-1)
      IF (DIFF/DEC.LT.10) THEN
                TICS=1*DEC
           ELSEIF (DIFF/DEC.LT.20) THEN
                TICS=2*DEC
           ELSE
                TICS=5*DEC
      ENDIF
      MINVALUE=AINT(MINVALUE/TICS)*TICS
      MAXVALUE=AINT(MAXVALUE/TICS)*TICS
      IF (MINVALUE.LT. 0.0)  MINVALUE=MINVALUE-TICS
      IF (MAXVALUE.GT. 0.0)  MAXVALUE=MAXVALUE+TICS
      RETURN
      END
$SUBTITLE:'SUBROUTINE LSSQLN; LEAST SQUARES LINE FIT'
$PAGE
      SUBROUTINE LSSQLN(X,Y,N,SLOPE,YINTRC,YCALC,STDERR)
C
C
C     LEAST SQUARES CURVEFIT TO A LINE
C
      DIMENSION X(1),Y(1),YCALC(1)
      SXS=0
      SX =0
      SY =0
      SXY=0
      DO 10,I=1,N
           SXS=SXS+X(I)**2
           SX =SX +X(I)
           SXY=SXY+X(I)*Y(I)
           SY =SY +Y(I)
   10      CONTINUE
      SLOPE=(N*SXY-SX*SY)/(N*SXS-SX**2)
      YINTRC=(SY*SXS-SXY*SX)/(N*SXS-SX**2)
      DO 20, I=1,N
           YCALC(I)=X(I)*SLOPE+YINTRC
           SYYCAL=SYYCAL+(Y(I)-YCALC(I))**2
   20      CONTINUE
      STDERR=SQRT(SYYCAL/(N-1))
      RETURN
      END
$SUBTITLE:'SUBROUTINE LSSQ3D; LST SQRS LIN FIT TO 2 IND. VARIABLES'
$PAGE
      SUBROUTINE LSSQ3D(P,V,T,N,A0,A1,A2)
      REAL P(N),V(N),T(N)
      SP=0
      SV=0
      ST=0
      SV2=0
      ST2=0
      SPV=0
      SPT=0
      SVT=0
      DO 10, I=1,N
```

```
            SP = SP+P(I)
            SV = SV+V(I)
            ST = ST+T(I)
            SV2= SV2+V(I)*V(I)
            ST2= ST2+T(I)*T(I)
            SPV= SPV+P(I)*V(I)
            SPT= SPT+P(I)*T(I)
            SVT= SVT+V(I)*T(I)
   10       CONTINUE
      A2=((N*SPT-SP*ST)/(N*SVT-SV*ST)-(N*SPV-SP*SV)/(N*SV2-SV*SV))
     1 /((N*ST2-ST*ST)/(N*SVT-SV*ST)-(N*SVT-SV*ST)/(N*SV2-SV*SV))
       A1=(N*SPT-SP*ST-A2*(N*ST2-ST*ST))/(N*SVT-SV*ST)
       A0=(SP-A1*SV-A2*ST)/N
       RETURN
       END
$SUBTITLE:'SUBROUTINE PDATAPLOT; PLOT PRESS. WAVE WITH ADIABAT & ISOT'
$PAGE
C
C
C
      SUBROUTINE PDATAPLOT (PERIOD,TIME,PPLOTDAT,DATE,CTIME)
      CHARACTER DATE*14,CTIME*14,RESPONSE*1
      REAL TIME(1),PPLOTDAT(1)
$INCLUDE: 'HGRAPHCONSTS.FOR'
C
C
      CALL SCALES(PPLOTDAT(1),384,PMIN,PMAX,PTICS)
      PER5=PERIOD/5.0
      CALL INIPLT(0,.FALSE.,1.0)
      CALL SCALE (0.0,PERIOD,PMIN,PMAX)
      CALL AXIS  (PER5,'10.1'//CHAR(0),'TIME (SECONDS)'//CHAR(0),2
     1     ,PTICS,'10.0'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(TIME,PPLOTDAT(1),128,0,0,0,0,0)
      CALL POLYLINE(TIME,PPLOTDAT(257),128,0,0,0,0,7)
      CALL POLYLINE(TIME,PPLOTDAT(129),128,0,0,0,0,8)
      CALL WRITELEGEND(' ACTUAL PRESSURE'//CHAR(0),0,0,0,0,0)
      CALL WRITELEGEND(' ADIABATIC PRESSURE'//CHAR(0),0,0,0,0,7)
      CALL WRITELEGEND(' ISOTHERMAL PRESSURE'//CHAR(0),0,0,0,0,8)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,1,BASE,BASE)
      CALL WRITESTRING('   PRESSURE CURVES'//CHAR(0),0,2)
      CALL ENDPLT
C
C
C
      WRITE (*,'(A\)') ' MAKE A HARD COPY FILE? (Y):'
READ  (*,'(A1)') RESPONSE
IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 100
CALL FILEUPDATE(ICOUNT)
C
C
CALL INIPLT(ICOUNT,.FALSE.,1.0)
CALL SCALE (0.0,PERIOD,PMIN,PMAX)
CALL AXIS  (PER5,'10.1'//CHAR(0),'TIME (SECONDS)'//CHAR(0),2
     1     ,PTICS,'10.0'//CHAR(0),'PRESSURE (PSI)'//CHAR(0),2)
      CALL POLYLINE(TIME,PPLOTDAT(1),128,0,0,0,0,0)
      CALL POLYLINE(TIME,PPLOTDAT(257),128,0,0,0,0,7)
      CALL POLYLINE(TIME,PPLOTDAT(129),128,0,0,0,0,8)
      CALL WRITELEGEND(' ACTUAL PRESSURE'//CHAR(0),0,0,0,0,0)
      CALL WRITELEGEND(' ADIABATIC PRESSURE'//CHAR(0),0,0,0,0,7)
      CALL WRITELEGEND(' ISOTHERMAL PRESSURE'//CHAR(0),0,0,0,0,8)
      CALL JUSTIFYSTRING(2000,6500,DATE//CTIME//CHAR(0),0,1,BASE,BASE)
      CALL WRITESTRING('   PRESSURE CURVES'//CHAR(0),0,2)
      CALL ENDPLT
  100     CONTINUE
      RETURN
      END
$SUBTITLE:'SUBROUTINE FFT; CALCULATES FAST FOURIER TRANSFORM'
$PAGE
C
C
C
      SUBROUTINE FFT (X,Y,N,M)
      REAL X(1),Y(1)
      N2=N
      DO 10, K=1,M
           N1=N2
           N2=N2/2
           E =6.283185307/N1
           A =0
           DO 20, J=1,N2
```

```
                C=COS(A)
                S=-SIN(A)
                A=J*E
                DO 30, I=J,N,N1
                      L=I+N2
                      XT  = X(I)-X(L)
                      X(I)= X(I)+X(L)
                      YT  = Y(I)-Y(L)
                      Y(I)= Y(I)+Y(L)
                      X(L)= C*XT-S*YT
                      Y(L)= C*YT+S*XT
                      CONTINUE
   30                 CONTINUE
   20           CONTINUE
   10      CONTINUE
C     ************************************************************
  100      J=1
       N1=N-1
       DO 104, I=1,N1
            IF (I.GE.J) GOTO 101
            XT=X(J)
            X(J)=X(I)
            X(I)=XT
            XT=Y(J)
            Y(J)=Y(I)
            Y(I)=XT
  101       K=N/2
  102       IF (K.GE.J) GOTO 103
            J=J-K
            K=K/2
            GOTO 102
  103       J=J+K
  104      CONTINUE
       DO 105, I=1,N
  105      CONTINUE
       RETURN
       END
$SUBTITLE:'SUBROUTINE DECOMP; DECOMPOSE PRESS. & VOL. INTO HARMONICS'
$PAGE
C
C
      SUBROUTINE DECOMP
     1 (VOLDAT,PRESDAT,PSYNC,VSYNC,PMAG,PPHASE,VMAG,PERIOD,
     2 PPLOTDAT)
C
C       ***** SUBROUTINE TO DECOMPOSE & NORMALIZE PRESSURE ******
C
C
      CHARACTER DATE*14,CTIME*14
C
      REAL VOLDAT(1),PRESDAT(1),PADIABATIC(384),PMAG(6),PPHASE(6)
     1 ,PTFANSFORM(384),PISOTHERM(384),KK,IM,RE,VTRANSFORM(128)
     2 ,ISOMAG,ISORE,ISOIM,VMAG(6),TIME(1),STROKE,OFFSET,PPLOTDAT(384)
     3 ,ADIABAT(384),AVGG(384),APMAG(4)
C
C
C
C       DEFINE PARAMETER kk =  RATIO OF SPECIFIC HEATS
C       THIS PARAMETER MUST BE CHANGED TO MATCH TEMPERATURE CONDITIONS
C
      PARAMETER (KK=1.667, PI=3.14159265)
C
C       CALCULATE ADIABATIC PRESSURE
C
      DO 5, I=1,384
           PISOTHERM(I)   = PSYNC*(VSYNC/VOLDAT(I))
           PADIABATIC(I) = PSYNC*((VSYNC/VOLDAT(I))**KK)
    5      CONTINUE
      CALL AVERAGE(PISOTHERM(1),AVGG(1))
      CALL AVERAGE(PADIABATIC(1),AVGG(1))
C
      DO 10, I=1,128
           PTRANSFORM(I)      = PRESDAT(I+128)
           PTRANSFORM(I+128) = PISOTHERM(I+128)
           PTRANSFORM(I+256) = PADIABATIC(I+128)
           VTRANSFORM(I)      = VOLDAT(I+128)
           PISOTHERM(I)       = PISOTHERM(I+128)
           PADIABATIC(I)      = PADIABATIC(I+128)
C
   10      CONTINUE
C
```

```
C
      DO 30, I=1,384
      PPLOTDAT(I) = PTRANSFORM(I)
   30      CONTINUE
C
C      PERFORM FFT
C
      CALL FFT(PTRANSFORM(1),PADIABATIC(1),128,7)
      CALL FFT(PISOTHERM(1), VTRANSFORM(1), 128,7)
C
      DO 20, I=1,4
           K = 129-I
           J = 1+I
           ACRE  = (PTRANSFORM(K) + PTRANSFORM(J))
           ISORE = (PISOTHERM(K)  + PISOTHERM(J))
           ADRE  = (PADIABATIC(K) + PADIABATIC(J))
           VORE  = (VTRANSFORM(K) + VTRANSFORM(J))
           ISOIM = (VTRANSFORM(K) - VTRANSFORM(J))
           ACIM  = (PADIABATIC(K) - PADIABATIC(J))
           ADIM  = (PTRANSFORM(J) - PTRANSFORM(K))
           VOIM  = (PISOTHERM(J)  - PISOTHERM(K))
           ACMAG   = ((ACRE**2)+(ACIM**2))**0.5
           ACPHASE = ATAN2(ACIM,ACRE)
           ADMAG   = ((ADRE**2)+(ADIM**2))**0.5
           ADPHASE = ATAN2(ADIM,ADRE)
           ISOMAG  = ((ISORE**2)+(ISOIM**2))**0.5
           PMAG(I) = (ACMAG-ISOMAG)/(ADMAG-ISOMAG)
           APMAG(I)= ACMAG
           VMAG(I) = (((VORE**2)+(VOIM**2))**0.5)/128
           PPHASE(I)=(ADPHASE-ACPHASE)*360/(2*PI*I)-1.40625-0.03/PERIOD
   20  CONTINUE
      PMAG(0)=PTRANSFORM(1)/128
      PMAG(5)=APMAG(1)/128
      PMAG(6)=APMAG(2)/128
      PMAG(3)=APMAG(3)/128
      VMAG(0)=VTRANSFORM(1)/128
      RETURN
      END
$SUBTITLE:'SUBROUTINE AVERAGE; AVERAGE THE PRESSURE WAVE'
$PAGE
C
C
      SUBROUTINE AVERAGE(DATA,AVGG)
      REAL DATA(1),AVGG(1)
      DO 10, K=1,256
           AVG=0.0
           DO 20, I=0,127
                AVG=AVG+DATA(K+I)
   20           CONTINUE
           AVGG(K)=AVG/129
   10      CONTINUE
      DO 30, I=1,256
           DATA(I+64)=DATA(I+64)-AVGG(I)+AVGG(65)
   30      CONTINUE
      RETURN
      END
$SUBTITLE: 'SUBROUTINE PLOTPTDRIFT; PRESS. TRANS. DRIFT DURING RUN'
$PAGE
C
C
      SUBROUTINE PLOTPTDRIFT(X,Y,DATE,CTIME)
C
C
      CHARACTER DATE*14,CTIME*14,RESPONSE*1,XTITLE*13,YTITLE*15,TITLE*35
      REAL X(1),Y(1)
$INCLUDE: 'HGRAPHCONSTS.FOR'
      NOPTS=256
      XTITLE='TIME (SECS)'//CHAR(0)
      YTITLE='PRESSURE (PSI)'//CHAR(0)
      TITLE= 'PRESSURE WAVE MOVING AVERAGE'//CHAR(13)//CHAR(0)
C
      CALL SCALES(Y(1),NOPTS,YMIN,YMAX,YTICS)
      CALL SCALES(X(65),NOPTS,XMIN,XMAX,XTICS)
      CALL INIPLT(0,.FALSE.,1.0)
      CALL SCALE (XMIN,XMAX,YMIN,YMAX)
      CALL SETXLABEL(BELOW,NONE,2,'10.1'//CHAR(0))
      CALL SETYLABEL(LEFT,NONE,2,'10.2'//CHAR(0))
C
      CALL SETXTICS(XMIN,XMAX,XTICS,200,BELOW,2,0)
      CALL SETYTICS(YMIN,YMAX,YTICS,200,LEFT,2,0)
```

```
C
      CALL SETXAXIS(XMIN,YMAX,XMAX,LINEAR)
      CALL SETYAXIS(XMAX,YMIN,YMAX,LINEAR)
C
      CALL DRAWAXIS(CHAR(0),2,CENTER,BELOW,CHAR(0),2,LEFT,CENTER)
C
      CALL SETXLABEL(BELOW,MAJOR,2,'10.1'//CHAR(0))
      CALL SETYLABEL(LEFT,MAJOR,2,'10.2'//CHAR(0))
C
      CALL SETXTICS(XMIN,XMAX,XTICS,200,ABOVE,2,0)
      CALL SETYTICS(YMIN,YMAX,YTICS,200,RIGHT,2,0)
C
      CALL SETXAXIS(XMIN,YMIN,XMAX,LINEAR)
      CALL SETYAXIS(XMIN,YMIN,YMAX,LINEAR)
C
      CALL DRAWAXIS(XTITLE,2,CENTER,BELOW,YTITLE,2,LEFT,CENTER)
      CALL POLYLINE(X(65),Y(1),NOPTS,0,9,2,0,0)
      CALL JUSTIFYSTRING (2000,6700,TITLE,0,3,BASE,BASE)
      CALL WRITESTRING (DATE//CTIME//CHAR(0),0,2)
      CALL ENDPLT
C
C
      WRITE (*,'(A\)') ' MAKE A HARD COPY FILE? (Y):'
READ  (*,'(A1)') RESPONSE
IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 100
CALL FILEUPDATE(ICOUNT)
C
C
CALL INIPLT(ICOUNT,.FALSE.,1.0)
CALL SCALE (XMIN,XMAX,YMIN,YMAX)
CALL SETXLABEL(BELOW,NONE,2,'10.1'//CHAR(0))
CALL SETYLABEL(LEFT,NONE,2,'10.2'//CHAR(0))
C
      CALL SETXTICS(XMIN,XMAX,XTICS,200,BELOW,2,0)
      CALL SETYTICS(YMIN,YMAX,YTICS,200,LEFT,2,0)
C
      CALL SETXAXIS(XMIN,YMAX,XMAX,LINEAR)
      CALL SETYAXIS(XMAX,YMIN,YMAX,LINEAR)
C
      CALL DRAWAXIS(CHAR(0),2,CENTER,BELOW,CHAR(0),2,LEFT,CENTER)
C
      CALL SETXLABEL(BELOW,MAJOR,2,'10.1'//CHAR(0))
      CALL SETYLABEL(LEFT,MAJOR,2,'10.2'//CHAR(0))
C
      CALL SETXTICS(XMIN,XMAX,XTICS,200,ABOVE,2,0)
      CALL SETYTICS(YMIN,YMAX,YTICS,200,RIGHT,2,0)
C
      CALL SETXAXIS(XMIN,YMIN,XMAX,LINEAR)
      CALL SETYAXIS(XMIN,YMIN,YMAX,LINEAR)
C
      CALL DRAWAXIS(XTITLE,2,CENTER,BELOW,YTITLE,2,LEFT,CENTER)
      CALL POLYLINE(X(65),Y(1),NOPTS,0,9,2,0,0)
      CALL JUSTIFYSTRING (2000,6700,TITLE,0,3,BASE,BASE)
      CALL WRITESTRING (DATE//CTIME//CHAR(0),0,2)
      CALL ENDPLT
  100     CONTINUE
      RETURN
      END
C
C
$SUBTITLE:'SUBROUTINE FILEUPDATE; FIND UNUSED PLOT FILE NO.'
$PAGE
      SUBROUTINE FILEUPDATE(ICOUNT)
      CHARACTER CNP*2
      LOGICAL USED
      IF (ICOUNT.EQ.0) ICOUNT=10
  567     CONTINUE
      WRITE (CNP,'(I2)') ICOUNT
      INQUIRE(FILE='PLOT'//CNP//'.DAT',EXIST=USED)
      IF (.NOT.USED) THEN
          CONTINUE
      ELSE
          ICOUNT=ICOUNT+1
          GOTO 567
      END IF
      RETURN
      END
```

D.7 DRIFTH.FOR

```
$TITLE: 'DRIFTH.FOR'
$SUBTITLE:'MAIN PROGRAM'
**************PROGRAM DRIFT*********************
$NOTRUNCATE
$INCLUDE: 'HGRAPHALIAS.FOR'
      IMPLICIT REAL (A-H,L-Z)
C
      REAL A1,A2,PRESS,TEMP,PERIOD,STROKE,OFFSET,LVSP,
     1 LVYIN,VMAG(0:6),INCHES,PPLOTDAT(384),AVGG(256),TEMPP(23,7)
C
      CHARACTER DATE*14,CTIME*14,RESPONSE*1,VS1*10,VS2*10,
     1 TS1*10,TS2*10,TS3*10,TS4*10,CREYNOLD*14,JUNK*14
C
      CHARACTER FNAME*30,CMASS*11,CVR*7,CPER*9,TITLE*90
C
      DIMENSION CECCAL1(40),CECCAL2(40),DYTCAL1(40),DYTCAL2(40),
     1 CECSYNC1(80),CECSYNC2(30),DYTSYNC1(80),DYTSYNC2(30),
     2 CAL1TIME(40),CAL2TIME(40),SYNC1TIME(10),SYNC2TIME(30),YCALC(400)
C
      DIMENSION PCALC(40),PCALC2(40),PCALC3(40),CAL3TIME(80),
     1 VOLDAT(385),PRESDAT(384),TIME(384),PMAG(0:6)
     2  ,PPHASE(6),X(10),Y(10)
C
      EQUIVALENCE (DYTSYNC1(11),DYTSYNC2(1)),(CECSYNC1(11),CECSYNC2(1))
     1 ,(DYTSYNC1(41),DYTCAL2(1)),(CECSYNC1(41),CECCAL2(1))
C
      PARAMETER (PI=3.14159265)
C
C     ***************** OPEN FILE TO BE PROCESSED ******************
 4011      WRITE(*,*) 'CURRENT FILE:',FNAME
      WRITE(*,'(A\)') ' ENTER FILE NAME:'
                           READ (*,'(A30)') FNAME
                           OPEN (UNIT=1,FILE=FNAME,STATUS='OLD')
                           REWIND 1
C
C
C      ************* READ IN PARAMETERS OF DATA RUN *****************
                           READ (1,1000)DATE,CTIME,A1,A2,PRESS,TEMP,VOLUME
                           READ (1,'(7F14.0)') PERIOD,STROKE,OFFSET,LVSP,LVYIN,LVSP,LVYIN
                           READ (1,'(4F14.0)')(X(I),Y(I),X(I),Y(I), I=1,10)
C
C      READ IN FIRST PRESSURE CALIBRATION DATA SET
      DO 10, I=1,40
           READ (1,1010)CECCAL1(I),DYTCAL1(I)
   10  CONTINUE
      READ (1,1020)ST1TIME1,ST2TIME1,ST3TIME1,ST4TIME1,ENDTIME1
C
C      READ IN SECOND PRESSURE CALIBRATION DATA SET
      DO 20, I=1,40
           READ (1,1010) CECCAL2(I),DYTCAL2(I)
   20  CONTINUE
      READ (1,1030)ST1TIME2,ST2TIME2,ST3TIME2,ST4TIME2,STDATTIME,
     1 ENDDATTIME,ST1SYNC,END1SYNC,ST2SYNC,END2SYNC
 1030  FORMAT(10F14.0)
C
C     READ IN FIRST PRESSURE SYNCHRONIZATION DATA SET
      DO 30, I=1,10
           READ (1,1010)CECSYNC1(I),DYTSYNC1(I)
   30  CONTINUE
C
C      READ IN SECOND PRESSURE SYNCHRONIZATION DATA SET
      DO 40, I=1,30
           READ (1,1010)CECSYNC2(I),DYTSYNC2(I)
   40  CONTINUE
C
C      READ IN P-V DATA
      DO 50, I=1,384
           READ (1,1010) VOLDAT(I),PRESDAT(I)
C     WRITE(*,*) VOLDAT(I),PRESDAT(I)
   50  CONTINUE
C
C
C     READ IN TEMPERATURE DATA
C
      DO 60, I=1,23
           READ (1,'(7F14.0)') (TEMPP(I,J), J=1,7)
```

```
            TEMPP(I,1) = GERM1005(TEMPP(I,1))
            TEMPP(I,2) = GERM233 (TEMPP(I,2))
            T1005      = T1005 + TEMPP(I,1)
            T233       = T233  + TEMPP(I,2)
   60       CONTINUE
            T1005 = T1005/23.
            T233  = T233 /23.
C      ****************************************************************
       WRITE (*,'(A\)') 'ENTER PRESSURE OFFSET (PSI):'
                 READ  (*,'(F14.0)') PRESSOFF
                 PRESSOFF = 14.7 + PRESSOFF
C
C
                 AVGCSYNC1=0
                 AVGCSYNC2=0
C
C      ***************** CREATE TIME DATA ARRAYS *********************
                 DO 100, I=1,10
                 CAL1TIME(I)    =(I-1)*0.5
                 CAL1TIME(I+10)=(I+9)*0.5
                 CAL1TIME(I+20)=(I+19)*0.5
                 CAL1TIME(I+30)=(I+29)*0.5
C
            CAL2TIME(I)    =(I-1)*0.5
            CAL2TIME(I+10)=(I-1)*0.5 + ST2TIME-ST1TIME
            CAL2TIME(I+20)=(I-1)*0.5 + ST3TIME-ST1TIME
            CAL2TIME(I+30)=(I-1)*0.5 + ST4TIME-ST1TIME
C
            CAL3TIME(I)    =(I-1)*0.5
            CAL3TIME(I+40)=(I-1)*0.5 + ST1TIME2-ST1SYNC
            CAL3TIME(I+50)=(I-1)*0.5 + ST2TIME2-ST1SYNC
            CAL3TIME(I+60)=(I-1)*0.5 + ST3TIME2-ST1SYNC
            CAL3TIME(I+70)=(I-1)*0.5 + ST4TIME2-ST1SYNC
C
       SYNC1TIME(I)=(I-1)*0.5
C       CONVERT RAW CEC SYNCH. OUTPUT TO PRESSURES AND AVERAGE *****
       CECSYNC1(I) =(CECSYNC1(I)-A2)/A1+PRESSOFF
       AVGSYNC1=AVGSYNC1+CECSYNC1(I)
C
  100   CONTINUE
       AVGSYNC1=AVGSYNC1/10
       DO 110, I=1,30
            SYNC2TIME(I)=(I-1)*0.5
            CAL3TIME(I+10)=(I-1)*0.5 + ST2SYNC-ST1SYNC
C       CONVERT RAW CEC SYNCH. OUTPUT TO PRESSURES AND AVERAGE *****
            CECSYNC2(I) =(CECSYNC2(I)-A2)/A1+PRESSOFF
            AVGSYNC2=AVGSYNC2+CECSYNC2(I)
  110   CONTINUE
       AVGSYNC2=AVGSYNC2/30
C
C
C       CONVERT RAW CEC CALIBRATION OUTPUT TO PRESSURES **************
       DO 120, I=1,40
            CECCAL1(I)=(CECCAL1(I)-A2)/A1+PRESSOFF
            CECCAL2(I)=(CECCAL2(I)-A2)/A1+PRESSOFF
  120   CONTINUE
C
C       FIND BEST FIT FOR DYTRAN OUTPUT DATA TO CEC OUTPUT DATA
C       ********* FIRST PRESSURE CALIBRATION *************
C       USING LEAST SQUARES FIT IN BOTH PRESSURE AND TIME TO COMPENSATE
C       FOR TIME DRIFT.
C
       CALL LSSQ3D(CECCAL1,DYTCAL1,CAL2TIME,40,A10,A11,A12)
C       STORE PRESSURE AND TIME SENSITIVITIES AS STRINGS FOR GRAPHICS
       WRITE (VS1,5000)A11
       WRITE (TS1,5000)A12
C       USE CURVE FIT PARAMS TO CONVERT DYTRAN OUTPUT TO PRESSURE
       DO 121, I=1,40
            PCALC(I)=A10+A11*DYTCAL1(I)+A12*CAL2TIME(I)
  121   CONTINUE
C
C
C
C      ********************
C
C
C
C      LET A3 EQUAL THE VOLTAGE-PRESSURE SENSITIVITY OF THE FIRST CAL.
       A3=A11
c      WRITE (*,'(A)') ' ENTER P.T. SENSITIVITY:'
c                      READ (*,'(F14.0)') A3
```

```
C
C
C                    LET OFFSET A4 EQUAL)') 'ENTER P.T. OFFSET:'
c     READ (*,'(F14.0)') A4
C
C
C**********************************************************************
C
C
C
C
C     ***** CALCULATE PRESSURE TIME SLOPES FOR CECSYNC RUNS **********
      CALL LSSQLN(SYNC2TIME,CECSYNC2,30,CTSLP2,YINTRC,YCALC,STDERR)
      CALL LSSQLN(SYNC1TIME,CECSYNC1,10,CTSLP1,YINTRC,YCALC,STDERR)
C
C     **** CALCULATE PRESSURE TIME SLOPES FOR DYTSYNC RUNS USING
C          PRESSURE SLOPES FROM FIRST CALIBRATION ******************
      DO 123, I=1,40
           DYTSYNC1(I)=DYTSYNC1(I)*A1
  123      CONTINUE
      CALL LSSQLN(SYNC2TIME,DYTSYNC2,30,DTSLP2,YINTRC,YCALC,STDERR)
      CALL LSSQLN(SYNC1TIME,DYTSYNC1,10,DTSLP1,YINTRC,YCALC,STDERR)
C**********************************************************************
C
C
C
C
C
CC
C
C
C**********************************************************************
C     CORRECT 2ND P CALIBRATION FOR TIME DRIFT BASED ON 2ND SYNCH. RUN
C     DO 200, I=1,40
C          DYTCAL2(I)=DYTCAL2(I)-TSLOPE3 * CAL3TIME(I+40)
C  200  CONTINUE
C     CORRECT 2ND P CALIBRATION FOR TIME DRIFT BASED ON 2ND SYNCH. RUN
C**********************************************************************
C
C
C
C
C**********************************************************************
C     ************** SECOND PRESSURE CALIBRATION *****************
C     FIND BEST FIT FOR DYTRAN OUTPUT DATA TO CEC OUTPUT DATA
C     USING LEAST SQUARES FIT IN BOTH PRESSURE AND TIME
      CALL LSSQ3D(CECCAL2,DYTCAL2,CAL2TIME,40,A0,A21,A22)
C**********************************************************************
C
C
C
C     RECORD VARIOUS SENSITIVITIES AS STRINGS FOR GRAPHICS
      TSLP2=CTSLP1-DTSLP1
      TSLP4=CTSLP2-DTSLP2
      WRITE (VS2,5000)A21
      WRITE (TS2,5000)TSLP2
      WRITE (TS3,5000)A22
      WRITE (TS4,5000)TSLP4
C
C     CONVERT SECOND DYTRAN CALIBRATION AND BOTH SYNCH. RUNS
C     TO PRESSURE USING FIRST CAL PRESS VS. VOLT SENS. AND
C     ALSO ORIGINAL OFFSET.
      DO 122, I=1,40
           PCALC2(I)=A0+A21*DYTCAL2(I)
           PCALC3(I)=A0+A21*DYTSYNC1(I)+A2*CAL3TIME(I)
  122  CONTINUE
      CAL1MAX=(AINT(CAL1TIME(40)/5)+1.0)*5
C
C
C
C      *************** CALCULATE PERTINENT PV DATA ******************
C     ***************************************************************
C     NOTE: PRESDAT IS CONVERTED USING THE FOLLOWING:
C          PRESS VS. VOLT. SENS. FROM FIRST CAL. RUN  (A1)
C          OFFSET AS AVERAGE OF THE LAST OF THE FIRST SYNCH. AND
C          THE FIRST OF THE SECOND SYNCH.             (A3)
C
C
C
      VMAX=0.0
      VMIN=100
```

```
      PREF = -100
      CLVOL=1.12
      DO 210, I=1,384
           TIME(I)=(I-1)*PERIOD/128
           INCHES=0.0
           DO 211, J=1,10
                INCHES=INCHES +
     1               X(J)*2.0/33.0*COS(J*PI*VOLDAT(I)/4096.0)+
     2               Y(J)*2.0/33.0*SIN(J*PI*VOLDAT(I)/4096.0)
  211      CONTINUE
           INCHES=INCHES+VOLDAT(I)*LVSP+LVYIN
           VOLDAT(I)=clvol+(PI*(1.469**2))/4*INCHES
           VMAX=AMAX1(VOLDAT(I),VMAX)
           VMIN=AMIN1(VOLDAT(I),VMIN)
           PRESDAT(I)=A4+A3*PRESDAT(I)
C          PREF = AMAX1(PRESDAT(I),PREF)
  210  CONTINUE
C     ***** TIME DRIFT IS COMPENSATED FOR BY RUNNING AVERAGE ******
C
C
C
      CALL AVERAGE(PRESDAT(1),AVGG(1))
C
C
C     ***************** THE ABOVE ROUTINE ALTERS PRESDAT ************
      VOLDAT(385)=0.0
      VREF = (VMIN+VMAX)/2.
      VOLRAT = VMAX/VMIN
C
C
      VOLD=VOLDAT(129)
      DO 220, I=130,256
           VNEW=VOLDAT(I)
           IF (VNEW.GT.VOLDAT(I-5).AND.VNEW.LT.VOLDAT(I+5)) THEN
           IF (VNEW.GT.VOLD.AND.VNEW.GT.VREF.AND.VOLD.LT.VREF) THEN
              GOTO 221
           END IF
           END IF
           VOLD=VNEW
  220      CONTINUE
      WRITE (*,*) 'ERROR FINDING VREF'
      STOP
  221 CONTINUE
      PREF= 0
      DO 222, J=I-10,I+10
           PREF=PREF+PRESDAT(J)
  222      CONTINUE
      PREF=PREF/21.
C
C
C     ****************************************************************
      WRITE (*,'(A\)') ' ENTER MASS OF HELIUM IN CYLINDER:'
                       READ  (*,'(F14.0)') MASS
C                      ****************************************************************
C                      MASS= (PRESS*(CLVOL+(PI*(1.469**2))/4*
C     1                (VOLUME*LVSP+LVYIN)))/(18.381*TEMP)
C                      ***************** RHO BELOW IS IN MOLES / LITER ***************
                       RHO =DENSITY(VREF,MASS)
C     ************* SIRHO BELOW IS IN KG/M3 **************************
      SIRHO = RHO * 4.0026
      TREF=TFDENP(PREF/14.7,RHO)
      SREF=ENTROP(RHO,TREF)
      CALL DERIV(RJ1,RJ2,RJ3,RJ4,CSUBP,RJ5,PREF/14.7*101325,SIRHO,TREF)
      TCON=THCON(SIRHO,TREF)
      MIKIC = (TREF**1.93 + SQRT(SIRHO*CSUBP*TCON))/(TREF**1.93)
C
C     FIND DRIFT IN P-V DATA
C
      CALL SCALES (VOLDAT(1),385,VMIN,VMAX,VTICS)
      CALL SCALES (PRESDAT(1),384,PMIN,PMAX,PTICS)
      CALL SCALES (TIME(1),384,TMIN,TMAX,TTICS)
      PER3=3.0*PERIOD
C
C
C     ****************************************************************
C
C     CALCULATE REYNOLDS & PECLET NOS. (USE REAL PROPERTIES)
C
      CALL DERIV (TRASH1,TRASH2,TRASH3,TRASH4,CP,TRASH5,PREF,RHO,TREF)
      VISCOSITY = VISC(RHO,TREF)
      THERMCOND =THCON(RHO,TREF)
```

```
      PRANDTL   =VISCOSITY*CP/THERMCOND
      VISCOSITY =VISCOSITY * 1000. / 39.37
      REYNOLD=(2*MASS/(VMAX+VMIN))*(2*STROKE/PERIOD)*1.460/VISCOSITY
      PEC=REYNOLD*PRANDTL
      WRITE (CREYNOLD,'(F14.2)')REYNOLD
      WRITE(CMASS,'(F10.8)')MASS
      WRITE(CVR,'(F6.3)')VOLRAT
      WRITE(CPER,'(F9.2)')PERIOD
C
C
C      ********** DECOMPOSE PRESSURE WAVE **************
C
      CALL DECOMP(VOLDAT(1),PRESDAT(1),TREF,
     1 SREF,PMAG(1),PPHASE(1),VMAG(1),PERIOD,
     2 PPLOTDAT(1),MASS)
C
C
C
C**********************************************************************
C
C
      CALL SCALES(PPHASE(1),3,PPHASEMIN,PPHASEMAX,PPHASETICS)
      CALL SCALES(PMAG(1),4,PMAGMIN,PMAGMAX,PMAGTICS)
      CALL SCALES(VMAG(1),6,VMAGMIN,VMAGMAX,VMAGTICS)
      CALL SCALES(CECCAL1(1),40,CEC1MIN,CEC1MAX,YTICS1)
      CALL SCALES(DYTCAL1(1),40,DYT1MIN,DYT1MAX,YTICS2)
      CALL SCALES(CECCAL2(1),40,CEC2MIN,CEC2MAX,YTICS3)
      CALL SCALES(DYTCAL2(1),40,DYT2MIN,DYT2MAX,YTICS4)
      CALL SCALES(CECSYNC1(1),40,CSYNCMIN,CSYNCMAX,CSYNTICS)
C     CALL SCALES(PCALC3(1),40,CSYNCMIN,CSYNCMAX,CSYNTICS)
      CALL SCALES(DYTSYNC1(1),40,DSYNCMIN,DSYNCMAX,DSYNTICS)
C     CALL SCALES(YCALC(1),35,CORRMIN,CORRMAX,CORRTICS)
C**********************************************************************
C
C
C
 4000      CONTINUE
      WRITE (*,'(A)') '1'
      WRITE (*,'(A)') '1'
      WRITE (*,'(A)') '1'
      WRITE (*,'(A)') '1'
      WRITE (*,*) ' CURRENT FILE:',FNAME
      WRITE (*,'(A)') '1'
      WRITE (*,*) ' ************************************************'
      WRITE (*,*) ' 1.  DISPLAY PRESSURE TRANSDUCER SENSITIVITIES'
      WRITE (*,*) ' 2.  WRITE TO CORE.DAT'
      WRITE (*,*) ' 3.  DISPLAY PLOT P-V'
      WRITE (*,*) ' 4.  DISPLAY PLOT PRESSURE MAGNITUDE SPECTRUM'
      WRITE (*,*) ' 5.  DISPLAY PLOT PRESSURE PHASE SPECTRUM'
      WRITE (*,*) ' 6.  DISPLAY PLOT PRESSURE CALIBRATION #1'
      WRITE (*,*) ' 7.  DISPLAY PLOT PRESSURE CALIBRATION #2'
      WRITE (*,*) ' 8.  DISPLAY PLOT PRESSURE SYNCHRONIZATION'
      WRITE (*,*) ' 9.  DISPLAY PLOT PRESSURE WAVES'
      WRITE (*,*) ' 10. DISPLAY PLOT PRESSURE TRANSDUCER DRIFT'
      WRITE (*,*) ' 11. PROCESS NEW FILE'
      WRITE (*,*) ' 12. EXIT PROGRAM'
      WRITE (*,*) ' ************************************************'
      WRITE (*,'(A)') '1'
      WRITE (*,'(A\)')' ENTER NUMBER OF DESIRED ACTION FROM ABOVE LIST:'
READ (*,'(I2)') IAC
GOTO (4001,4002,4003,4004,4005,4006,4007,4008,4009,4010,4011,4012)
     1     IAC
GOTO 4000
C
C
C
C**********************************************************************
C
C
CPRINT OUT VARIOUS PRESSURE TRANSDUCER SENSITIVITIES
C
 4001      CONTINUE
      WRITE (*,*) ' DATA FILE:',FNAME
      WRITE (*,*) ' ',DATE,'     ',CTIME
      WRITE (*,*) ' MASS (GRAMS):.............................',CMASS
      WRITE (*,*) ' VOLUME RATIO:.............................',CVR
      WRITE (*,*) ' PERIOD (SECS):............................',CPER
      WRITE (*,*) ' TEMPERATURE RANGE:..................... LHE'
      WRITE (*,*) ' REYNOLDS NO.:.............................',
     1     CREYNOLD
      WRITE (*,3) ' PECLET NO.:...............................',PEC
```

```
       WRITE (*,3) ' FIRST HARMONIC PRESSURE MAGNITUDE:.....',
      1     PMAG(1)
       WRITE (*,3) ' SECOND HARMONIC PRESSURE MAGNITUDE:.....',
      1     PMAG(2)
       WRITE (*,3) ' FIRST HARMONIC PRESSURE PHASE:.....',
      1     PPHASE(1)
       WRITE (*,3) ' SECOND HARMONIC PRESSURE PHASE:.....',
      1     PPHASE(2)
       WRITE (*,*) ' *********** FIRST PRESSURE CALIBRATION *******'
       WRITE (*,3) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A11
       WRITE (*,3) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A12
       WRITE (*,*) ' *********** SECOND PRESSURE CALIBRATION ******'
       WRITE (*,3) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A21
       WRITE (*,3) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A22
       WRITE (*,*) ' *********** FIRST SYNCHRONIZATION ************'
       WRITE (*,3) ' CEC PRESSURE TIME SLOPE       (PSI/SEC):.....',
      1     CTSLP1
       WRITE (*,3) ' DYTRAN PRESSURE TIME SLOPE    (PSI/SEC):.....',
      1     DTSLP1
       WRITE (*,*) ' *********** SECOND SYNCHRONIZATION ***********'
       WRITE (*,3) ' CEC PRESSURE TIME SLOPE       (PSI/SEC):     ',
      1     CTSLP2
       WRITE (*,3) ' DYTRAN PRESSURE TIME SLOPE    (PSI/SEC):.....',
      1     DTSLP2
       WRITE (*,3) ' 1ST CYCLE START PRESSURE:.....',
      1     PRESDAT(1)
       WRITE (*,3) ' 1ST CYCLE END PRESSURE:.....',
      1     PRESDAT(128)
       WRITE (*,3) ' 2ND CYCLE START PRESSURE:.....',
      1     PRESDAT(129)
       WRITE (*,3) ' 2ND CYCLE END PRESSURE:.....',
      1     PRESDAT(256)
       WRITE (*,3) ' 3RD CYCLE START PRESSURE:.....',
      1     PRESDAT(257)
       WRITE (*,3) ' 3RD CYCLE END PRESSURE:.....',
      1     PRESDAT(384)
       WRITE (*,*)
       WRITE (*,3) ' REFERENCE PRESSURE (PSIA):.....',
      1     PREF
       WRITE (*,3) ' REFERENCE VOLUME (CU.IN.):.....',
      1     VREF
       WRITE (*,3) ' REFERENCE TEMPERATURE (KELVIN):.....',
      1     TREF
C
C
  3   FORMAT (A,F14.4)
C
      WRITE (*,'(A\)') ' ENTER Y FOR PRINTOUT:'
READ  (*,'(A1)') RESPONSE
IF (RESPONSE.NE.'Y'.AND.RESPONSE.NE.'y') GOTO 4000
C
C
OPEN (UNIT=9,FILE='PRN')
WRITE (9,*) ' DATA FILE:',FNAME
WRITE (9,*) ' ',DATE,'    ',CTIME
WRITE (9,'(A)') '0'
WRITE (9,'(A)') '0'
WRITE (9,*) ' MASS (GRAMS):.....',CMASS
      WRITE (9,*) ' VOLUME RATIO:.....',CVR
      WRITE (9,*) ' PERIOD (SECS):.....',CPER
      WRITE (9,*) ' TEMPERATURE RANGE:..... LN2'
      WRITE (9,*) ' REYNOLDS NO.:.....',
     1     CREYNOLD
      WRITE (9,3) ' PECLET NO.:.....',PEC
      WRITE (9,*) ' VISCOSITY (g/in-sec):.....',
     1     VISCOSITY
      WRITE (9,3) ' PRANDTL NO.:.....',
     1     PRANDTL
      WRITE (9,3) ' FIRST HARMONIC PRESSURE MAGNITUDE:.....',
     1     PMAG(1)
      WRITE (9,3) ' SECOND HARMONIC PRESSURE MAGNITUDE:.....',
     1     PMAG(2)
      WRITE (9,3) ' FIRST HARMONIC PRESSURE PHASE:.....',
     1     PPHASE(1)
      WRITE (9,3) ' SECOND HARMONIC PRESSURE PHASE:.....',
     1     PPHASE(2)
      WRITE (9,'(A)') '0'
      WRITE (9,'(A)') '0'
      WRITE (9,*) ' *********** FIRST PRESSURE CALIBRATION *******'
      WRITE (9,3) ' PRESSURE VOLTAGE SLOPE        (PSI/LSB):.....',A11
      WRITE (9,3) ' PRESSURE TIME SLOPE           (PSI/SEC):.....',A12
```

```
      WRITE (9,*) ' ************ SECOND PRESSURE CALIBRATION ******'
      WRITE (9,3) ' PRESSURE VOLTAGE SLOPE          (PSI/LSB):.....',A21
      WRITE (9,3) ' PRESSURE TIME SLOPE             (PSI/SEC):.....',A22
      WRITE (9,*) ' ************ FIRST SYNCHRONIZATION ************'
      WRITE (9,3)' CEC PRESSURE TIME SLOPE        (PSI/SEC):.....',CTSLP1
      WRITE (9,3)' DYTRAN PRESSURE TIME SLOPE     (PSI/SEC):.....',DTSLP1
      WRITE (9,*)' ************ SECOND SYNCHRONIZATION ************'
      WRITE (9,3)' CEC PRESSURE TIME SLOPE        (PSI/SEC):.....',CTSLP2
      WRITE (9,3)' DYTRAN PRESSURE TIME SLOPE     (PSI/SEC):.....',DTSLP2
      WRITE (9,'(A)') '0'
      WRITE (9,'(A)') '0'
      WRITE (9,3) ' 1ST CYCLE START PRESSURE:.................',
     1     PRESDAT(1)
      WRITE (9,3) ' 1ST CYCLE END PRESSURE:...................',
     1     PRESDAT(128)
      WRITE (9,3) ' 2ND CYCLE START PRESSURE:.................',
     1     PRESDAT(129)
      WRITE (9,3) ' 2ND CYCLE END PRESSURE:...................',
     1     PRESDAT(256)
      WRITE (9,3) ' 3RD CYCLE START PRESSURE:.................',
     1     PRESDAT(257)
      WRITE (9,3) ' 3RD CYCLE END PRESSURE:...................',
     1     PRESDAT(384)
      WRITE (9,*)
      WRITE (9,3) ' REFERENCE PRESSURE (PSIA):................',
     1     PREF
      WRITE (9,3) ' REFERENCE VOLUME (CU.IN.):................',
     1     VREF
      WRITE (9,3) ' REFERENCE TEMPERATURE (KELVIN):...........',
     1     TREF
      WRITE (9,3) ' AVERAGE TEMPERATURE R#233 :...............',
     1     T233
      WRITE (9,3) ' AVERAGE TEMPERATURE R#1005 :..............',
     1     T1005
      WRITE (9,'(A)') '1'
C
      GOTO 4000
C
C
C
C
C*********************************************************************
C
C      ************ WRITE DATA TO FILE ***************
C
C     OPEN FILE FOR STORAGE OF PARAMETERS AND RESULTS
 4002      CONTINUE
      OPEN (UNIT=2,FILE='MCORE.DAT',STATUS='OLD')
C
C      MOVE TO END OF FILE AND APPEND NEW DATA THERE
    8  READ (2,'(A14)',END=9)JUNK
      GOTO 8
    9  CONTINUE
      BACKSPACE 2
C
      WRITE(2,3000)FNAME,PERIOD,MASS,VOLRAT,REYNOLD,PEC,RRCKH,RRCKS,
     1 VMAG(0),VMAG(1),PMAG(0),PMAG(5),PMAG(6),PMAG(3),PMAG(1),PMAG(2),
     2 PPHASE(1),PPHASE(2),PPHASE(3)
 3000      FORMAT(A14,18F14.4)
      CLOSE (UNIT=2)
      GOTO 4000
C
C
C*********************************************************************
C
C
C
 4003      CONTINUE
      CALL PLOTPV
     1 (PER3,VMIN,VMAX,VTICS,PERIOD,TIME(1),VOLDAT(1),PMIN,PMAX,PTICS,
     2 PRESDAT(1),CMASS,CVR,CPER,CREYNOLD,DATE,CTIME)
      GOTO 4000
C
C
C      CALL PLOTVMSPEC (VMAGMIN,VMAGMAX,VMAGTICS,VMAG(1),DATE,CTIME)
C
C
 4004      CONTINUE
      CALL PLOTMSPEC (PMAGMIN,PMAGMAX,PMAGTICS,PMAG(1),DATE,CTIME)
      GOTO 4000
C
```

```
C
 4005      CONTINUE
      CALL PLOTPSPEC
     1 (PPHASEMIN,PPHASEMAX,PPHASETICS,PPHASE(1),DATE,CTIME)
      GOTO 4000
C
C
 4006      CONTINUE
      CALL PLOTPCAL1
     1 (DYT1MIN,DYT1MAX,YTICS2,DYTCAL1(1),CEC1MIN,CEC1MAX,YTICS1,
     2 CECCAL1(1),PCALC(1),CAL1MAX,CAL1TIME(1),DATE,CTIME,TS1,VS1)
      GOTO 4000
C
C
 4007      CONTINUE
      CALL PLOTPCAL2
     1 (DYT2MIN,DYT2MAX,YTICS4,DYTCAL2(1),CEC2MIN,CEC2MAX,YTICS3,
     2 CECCAL2(1),PCALC2(1),CAL1MAX,CAL1TIME(1),DATE,CTIME,TS3,VS2)
      GOTO 4000
C
C
 4008      CONTINUE
      CALL PLOTPSYNC
     1 (DSYNCMIN,DSYNCMAX,DSYNTICS,SYNC1TIME(1),DYTSYNC1(1),
     2 SYNC2TIME(1),DYTSYNC2(1),YCALC(1),CSYNCMIN,CSYNCMAX,
     3 CSYNTICS,CECSYNC1(1),CECSYNC2(1),DATE,CTIME,TS2,TS4)
      GOTO 4000
C
C
 4009 CONTINUE
      OPEN (UNIT=12,FILE='PDAT.DAT')
      OPEN (UNIT=13,FILE='PISO.DAT')
      OPEN (UNIT=14,FILE='PADI.DAT')
      DO 711, I=1,128
           WRITE(12,'(2F14.4)')TIME(I),PPLOTDAT(I)
           WRITE(13,'(2F14.4)')TIME(I),PPLOTDAT(I+128)
           WRITE(14,'(2F14.4)')TIME(I),PPLOTDAT(I+256)
  711      CONTINUE
      CLOSE (12)
      CLOSE (13)
      CLOSE (14)
      CALL PDATAPLOT (PERIOD,TIME(1),PPLOTDAT(1),DATE,CTIME)
      GOTO 4000
C
C
 4010      CONTINUE
      CALL PLOTPTDRIFT (TIME(1),AVGG(1),DATE,CTIME)
      GOTO 4000
 1000  FORMAT(2A14,9F14.0)
 1010  FORMAT(2F14.0)
 1020 FORMAT(5F14.0)
 2000  FORMAT(' MAKE A HARDCOPY FILE ? (Y,N)',\)
 5000  FORMAT(F10.4)
 4012CONTINUE
END
C
C
C
$SUBTITLE: 'FUNCTION GERM233(R); CONVERTS VOLTAGE TEMP. '
$PAGE
C
C
      FUNCTION GERM233(R)
C
C     THIS FUNCTION CONVERTS THE VOLTAGE OUTPUT OF GERMANIUM RESISTOR
C     SERIAL #233 WHEN EXCITED BY 10 MICROAMPS TO TEMPERATURE IN KELVIN
C
C     THIS FUNCTION USES AN EMPERICAL CURVE FIT WHICH DEPENDS ONLY ON
C     THE RESISTANCE AT LN2 AND LHE TEMPERATURES
C
      REAL R,A,B
      A = -0.3742035
      B = 1.661581
      R = ABS(R)*1.E5
C
      GERM233 = 10**(((ALOG10(R))**(1./3.)-B)/A)
      RETURN
      END
```

```
$SUBTITLE: 'FUNCTION GERM1005(R); CONVERTS VOLTAGE TEMP. '
$PAGE
C
C
      FUNCTION GERM1005(R)
C
CC    THIS FUNCTION CONVERTS THE VOLTAGE OUTPUT OF GERMANIUM RESISTOR
C     SERIAL #1005 WHEN EXCITED BY 10 MICROAMPS TO TEMPERATURE IN KELVIN
C
C
C     THIS FUNCTION USES AN EMPERICAL CURVE FIT WHICH DEPENDS ONLY ON
C     THE RESISTANCE AT LN2 AND LHE TEMPERATURES
C
C
      REAL R,A,B
      A = -0.3747884
      B = 1.674015
      R = ABS(R)*1.E5
C
      GERM1005 = 10**(((ALOG10(R))**(1./3.)-B)/A)
      RETURN
      END
$SUBTITLE:'SUBROUTINE DECOMP; DECOMPOSE PRESS. & VOL. INTO HARMONICS'
$PAGE
C
C
      SUBROUTINE DECOMP
     1 (VOLDAT,PRESDAT,TISOTH,S,PMAG,PPHASE,VMAG,PERIOD,
     2 PPLOTDAT,MASS)
C
C      ***** SUBROUTINE TO DECOMPOSE & NORMALIZE PRESSURE ******
C
C
      CHARACTER DATE*14,CTIME*14
      REAL MASS
C
      REAL VOLDAT(1),PRESDAT(1),PADIABATIC(384),PMAG(6),PPHASE(6)
     1 ,PTRANSFORM(384),PISOTHERM(384),KK,IM,RE,VTRANSFORM(128)
     2 ,ISOMAG,ISORE,ISOIM,VMAG(6),TIME(1),STROKE,OFFSET,PPLOTDAT(384)
     3 ,ADIABAT(384),AVGG(384),APMAG(4)
C
C
C
C      DEFINE PARAMETER kk =  RATIO OF SPECIFIC HEATS
C      THIS PARAMETER MUST BE CHANGED TO MATCH TEMPERATURE CONDITIONS
C
      PARAMETER (KK=1.667, PI=3.14159265)
C
C      CALCULATE ADIABATIC PRESSURE
      DO 5, I=1,384
           RHO            = DENSITY(VOLDAT(I),MASS)
           PISOTHERM(I)   = 14.7 * DTPRES(RHO,TISOTH)
           PADIABATIC(I)  = 14.7 * PFDENS(RHO,S)
    5      CONTINUE
      CALL AVERAGE(PISOTHERM(1),AVGG(1))
      CALL AVERAGE(PADIABATIC(1),AVGG(1))
C
      DO 10, I=1,128
           PTRANSFORM(I)      = PRESDAT(I+128)
           PTRANSFORM(I+128)  = PISOTHERM(I+128)
           PTRANSFORM(I+256)  = PADIABATIC(I+128)
           VTRANSFORM(I)      = VOLDAT(I+128)
           PISOTHERM(I)       = PISOTHERM(I+128)
           PADIABATIC(I)      = PADIABATIC(I+128)
   10   CONTINUE
C
C
      DO 30, I=1,384
      PPLOTDAT(I) = PTRANSFORM(I)
   30      CONTINUE
C
C      PERFORM FFT
C
      CALL FFT(PTRANSFORM(1),PADIABATIC(1),128,7)
      CALL FFT(PISOTHERM(1), VTRANSFORM(1), 128,7)
C
      DO 20, I=1,4
           K = 129-I
           J = 1+I
```

```
            ACRE  = (PTRANSFORM(K) + PTRANSFORM(J))
            ISORE = (PISOTHERM(K)  + PISOTHERM(J))
            ADRE  = (PADIABATIC(K) + PADIABATIC(J))
            VORE  = (VTRANSFORM(K) + VTRANSFORM(J))
            ISOIM = (VTRANSFORM(K) - VTRANSFORM(J))
            ACIM  = (PADIABATIC(K) - PADIABATIC(J))
            ADIM  = (PTRANSFORM(J) - PTRANSFORM(K))
            VOIM  = (PISOTHERM(J)  - PISOTHERM(K))
            ACMAG   = ((ACRE**2)+(ACIM**2))**0.5
            ACPHASE = ATAN2(ACIM,ACRE)
            ADMAG   = ((ADRE**2)+(ADIM**2))**0.5
            ADPHASE = ATAN2(ADIM,ADRE)
            ISOMAG  = ((ISORE**2)+(ISOIM**2))**0.5
            PMAG(I) = (ACMAG-ISOMAG)/(ADMAG-ISOMAG)
            APMAG(I)= ACMAG
            VMAG(I) = (((VORE**2)+(VOIM**2))**0.5)/128
            PPHASE(I)=(ADPHASE-ACPHASE)*360/(2*PI*I)-1.40625-0.03/PERIOD
   20   CONTINUE
        PMAG(0)= PTRANSFORM(1)/128
        PMAG(5)= APMAG(1)/128
        PMAG(6)= APMAG(2)/128
        PMAG(3)= APMAG(3)/128
        VMAG(0)= VTRANSFORM(1)/128
C
        RETURN
        END
C
C
C
$TITLE: 'TFDENP.FOR; FUNCTION FOR USE WITH HEPROP.FOR'
$SUBTITLE:' CALCUALTES TEMP. GIVEN DENSITY AND PRESSURE'
C
C
        FUNCTION TFDENP(P,DEN)
C
C
C       I/O UNITS:
C                       TEMP.........KELVIN
C                       DENSITY......MOLES/LITER
C                       PRESSURE.....ATMOSPHERES
C
C
C
C       ****************************************************************
C
C       TOL = PRESSURE CONVERGENCE TOLERANCE
C       SEED = STARTING TEMPERATURE
C       TNEW = PRESENT ITERATION TEMPERATURE
C       TOLD = MOST RECENT ITERATION TEMPERATURE
C       PCALC= PRESSURE RESULTING FROM TNEW
C       PDIFF= PRESSURE CONVERGENCE TO BE COMPARED WITH TOL
C
C
C       DTPRES = NBS HEPROP FUNCTION TO CALC PRESS. GIVEN T & DENS.
C       DPDT = NBS HEPROP FUNCTION TO CALC PRESS. VS. TEMP. SLOPE
C
C       ****************************************************************
C
C
        TOL=0.001
        SEED=10.0
        TNEW=SEED
C
C
        DO 10, I=1,30
             TOLD=TNEW
             PCALC = DTPRES(DEN,TNEW)
             PDIFF = P - PCALC
             IF (ABS(PDIFF).LE.TOL) GOTO 200
             TNEW = PDIFF/DPDT(DEN,TNEW)+TOLD
    10       CONTINUE
        WRITE(*,'(A)') ' TFDENP DID NOT CONVERGE AFTER 30 ITERATIONS'
        STOP
   200      CONTINUE
        TFDENP = TNEW
        RETURN
        END
C
C
C
```

```
C
$TITLE:'FUNCTION DENSITY; FOR USE WITH HEPROP.FOR'
$SUBTITLE:' CALCULATES DENSITY GIVEN VOLUME & MASS. & CONVERTS UNITS'
$PAGE
C
$NOTRUNCATE
      FUNCTION DENSITY(VOL,MASS)
C
C
      REAL MASS,VOL
C
C
C     I/O UNITS:
C                       VOL     = CUBIC INCHES
C                       MASS    = GRAMS
C                       DENSITY = MOLES/LITER
C
C
C     ****************************************************************
C
C     VOL................ VOLUME
C     MASS................MASS
C     DENSITY.............DENSITY
C
C     ****************************************************************
C
C
      DENSITY=MASS*1000./(VOL*4.0026*16.387)
      RETURN
      END
C
C
$TITLE: 'PFDENS.FOR; FUNCTION FOR USE WITH HEPROP.FOR'
$SUBTITLE: ' CALCULATES PRESSURE GIVEN DENSITY AND ENTROPY'
$PAGE
C
C
      FUNCTION PFDENS(DEN,S)
C
C
C     I/O UNITS:
C                       DENSITY.............MOLES/LITER
C                       TEMPERATURE.........KELVIN
C                       PRESSURE............ATMOSPHERES
C                       ENTROPY.............JOULES/MOLE - KELVIN
C
C
C
C     ****************************************************************
C
C     DEN = DENSITY
C     SEED = STARTING TEMPERATURE
C     TOL = ENTROPY CONVERGENCE  TOLERANCE
C     TNEW = PRESENT ITERATION TEMPERATURE
C     TOLD = MOST RECENT ITERATION TEMPERATURE
C     SCALC= ENTROPY RESULTING FROM TNEW
C     SDIFF= ENTROPY CONVERGENCE TO BE COMPARED WITH TOL
C
C
C     DTPRES = NBS HEPROP FUNCTION TO CALC PRESS. GIVEN T & DENSITY
C     ENTROP = NBS HEPROP FUNCTION TO CALC ENTROPY GIVEN T & DENSITY
C     CV     = NBS HEPROP FUNCTION TO CALC SPECIFIC HEAT AT CONST. VOL.
C                       GIVEN T & DENSITY
C     NOTE: dS/dT AT CONST. V = CV/T
C
C     ****************************************************************
C
C
      TOL  = 0.001
      SEED = 10.0
      TNEW = SEED
C
C
      DO 10, I=1,30
           TOLD  = TNEW
           SCALC = ENTROP(DEN,TNEW)
           SDIFF = S - SCALC
           IF (ABS(SDIFF).LE.TOL) GOTO 200
           TNEW  = SDIFF/(CV(DEN,TNEW)/TNEW)+TOLD
   10      CONTINUE
      WRITE (*,'(A)') ' PFDENS DID NOT CONVERGE AFTER 30 ITERATIONS'
```

```
      STOP
200       CONTINUE
      PFDENS=DTPRES(DEN,TNEW)
      RETURN
      END
```

Appendix E
SOFTWARE

This appendix presents the computer programs used to solve the resultant equations of the analytic models presented in Chapter 4. A Compaq® Portable II personal computer is used to solve these equations. All the programs are in written in Microsoft® QuickBasic ver. 4.0.

E.1 One-Space Model

The program MODEL1.BAS is used to solve the nondimensional equation for the one-space model. The solution of the differential equation is found using the subroutine PCALC which uses a fourth order Runge-Kutta method.[25] This solution is a simulated pressure wave which corresponds to the specified parameters of the model. Adiabatic and isothermal pressure waves are also calculated using perfect gas relations. The pressure waves are reduced to Fourier series using subroutines DFT, which performs the discrete Fourier transform, and CNVPOL which transforms them into normalized Fourier series of the desired form.

E.2 Two-Space Model

The solution of the two-space model used in MODEL2.BAS is similar to the one-space model solution. The primary exception is the method of solution for the differential equations. This model has a system of two simultaneous differential equations. Again a fourth order Runge-Kutta method is used for the solution but the equations are solved in a stepwise simultaneous fashion.

E.3 Gap-Flow And Hybrid Models

As in the two-space model, both the gap-flow model and the hybrid model require that two first order differential equations be solved. The difference for these models is that the two equations do not make up a simulatneous system of equations. One of these

---

25 Kreysig, E., *Advanced Engineering Mathematics*, fourth edition, John Wiley & Sons, New York, 1979.

equations is used if the derivative of the pressure wave is positive and the other is used if the derivative of the pressure wave is negative. Again a fourth order Runge-Kutta method is used for this purpose.

E.4 Program Listings

E.4.1 One-Space Model

```
DECLARE SUB ENABRK ()
DECLARE SUB PCALC (PSTAR!(), VSTAR!(), PAD!(), PISOT!(), SPEED!, N%)
DECLARE SUB DFT (X!(), Y!(), A!(), B!(), N%)
DECLARE SUB CNVPOL (A!(), B!(), PMAG!(), PPHASE!(), VMAG!(), VPHASE!(), M%, N%)
'$TITLE: 'MODEL.BAS; FIRST ORDER GAS SPRING MODEL'
'$SUBTITLE: 'MAIN PROGRAM'
      DEFINT I-N
'        DEFDBL A-H,O-Z
      CONST GAMMA = 1.667, VR = 1.75
      DEF FNF (TEMP, TIME) = SPEED - TEMP * (SPEED + (GAMMA - 1!) * COS(TIME) / ((VR +
1) / (VR - 1) + SIN(TIME)))
      N = 128
      M = 4
      DIM VSTAR(N), PSTAR(N), A(N), B(N), PMAG(M), PPHASE(M), VMAG(M), VPHASE(M),
P(N), V(N), PAD(N), PISOT(N), C(N), D(N), PAPHASE(M), PAMAG(M), PTPHASE(M), PTMAG(M)
      OPEN "SIMM1.DAT" FOR OUTPUT AS #1
      OPEN "SIMM2.DAT" FOR OUTPUT AS #2
      OPEN "SIMP1.DAT" FOR OUTPUT AS #3
      OPEN "SIMP2.DAT" FOR OUTPUT AS #4
      DATA 0.021,0.05,0.065,0.085,0.1,0.2,0.3,0.4,0.5,0.8,0.9
      DATA 1.0,1.3,1.4,1.6,1.8,2.0,2.5,3.0,4.0,5.0,10.0,50.0,100.0
      FOR I = 1 TO 24
            READ SPEED
'             SPEED=1.0
            CALL ENABRK
            CALL PCALC(PSTAR(), VSTAR(), PAD(), PISOT(), SPEED, N)
            CALL DFT(PSTAR(), VSTAR(), A(), B(), N)
            CALL DFT(PAD(), PISOT(), C(), D(), N)
            CALL CNVPOL(A(), B(), PMAG(), PPHASE(), VMAG(), VPHASE(), M, N)
            CALL CNVPOL(C(), D(), PAMAG(), PAPHASE(), PTMAG(), PTPHASE(), M, N)
REM         CALL RCONST(P(),V(),A(),B(),PMAG(),PPHASE(),VMAG(),VPHASE(),M,N)
            PMAG(1) = (PMAG(1) - PTMAG(1)) / (PAMAG(1) - PTMAG(1))
            PMAG(2) = (PMAG(2) - PTMAG(2)) / (PAMAG(2) - PTMAG(2))
            PRINT SPEED, PMAG(1), PPHASE(1), PMAG(2), PPHASE(2)
            PRINT #1, 1 / SPEED, PMAG(1)
            PRINT #2, 1 / SPEED, PMAG(2)
            PRINT #3, 1 / SPEED, PPHASE(1)
            PRINT #4, 1 / SPEED, PPHASE(2)
      NEXT I
FIN:    END
'$SUBTITLE: 'SUBROUTINE CNVPOL; CONVERT TO POLAR COORDINATES'
'$PAGE
SUB CNVPOL (A(), B(), PMAG(), PPHASE(), VMAG(), VPHASE(), M, N) STATIC
      DEFINT I-N: '                          N = NUMBER OF POINTS IN CYCLE
'        DEFDBL A-H,O-Z
      FOR I = 1 TO M: '                        M = NUMBER OF HARMONICS EXAMINED
            K = N + 1 - I
            J = I + 1
            PREAL = (A(K) + A(J)) / N
            PIMAG = (B(K) - B(J)) / N
            VREAL = (B(K) + B(J)) / N
            VIMAG = (A(J) - A(K)) / N
            PPHASE = ATN(PIMAG / PREAL)
'                 IF PREAL<0 THEN PPHASE = PPHASE + 3.1415927
            IF VREAL = 0! THEN VREAL = .0001
            VPHASE = ATN(VIMAG / VREAL)
'                 IF VREAL<0 THEN VPHASE = VPHASE + 3.1415927
            PMAG(I) = SQR(PREAL ^ 2 + PIMAG ^ 2)
            PPHASE(I) = (PPHASE - VPHASE) * 360 / 6.2831853#
             ' PPHASE(I)=(PPHASE)*360/6.2831853
            VMAG(I) = SQR(VREAL ^ 2 + VIMAG ^ 2)
```

```
            VPHASE(I) = VPHASE * 360 / 6.2831853#
        NEXT I
END SUB

'$SUBTITLE:'SUBROUTINE DFT; DISCRETE FOURIER TRANSFORM'
'$PAGE
      SUB DFT (X(), Y(), A(), B(), N) STATIC
      DEFINT I-N
'        DEFDBL A-H,O-Z
      REDIM C(N), S(N)
SKIP:   Q = 6.283185307# / N
      FOR J = 1 TO N
            C(J) = COS(Q * (J - 1))
            S(J) = SIN(Q * (J - 1))
      NEXT J
      FOR J = 1 TO N
            AT = X(1)
            BT = Y(1)
            K = 1
            FOR I = 2 TO N
                  K = K + J - 1
                  IF K > N THEN K = K - N
                  AT = AT + C(K) * X(I) + S(K) * Y(I)
                  BT = BT + C(K) * Y(I) - S(K) * X(I)
            NEXT I
            A(J) = AT
            B(J) = BT
      NEXT J
      END SUB

'$SUBTITLE:'SUBROUTINE ENABRK; ALLOWW PROGRAM TERM. W/O DEBUG'
'$PAGE
SUB ENABRK STATIC
      KEY 15, CHR$(&H4) + CHR$(&H46)
      KEY(15) ON
      ON KEY(15) GOSUB FIN:
      KEY 16, CHR$(&H44) + CHR$(&H46)
      KEY(16) ON
      ON KEY(16) GOSUB FIN:
      END SUB

'$SUBTITLE: 'SUBROUTINE PCALC; CALCULATE PRESSURE WAVE'
'$PAGE
SUB PCALC (PSTAR(), VSTAR(), PAD(), PISOT(), SPEED, N) STATIC
      DEFINT I-N
'        DEFDBL A-H,O-Z
      DT = 2 * 3.1415927# / (2 * N)
      TEOLD = 1!
      TIOLD = 0!
      VSTAR(0) = (1 + (VR - 1) / (VR + 1) * SIN(TIOLD))
      PSTAR(0) = TEOLD / VSTAR(0)
      FOR I = 1 TO 6 * (2 * N)
            B1 = DT * FNF(TEOLD, TIOLD)
            B2 = DT * FNF(TEOLD + B1 / 2, TIOLD + DT / 2)
            B3 = DT * FNF(TEOLD + B2 / 2, TIOLD + DT / 2)
            B4 = DT * FNF(TEOLD + B3, TIOLD + DT)
            TNEW = TEOLD + (B1 + 2 * B2 + 2 * B3 + B4) / 6
            TIOLD = TIOLD + DT
            TEOLD = TNEW
      NEXT I
      FOR I = 1 TO (2 * N)
            B1 = DT * FNF(TEOLD, TIOLD)
            B2 = DT * FNF(TEOLD + B1 / 2, TIOLD + DT / 2)
            B3 = DT * FNF(TEOLD + B2 / 2, TIOLD + DT / 2)
           -B4 = DT * FNF(TEOLD + B3, TIOLD + DT)
           ^ TNEW = TEOLD + (B1 + 2 * B2 + 2 * B3 + B4) / 6
            TIOLD = TIOLD + DT
            VSTAR(I / 2) = (1 + (VR - 1) / (VR + 1) * SIN(TIOLD))
            PAD(I / 2) = 1 / (VSTAR(I / 2) ^ GAMMA)
            PISOT(I / 2) = 1 / VSTAR(I / 2)
            PSTAR(I / 2) = TNEW / VSTAR(I / 2)
            TEOLD = TNEW
      NEXT I
END SUB

'$SUBTITLE:'SUBROUTINE RCONST; RECONSTRUCT WAVE FROM POLAR COORDINATES'
'$PAGE
SUB RCONST (P(), V(), A(), B(), PMAG(), PPHASE(), VMAG(), VPHASE(), M, N) STATIC
      DEFINT I-N
'        DEFDBL A-H,O-Z
      FOR I = 1 TO N
            P(I) = A(1) / N
            V(I) = B(1) / N
```

```
            FOR K = 1 TO M
                  P(I) = P(I) + PMAG(K) * COS(6.283185307# * (K * (I - 1) / N -
PPHASE(K) / 360))
                  V(I) = V(I) + VMAG(K) * COS(6.283185307# * (K * (I - 1) / N -
VPHASE(K) / 360))
            NEXT K
      NEXT I
END SUB
```

E.4.2 Two-Space Model

```
DECLARE SUB MODEL2 (PSTAR!(), VSTAR!(), PAD!(), PISOT!(), NPPC%)
DECLARE SUB DFT (X!(), Y!(), A!(), B!(), N%)
DECLARE SUB CNVPOL (A!(), B!(), PMAG!(), PPHASE!(), VMAG!(), VPHASE!(), M%, N%)
DECLARE SUB RKSIMUL (X!(), Y!(), t!(), H!, N%)
'$TITLE: 'MODEL2B.BAS; FIRST ORDER GAS SPRING MODEL'
'$SUBTITLE: 'MAIN PROGRAM'
'
      CONST PI = 3.14159265#, GAMMA = 1.667, VR = 1.75
      DIM SHARED tau1, tau2, mu AS SINGLE
      DIM N  AS INTEGER, M AS INTEGER, speed AS SINGLE
'
DEF FNDVOV (t) = -COS(t) / ((VR + 1) / (VR - 1) + SIN(t))
'
'
DEF FNX (T1, T2, t) = FNDVOV(t) * (mu * (GAMMA - 1) * T1 + (1 - mu) * (GAMMA - 1) *
T2) + (T2 - T1) / (mu * tau2) - (T1 - 1) / (mu * tau1)
'
'
DEF FNY (T1, T2, t) = FNDVOV(t) * (mu * (GAMMA - 1) * T1 + (1 - mu) * (GAMMA - 1) *
T2) - (T2 - T1) / ((1 - mu) * tau2)
'
'
'    CALL ENABRK
      N = 128
      M = 4
      DIM VSTAR(N), PSTAR(N), A(N), B(N), PMAG(M), PPHASE(M), VMAG(M), VPHASE(M),
P(N), V(N), PAD(N), PISOT(N), C(N), D(N), PAPHASE(M), PAMAG(M), PTPHASE(M), PTMAG(M)
      OPEN "SIMM1.DAT" FOR OUTPUT AS #1
      OPEN "SIMM2.DAT" FOR OUTPUT AS #2
      OPEN "SIMP1.DAT" FOR OUTPUT AS #3
      OPEN "SIMP2.DAT" FOR OUTPUT AS #4
      DATA   0.102, 0.105, 0.11, 0.14, 0.17, 0.2, 0.28, 0.33, 0.4, 0.5, 0.6, 0.7
      DATA 0.8, 0.9, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 3.0, 4.0, 7.0, 10.0,20.0,40.0,100.0
      FOR I = 1 TO 27
      READ speed
           '  speed = speed - .05
      tau1 = .1 * speed
      tau2 = speed
      mu = .1 / speed
      CALL MODEL2(PSTAR(), VSTAR(), PAD(), PISOT(), N)
'   STOP
      CALL DFT(PSTAR(), VSTAR(), A(), B(), N)
      CALL DFT(PAD(), PISOT(), C(), D(), N)
      CALL CNVPOL(A(), B(), PMAG(), PPHASE(), VMAG(), VPHASE(), M, N)
      CALL CNVPOL(C(), D(), PAMAG(), PAPHASE(), PTMAG(), PTPHASE(), M, N)
      PMAG(1) = (PMAG(1) - PTMAG(1)) / (PAMAG(1) - PTMAG(1))
      PMAG(2) = (PMAG(2) - PTMAG(2)) / (PAMAG(2) - PTMAG(2))
      PRINT speed, PMAG(1), PPHASE(1), PMAG(2), PPHASE(2)
      PRINT #1, speed, PMAG(1)
      PRINT #2, speed, PMAG(2)
      PRINT #3, speed, ABS(PPHASE(1))
      PRINT #4, speed, ABS(PPHASE(2))
      NEXT I
'        BEEP
FIN:    END
DEFINT I-N
'$SUBTITLE: 'SUBROUTINE CNVPOL; CONVERT TO POLAR COORDINATES'
'$PAGE
SUB CNVPOL~(A(), B(), PMAG(), PPHASE(), VMAG(), VPHASE(), M, N) STATIC
      DEFINT I-N: '                          N = NUMBER OF POINTS IN CYCLE
'        DEFDBL A-H,O-Z
      FOR I = 1 TO M: '                        M = NUMBER OF HARMONICS EXAMINED
      K = N + 1 - I
      J = I + 1
      PREAL = (A(K) + A(J)) / N
      PIMAG = (B(K) - B(J)) / N
      VREAL = (B(K) + B(J)) / N
      VIMAG = (A(J) - A(K)) / N
      PPHASE = ATN(PIMAG / PREAL)
'               IF PREAL<0 THEN PPHASE = PPHASE + 3.1415927
      IF VREAL = 0! THEN VREAL = .0001
      VPHASE = ATN(VIMAG / VREAL)
'               IF VREAL<0 THEN VPHASE = VPHASE + 3.1415927
      PMAG(I) = SQR(PREAL ^ 2 + PIMAG ^ 2)
      PPHASE(I) = (PPHASE - VPHASE) * 360 / 6.2831853#
             ' PPHASE(I)=(PPHASE)*360/6.2831853
```

```
        VMAG(I) = SQR(VREAL ^ 2 + VIMAG ^ 2)
        VPHASE(I) = VPHASE * 360 / 6.2831853#
        NEXT I
END SUB
'
'
'$SUBTITLE:'SUBROUTINE ENABRK; ALLOW PROGRAM TERM. W/O DEBUG'
'$PAGE
'SUB ENABRK STATIC
'     KEY 15, CHR$(&H04)+CHR$(&H46)
'     KEY(15) ON
'     ON KEY(15) GOSUB FIN:
'     KEY 16, CHR$(&H44)+CHR$(&H46)
'     KEY (16) ON
'     ON KEY(16) GOSUB FIN:
'     END SUB
'$SUBTITLE:'SUBROUTINE DFT; DISCRETE FOURIER TRANSFORM'
'$PAGE
      SUB DFT (X(), Y(), A(), B(), N) STATIC
      DEFINT I-N
'        DEFDBL A-H,O-Z
      REDIM C(N), S(N)
SKIP:   Q = 6.283185307# / N
      FOR J = 1 TO N
      C(J) = COS(Q * (J - 1))
      S(J) = SIN(Q * (J - 1))
      NEXT J
      FOR J = 1 TO N
      AT = X(1)
      BT = Y(1)
      K = 1
      FOR I = 2 TO N
      K = K + J - 1
      IF K > N THEN K = K - N
      AT = AT + C(K) * X(I) + S(K) * Y(I)
      BT = BT + C(K) * Y(I) - S(K) * X(I)
      NEXT I
      A(J) = AT
      B(J) = BT
      NEXT J
      END SUB
'
'
'$SUBTITLE:'MODEL2.BAS; 2ND ORDER GAS SPRING MODEL'
'$PAGE
      SUB MODEL2 (PSTAR(), VSTAR(), PAD(), PISOT(), NPPC%) STATIC
'DEFDBL A-Z
      M% = 24
'
'
'       DEFINE THE FOLLOWING DIFFERENTIAL EQUATIONS
'       FNX IS THE DERRIVATIVE OF T1, dT1/dt = dT1/dt(T1,T2,t)
'       FNY IS THE DERRIVATIVE OF T2, dT2/dt = dT2/dt(T1,T2,t)
'
'
      N% = M% * NPPC%
      DELTAT = 2 * PI / N%
      REDIM T1(N%), T2(N%), time(N%)
'
'
'   OPEN "PAD.DAT" FOR OUTPUT AS #1
'   OPEN "PISOT.DAT" FOR OUTPUT AS #2
'   OPEN "PSTAR.DAT" FOR OUTPUT AS #3
'
'
T1(0) = 1!
T2(0) = 1!
time(0) = 0!
'
'
      CALL RKSIMUL(T1(), T2(), time(), DELTAT, N%)
'
'
      FOR I = 0 TO NPPC%
      VSTAR(I) = (1! + (VR - 1) / (VR + 1) * SIN(time(I * M%)))
      PAD(I) = 1! / (VSTAR(I) ^ GAMMA)
      PISOT(I) = 1! / (VSTAR(I))
      PSTAR(I) = (mu * T1(I * M%) + (1! - mu) * T2(I * M%)) / VSTAR(I)
'       PRINT #1, time(I*M%),PAD(I)
'       PRINT #2, time(I*M%),PISOT(I)
```

```
'       PRINT #3, time(I*M%),PSTAR(I)
      NEXT I
      END SUB
'
'
'
REM $SUBTITLE: 'SUBROUTINE RKSIMUL; 4TH ORDER R-K 2 SIMUL 1ST ORDER EQNS'
REM PAGE
SUB RKSIMUL (X(), Y(), t(), H, N%) STATIC
'
'
'       SOLUTION OF TWO SIMULTANEOUS 1ST ORDER ODEs OF THE FOLLOWING FORM:
'       dX/dt = FNX(X,Y,t)
'       dY/dt = FNY(X,Y,t)
'       THESE EQUATIONS MUST BE DECLARED AS EXTERNAL FUNCTIONS
'
'       THE ARRAY X() WILL CONTAIN THE SOLUTION FOR THE FIRST EQUATION
'       THE ARRAY Y() WILL CONTIAN THE SOLUTION FOR THE SECOND EQUATION
'       THE ARRAY t() WILL CONTAIN VALUES OF THE INDEPENDENT VARIABLE
'
'
'       X(0) = INITIAL VALUE OF X (DEFINED EXTERNAL TO THIS SUBPROGRAM)
'       Y(0) = INITIAL VALUE OF Y (DEFINED EXTERNAL TO THIS SUBPROGRAM)
'       t(0) = INITIAL VALUE OF t (DEFINED EXTERNAL TO THIS SUBPROGRAM)
'       H = TIME STEP SIZE (DEFINED EXTERNAL TO THIS SUBPROGRAM)
'       N = TOTAL NUMBER OF TIME STEPS (DEFINED EXTERNAL TO THIS SUBPROGRAM)
'
'
'
      FOR I% = 0 TO N% - 1
'
      XX = X(I%): YY = Y(I%): tt = t(I%)
      B1 = H * FNY(XX, YY, tt)
      C1 = H * FNX(XX, YY, tt)
      XX = X(I%) + C1 / 2!: YY = Y(I%) + B1 / 2!: tt = t(I%) + H / 2!
      B2 = H * FNY(XX, YY, tt)
      C2 = H * FNX(XX, YY, tt)
      XX = X(I%) + C2 / 2!: YY = Y(I%) + B2 / 2!: tt = t(I%) + H / 2!
      B3 = H * FNY(XX, YY, tt)
      C3 = H * FNX(XX, YY, tt)
      XX = X(I%) + C3: YY = Y(I%) + B3: tt = t(I%) + H
      B4 = H * FNY(XX, YY, tt)
      C4 = H * FNX(XX, YY, tt)
      X(I% + 1) = X(I%) + (C1 + 2 * C2 + 2 * C3 + C4) / 6!
      Y(I% + 1) = Y(I%) + (B1 + 2 * B2 + 2 * B3 + B4) / 6!
      t(I% + 1) = t(I%) + H
'
      NEXT I%
END SUB
```

E.4.3 Gap-Flow Model

```
'$TITLE:'HYBRIDB.BAS; SIIMULATION OF MASS FLOW IN GAP'
'$SUBTITLE:'ASSUMES A ONE SPACE CYLINDER W/ ISOTHERMAL GAP'
DECLARE FUNCTION DP# (t AS DOUBLE, p AS DOUBLE)
DECLARE FUNCTION dpi# (t AS DOUBLE, p AS DOUBLE)
DECLARE SUB ENABRK ()
DECLARE SUB HYBRID (pstar!(), vstar!(), pad!(), pisot!(), SPEED!, n%)
DECLARE SUB DFT (X!(), Y!(), A!(), B!(), n%)
DECLARE SUB CNVPOL (A!(), B!(), pmag!(), pphase!(), VMAG!(), VPHASE!(), M%, n%)
      CONST pi = 3.14159265358#, n% = 256, DELTA = 2! * pi / n%
      CONST GAMMA = 1.667, VR = 1.75, VRG = 10.4, TR = 1.3
      DIM SHARED tau
      DIM M AS INTEGER
      M = 4
      DIM vstar(n%), pstar(n%), A(n%), B(n%), pmag(M), pphase(M), VMAG(M), VPHASE(M),
p(n%), v(n%), pad(n%), pisot(n%), C(n%), D(n%), PAPHASE(M), PAMAG(M), PTPHASE(M),
PTMAG(M)
      OPEN "SIMM1.DAT" FOR OUTPUT AS #1
      OPEN "SIMM2.DAT" FOR OUTPUT AS #2
      OPEN "SIMP1.DAT" FOR OUTPUT AS #3
      OPEN "SIMP2.DAT" FOR OUTPUT AS #4
      OPEN "gap.dat" FOR OUTPUT AS #5
      OPEN "gapit.dat" FOR OUTPUT AS #6
      OPEN "gapad.dat" FOR OUTPUT AS #7
      DATA 0.021,0.05,0.065,0.085,0.1,0.2,0.3,0.4,0.5,0.8,0.9
      DATA 1.0,1.3,1.4,1.6,1.8,2.0,2.5,3.0,4.0,5.0,10.0,50.0,100.0
      FOR I = 1 TO 1
      READ SPEED
      CALL ENABRK
      CALL HYBRID(pstar(), vstar(), pad(), pisot(), SPEED, n%)
      FOR j% = 1 TO n%
      PRINT #5, j%, pstar(j%)
      PRINT #6, j%, pisot(j%)
      PRINT #7, j%, pad(j%)
      NEXT j%
      CALL DFT(pstar(), vstar(), A(), B(), n%)
      CALL DFT(pad(), pisot(), C(), D(), n%)
      CALL CNVPOL(A(), B(), pmag(), pphase(), VMAG(), VPHASE(), M, n%)
      CALL CNVPOL(C(), D(), PAMAG(), PAPHASE(), PTMAG(), PTPHASE(), M, n%)
REM       CALL RCONST(P(),V(),A(),B(),PMAG(),PPHASE(),VMAG(),VPHASE(),M,n%)
      pmag(1) = (pmag(1) - PTMAG(1)) / (PAMAG(1) - PTMAG(1))
      pmag(2) = (pmag(2) - PTMAG(2)) / (PAMAG(2) - PTMAG(2))
      loss = (.71 * pmag(1) + .29) * SIN(pphase(1) / 180 * pi + pi)
      PRINT SPEED, pmag(1), pphase(1), pmag(2), pphase(2), loss
      PRINT #1, 1 / SPEED, pmag(1)
      PRINT #2, 1 / SPEED, pmag(2)
      PRINT #3, 1 / SPEED, pphase(1)
      PRINT #4, 1 / SPEED, pphase(2)
      NEXT I
FIN:    END
DEFINT I-N
'$SUBTITLE: 'SUBROUTINE CNVPOL; CONVERT TO POLAR COORDINATES'
'$PAGE
SUB CNVPOL (A(), B(), pmag(), pphase(), VMAG(), VPHASE(), M, n) STATIC
      DEFINT I-N: '                          N = NUMBER OF POINTS IN CYCLE
'        DEFDBL A-H,O-Z
      FOR I = 1 TO M: '                        M = NUMBER OF HARMONICS EXAMINED
      K = n + 1 - I
      j = I + 1
      PREAL = (A(K) + A(j)) / n
      PIMAG = (B(K) - B(j)) / n
      VREAL = (B(K) + B(j)) / n
      VIMAG = (A(j) - A(K)) / n
      pphase = ATN(PIMAG / PREAL)
'               IF PREAL<0 THEN PPHASE = PPHASE + 3.1415927
      IF VREAL = 0! THEN VREAL = .0001
      VPHASE = ATN(VIMAG / VREAL)
'               IF VREAL<0 THEN VPHASE = VPHASE + 3.1415927
      pmag(I) = SQR(PREAL ^ 2 + PIMAG ^ 2)
      pphase(I) = (pphase - VPHASE) * 360 / 6.2831853#
            ' PPHASE(I)=(PPHASE)*360/6.2831853
      VMAG(I) = SQR(VREAL ^ 2 + VIMAG ^ 2)
      VPHASE(I) = VPHASE * 360 / 6.2831853#
      NEXT I
END SUB
```

```
'$SUBTITLE:'SUBROUTINE DFT; DISCRETE FOURIER TRANSFORM'
'$PAGE
      SUB DFT (X(), Y(), A(), B(), n) STATIC
      DEFINT I-N
'        DEFDBL A-H,O-Z
      REDIM C(n), S(n)
SKIP:   Q = 6.283185307# / n
      FOR j = 1 TO n
      C(j) = COS(Q * (j - 1))
      S(j) = SIN(Q * (j - 1))
      NEXT j
      FOR j = 1 TO n
      AT = X(1)
      BT = Y(1)
      K = 1
      FOR I = 2 TO n
      K = K + j - 1
      IF K > n THEN K = K - n
      AT = AT + C(K) * X(I) + S(K) * Y(I)
      BT = BT + C(K) * Y(I) - S(K) * X(I)
      NEXT I
      A(j) = AT
      B(j) = BT
      NEXT j
      END SUB

DEFSNG I-N
FUNCTION D2# (t AS DOUBLE, p AS DOUBLE)
DP# = (COS(t) / (((VR + 1!) / (VR - 1!) * (VRG / (TR + VRG)) + SIN(t)) * (1! / ((p) -
VRG - 1!) - 1! / (GAMMA * (p)))))
END FUNCTION

FUNCTION dpi# (t AS DOUBLE, p AS DOUBLE)
dpi# = ((p) * (-GAMMA) * TR * COS(t) / ((VR + 1!) / (VR - 1!) * (GAMMA + (VRG * (TR -
GAMMA)) / (VRG + TR)) + TR * SIN(t)))
END FUNCTION

DEFINT I-N
'$SUBTITLE:'SUBROUTINE ENABRK; ALLOWW PROGRAM TERM. W/O DEBUG'
'$PAGE
SUB ENABRK STATIC
      KEY 15, CHR$(&H4) + CHR$(&H46)
      KEY(15) ON
      ON KEY(15) GOSUB FIN:
      KEY 16, CHR$(&H44) + CHR$(&H46)
      KEY(16) ON
      ON KEY(16) GOSUB FIN:
      END SUB

DEFSNG I-N
SUB HYBRID (pstar(), vstar(), pad(), pisot(), tau, nopts AS INTEGER)
      DIM p AS DOUBLE, t AS DOUBLE, b1 AS DOUBLE, b2 AS DOUBLE, b3 AS DOUBLE
      DIM b4 AS DOUBLE, v AS DOUBLE

      p = 1!
      t1 = 0!
      FOR t = 0 TO 6 * nopts * DELTA STEP DELTA
      v = 1! + (VR - 1!) / (VR + 1!) * SIN(t)
      IF DP#(t, p) >= 0! THEN
      b1 = DELTA * DP#(t, p)
      b2 = DELTA * DP#(t + DELTA / 2!, p + b1 / 2!)
      b3 = DELTA * DP#(t + DELTA / 2!, p + b2 / 2!)
      b4 = DELTA * DP#(t + DELTA, p + b3)
      ELSE
      b1 = DELTA * dpi#(t, p)
      b2 = DELTA * dpi#(t + DELTA / 2!, p + b1 / 2!)
      b3 = DELTA * dpi#(t + DELTA / 2!, p + b2 / 2!)
      b4 = DELTA * dpi#(t + DELTA, p + b3)
      END IF
      p = p + (b1 + 2! * b2 + 2! * b3 + b4) / 6!
      NEXT t
      pref = p
      count% = 0
      FOR count% = 1 TO nopts
      t = t + DELTA
      v = 1! + (VR - 1!) / (VR + 1!) * SIN(t)
      pad(count%) = pref * (1! / v) ^ GAMMA
      pisot(count%) = pref * (1! / v)
      pstar(count%) = p
      vstar(count%) = v
      PRINT pstar(count%), vstar(count%)
      t1 = t1 + DELTA
```

```
      IF DP#(t, p) >= 0! THEN
      b1 = DELTA * DP#(t, p)
      b2 = DELTA * DP#(t + DELTA / 2!, p + b1 / 2!)
      b3 = DELTA * DP#(t + DELTA / 2!, p + b2 / 2!)
      b4 = DELTA * DP#(t + DELTA, p + b3)
      ELSE
      b1 = DELTA * dpi#(t, p)
      b2 = DELTA * dpi#(t + DELTA / 2!, p + b1 / 2!)
      b3 = DELTA * dpi#(t + DELTA / 2!, p + b2 / 2!)
      b4 = DELTA * dpi#(t + DELTA, p + b3)
      END IF
      p = p + (b1 + 2! * b2 + 2! * b3 + b4) / 6!
      NEXT count%
END SUB

DEFINT I-N
'$SUBTITLE:'SUBROUTINE RCONST; RECONSTRUCT WAVE FROM POLAR COORDINATES'
'$PAGE
SUB RCONST (p(), v(), A(), B(), pmag(), pphase(), VMAG(), VPHASE(), M, n) STATIC
      DEFINT I-N
'        DEFDBL A-H,O-Z
      FOR I = 1 TO n
      p(I) = A(1) / n
      v(I) = B(1) / n
      FOR K = 1 TO M
      p(I) = p(I) + pmag(K) * COS(6.283185307# * (K * (I - 1) / n - pphase(K) / 360))
      v(I) = v(I) + VMAG(K) * COS(6.283185307# * (K * (I - 1) / n - VPHASE(K) / 360))
      NEXT K
      NEXT I
END SUB
```

E.4.4 Hybrid Model

```
'$TITLE:'hyb.BAS; SIIMULATION OF MASS FLOW IN GAP' */
'$SUBTITLE:'ASSUMES A ONE SPACE CYLINDER W/ ISOTHERMAL GAP' */
DECLARE FUNCTION DP! (T AS DOUBLE, P AS DOUBLE)
DECLARE FUNCTION DPI! (T AS DOUBLE, P AS DOUBLE)
CONST VR = 1.75, GAMMA = 1.667
CONST PI = 3.14159265#, NOPTS = 128, DELTA = 2! * PI / NOPTS
      DIM P AS DOUBLE, T AS DOUBLE, b1 AS DOUBLE, b2 AS DOUBLE, b3 AS DOUBLE
      DIM b4 AS DOUBLE, V AS DOUBLE, PAD AS DOUBLE
      DIM SHARED tau AS DOUBLE, tr AS DOUBLE, vrg AS DOUBLE
      DATA 0.1, 1.0, 2.0, "K", 0.1, 1.0, 4.0, "L", 0.1, 1.0, 8.0, "M"
      DATA 0.1, 2.0, 2.0, "d", 0.1, 4.0, 2, "e", 0.1, 2.0, 4.0, "f"
      DATA 0.1, 2.0, 8.0, "g"
FOR count% = 1 TO 3
      READ tau, tr, vrg, num$
      P = 1!
      t1 = 0!
      OPEN "Phyb" + num$ + ".DAT" FOR OUTPUT AS #1
      OPEN "PhybAD" + num$ + ".DAT" FOR OUTPUT AS #2
      OPEN "PhybIT" + num$ + ".DAT" FOR OUTPUT AS #3
      FOR T = 0 TO 6 * NOPTS * DELTA STEP DELTA
      V = 1! + (VR - 1!) / (VR + 1!) * SIN(T)
      PAD = (1! / V) ^ GAMMA
      IF DPI(T, P) <= 0! THEN
      b1 = DELTA * DP(T, P)
      b2 = DELTA * DP(T + DELTA / 2!, P + b1 / 2!)
      b3 = DELTA * DP(T + DELTA / 2!, P + b2 / 2!)
      b4 = DELTA * DP(T + DELTA, P + b3)
      ELSE
      b1 = DELTA * DPI(T, P)
      b2 = DELTA * DPI(T + DELTA / 2!, P + b1 / 2!)
      b3 = DELTA * DPI(T + DELTA / 2!, P + b2 / 2!)
      b4 = DELTA * DPI(T + DELTA, P + b3)
      END IF
      P = P + (b1 + 2! * b2 + 2! * b3 + b4) / 6!
      NEXT T
      Pref = P
      FOR T = 6 * NOPTS * DELTA TO 7 * NOPTS * DELTA STEP DELTA
      V = 1! + (VR - 1!) / (VR + 1!) * SIN(T)
      PAD = Pref * (1! / V) ^ GAMMA
      PRINT #1, t1, P
      PRINT #2, t1, PAD
      PRINT #3, t1, Pref * (1! / V)
      t1 = t1 + DELTA

      IF DPI(T, P) <= 0! THEN
      b1 = DELTA * DP(T, P)
      b2 = DELTA * DP(T + DELTA / 2!, P + b1 / 2!)
      b3 = DELTA * DP(T + DELTA / 2!, P + b2 / 2!)
      b4 = DELTA * DP(T + DELTA, P + b3)
      ELSE
      b1 = DELTA * DPI(T, P)
      b2 = DELTA * DPI(T + DELTA / 2!, P + b1 / 2!)
      b3 = DELTA * DPI(T + DELTA / 2!, P + b2 / 2!)
      b4 = DELTA * DPI(T + DELTA, P + b3)
      END IF
      P = P + (b1 + 2! * b2 + 2! * b3 + b4) / 6!
      NEXT T
CLOSE #1
CLOSE #2
CLOSE #3
NEXT count%

FUNCTION DP (T AS DOUBLE, P AS DOUBLE)
DP = ((GAMMA - 1!) * P * tau * (1 / P + (vrg + (vrg + tr) * (VR - 1) / (VR + 1) *
SIN(T)) / (P - vrg - 1!)) - P * GAMMA * COS(T)) / ((VR + 1!) / (VR - 1!) * (vrg +
GAMMA) / (vrg + tr) + SIN(T))
END FUNCTION

FUNCTION DPI (T AS DOUBLE, P AS DOUBLE)
DPI = ((GAMMA - 1!) * tau * (1 / P + (vrg + (vrg + tr) * (VR - 1!) / (VR + 1!) *
SIN(T)) / (P - vrg - 1!)) - GAMMA * COS(T)) / ((1 / P + GAMMA / (vrg + 1 - P)) * ((VR
+ 1!) / (VR - 1!) * vrg / (vrg + tr) + SIN(T)))
END FUNCTION
```