To the Graduate Council:

I am submitting herewith a dissertation written by D. Scott Crocker entitled "Thermally-Induced Convection Heat Transfer in Confined Air Subjected to Rapid Boundary Heating." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mechanical Engineering.

M. Parang, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

THERMALLY-INDUCED CONVECTION HEAT TRANSFER IN CONFINED AIR SUBJECTED TO RAPID BOUNDARY HEATING

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

D. Scott Crocker

August 1991

ACKNOWLEDGEMENTS

I would like to thank Dr. M. Parang, my major professor, for his direction and inspiration. He is a superb educator and a good friend. Thanks to my wife, Ruth, for her patience and understanding. Her encouragement was critical.

I am grateful for the funding provided during part of my studies through NASA Lewis Research Center.

ABSTRACT

Thermally-induced convection heat transfer is investigated in a series of experiments and by numerical modeling. Applications and potential importance of this process are discussed for both gravity and microgravity. The design of an in-space experiment for studying the process in microgravity is presented.

Heat transfer in air enclosed between two parallel plates and subjected to rapid heat addition from the upper plate is investigated experimentally. The transient fluid temperature was measured using a new and unique optical technique developed for this purpose. The temperature along a cross-section of the test volume was found by measuring the refractive effects on a planar laser sheet that traversed the test volume. The heated plate temperature was measured by thermocouple.

The optical temperature measurement technique is adapted for the in-space experiment design. A system consisting of a series of prisms that amplifies the laser deflection is presented. This is necessary since the space experiment must be contained in a small volume. The ground experiments demonstrate that the experimental technique is suitable and feasible to implement in a space experiment.

Numerical modeling of the process is investigated by first analyzing previous numerical work. Shortcomings of the previous work are discussed and several changes are introduced that improve the accuracy of the numerical scheme. Secondly, an alternative numerical scheme is presented that neglects the acoustic velocity perturbations. The convective effects of the bulk motion generated in the fluid by the heat addition are calculated by requiring that continuity be satisfied in a spatially homogeneous pressure field. The momentum equations are eliminated in this scheme, so the numerical solution is greatly simplified.

The results of the latter numerical model are compared with the experimental results. Good agreement is found for short times, but at longer times the experimental temperatures are below the numerical results. The numerical model is also compared with other previous numerical and analytical works.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

a	speed of sound
c_p	constant pressure heat capacity
c_v	constant volume heat capacity
C	Gladstone-Dale constant
d	laser beam diameter
E	energy
f	lens focal length; or heated wall flatness error
H	length of test section in y-direction
i	node index
k	electrical resistance of heater element material; or thermal conductivity
K	constant
L	length of test section in z-direction
n	index of refraction
P	pressure
Pe	Peclet number = RePr
Pr	Prandtl number = $\mu' c_p' / k'$
q'''	heat generation
$\dot{q}$	heat transfer rate
r	angle ratio of laser entering and exiting prism
R	gas constant
Re	Reynolds number = $(\rho' H' \sqrt{R' T_o'}) / \mu'$
t	time; or thickness of laser shhet; or heater element thickness
T	temperature
v	fluid velocity in y-direction

V volts

w width of laser sheet (see Figure 2-6)

W length of test section in x-direction

x space coordinate normal to laser beam and parallel to heated wall

y space coordinate normal to heated wall

z space coordinate parallel to laser beam

α incident angle of laser beam; or thermal diffusivity

β calibration angle

γ angle of prism orientation; orspecific heat ratio

θ angle of laser sheet from normal to heated wall

λ second coefficient of viscosity

μ first coefficient of viscosity

ρ density

Subscripts

a adjusted property

b edge of thermal boundary layer

c cylindrical lens; or core property

d deflected

e exit of test section; or end of measured temperature profile

ex indicates "extra" volume

o initial value

s value at screen; or spherical lens

u undeflected

w value at the wall

CHAPTER 1

INTRODUCTION

1.1 Background

When a compressible fluid is subjected to a large sudden heat flux, a localized expansion of the fluid occurs which results in acoustic-type fluid velocity perturbations which are superimposed on a bulk velocity field. The heat flux may occur at a boundary as in direct heating of a wall or, as in the case of cryogenic fluid transport, sudden contact with a surface at room temperature. Heat addition may also occur as a result of chemical reaction within the flow. Immediately after the initiation of the heat flux, the temperature of the fluid adjacent to the heat source increases rapidly as a result of conduction heat transfer. The temperature increase causes a volume expansion (with an effect similar to that of a moving piston) which, in turn, increases the local fluid pressure. The resulting pressure waves travel through the fluid medium at the local speed of sound. These pressure waves, which are of thermal origin, may be termed thermoacoustic or thermoconvective waves. In addition to the acoustic-type fluid motion which occurs on a fast acoustic time scale, there is also a bulk fluid motion which develops on a slower conduction time scale as the fluid responds to the changing temperature and pressure fields. This bulk fluid motion and the superimposed acoustic-type motion add a convective heat transfer mode to an otherwise quiescent fluid.

If the rate of heat addition is sufficiently large, the resulting thermally-induced motion may have significant effects, especially in a microgravity environment where natural convection is usually negligible. The effects will generally be greatest in a fluid near the critical point where thermodynamic and transport properties are strong functions of temperature. Technological processes where this thermally-induced motion may be important are numerous. Cryogenic fluid storage and transfer systems experience large sudden temperature changes in transfer lines and in filling of storage containers. Accurate prediction of heat transfer rates in cryogenic systems are especially critical for spacecraft applications. Water or other fluids near the critical point may experience significant oscillations in applications such as steam power generation or refrigeration systems. Combustion systems can also experience strong acoustic oscillations which can affect the flow field and possibly degrade performance of the system. Space-based manufacturing processes where precise knowledge of heat flux may be required may also be influenced by thermally-induced convective motion if temperature gradients are sufficiently large.

1.2 Literature Review

Historical

The interaction between heat transfer and sound waves was first observed in 1850 by Sondhauss and in 1859 by Rijke as reported by Feldman [1,2]. Sondhauss noted that glassblowers' blowing tube would often "sing" when the glass bulb was heated. Rijke found that a tone was produced when a heat source was placed in the lower half of a vertical open-ended tube. The criterion for sustaining thermally-induced acoustic vibrations was given by Lord Rayleigh [3] as, "If heat be given to the air at the moment of greatest condensation or taken from it at the moment of greatest rarefaction, the vibration is encouraged."

Experimental--gravity environment

Many experiments have been performed to measure the acoustic response to a temperature disturbance in a fluid or the effect of an acoustic field on heat transfer and fluid flow. Selected experimental observations are briefly discussed below.

Thermally-induced oscillations or convective heat transfer effects have been observed in several cryogenic systems in a gravity environment. Large oscillations were observed by Clement and Gaffney [4] when filling a flask with liquid helium using a transfer siphon. In these experiments the temperature of the cold end of the siphon was the boiling point of helium in the vented storage vessel and the hot end was at room temperature. It was found that the temperature distribution along the length of the tube is the key parameter determining oscillation frequency.

Density waves have been measured in supercritical helium flowing in high aspect ratio channels designed to cool superconducting power transmission lines by Daney, Ludtke and Jones [5]. Unstable density waves were observed as temperature (density) disturbances move down the heated channel at the fluid velocity producing flow and temperature oscillations. It was noted that the growth of thermal pulses occurs because warmer (less dense) fluid experiences greater heating per unit mass for constant wall heat flux thus enhancing the growth of the thermal pulse. Temperature oscillations as high as 11°K were observed which could result in the failure of a superconducting power transmission system if the superconductor reaches its transition temperature.

Acoustic signals generated by a laser-induced transient heat pulse have been directly measured by

Carome, Clark and Moeller [6]. Thermally-induced acoustic oscillations were measured for supercritical water flowing through horizontal small bore tubes with uniform heat flux by Stewart, Stewart and Watson [7]. The water was at a pressure of 250 bars flowing with Reynolds numbers between 48000 and 90000. Oscillations were identified with standing pressure waves in the test section. Test section axial wall temperature had significant maxima and minima when oscillations occurred compared to a linear profile before oscillations. Heat transfer coefficients varied up to 100% between these maxima and minima.

Welsh, et al. [8] discuss several experiments describing the influence of sound in flow and heat transfer test rigs. Sound levels between 70 and 95 dB generated by the flow, but not obviously apparent to the experimenter, can alter both the flow and heat transfer characteristics. Resonant acoustic modes can "feed back" onto regions of separated flow with a resulting increase in heat transfer and aerodynamic drag.

Experimental--microgravity environment

There is a limited amount of experimental information on thermally-induced convection heat transfer processes in microgravity conditions. Low heat flux experiments were carried out in space and other work has been done in drop tower facilities and aboard KC-135 flights.

The first experiments to measure heat transfer in a low gravity environment were the Heat Flow and Convection experiments performed on Apollo flights 14 and 17. The Apollo 14 experiments [9] indicated a significant increase in heat transfer over that expected from pure conduction, however, improved versions of the experiments on Apollo 17 [10] showed no significant increase. Grodzka and Bannister concluded that the heat addition rate of 5 watts was too low to generate significant convective motion and that higher heat transfer rates on Apollo 14 were probably caused by g-jitter which was more severe on that flight.

Experiments were performed by Parang and Salah-Eddine [11,12] using the Two-Seconds Zero-Gravity Drop Tower facility at NASA Lewis Research Center. Air at atmospheric pressure and ambient temperature was enclosed in a cylindrical geometry and was heated rapidly at one boundary during the two second experiment time. Approximately 3.6 kW was applied to the boundary surface raising it to a temperature of 1400°F in two seconds. The transient air temperature was measured along the axis of the cylinder using thermocouples. The spatial temperature distribution was corrected for radiation-induced error and compared with a numerically calculated conduction-only temperature solution. The results indicated a heat transfer rate significantly higher than that calculated from the pure conduction solution. The measured heat transfer rate was, however, much less than the rate predicted by other numerical models. These experiments were repeated in a one-g environment without significant deviation of the results from the zero-g results. Parang and Catt [13] continued the work in a one-g environment by performing vacuum tests as well as tests in air to measure the radiation-induced error. It was found that vacuum tests produced higher thermocouple temperatures than air tests suggesting that thermally-induced air motion actually cools the thermocouples. This implies that the thermocouples were heated entirely by radiation even for the tests in air. Parang and Catt used this cooling effect to measure thermally-induced convection and found that there was significant bulk fluid motion. It is clear from these results that accurate air temperature measurements using thermocouples are difficult if not impossible to obtain.

Giarratano, et al. [14] performed an experimental study that was flown on a NASA KC-135 aircraft where approximately 25 seconds of low-gravity experimental time is available. A narrow strip of titanium was heated by an electric pulse in a test cell containing Refrigerant-13. The transient temperature profile of the fluid adjacent to the heated strip was measured using diffuse-light holographic interferometry. The accuracy of the data was difficult to assess because of experimental limitations on the KC-135. Ground experiments were also performed and the results compared to a constant pressure (i.e. no acoustic fluid oscillations) numerical model. The experimentally determined fluid temperature profiles (valid during the 70 milliseconds before the onset of natural convection) were consistently lower than those predicted by the numerical model.

Analytical and Numerical

There have been many theoretical investigations of the effect of boundary temperature variations on a compressible fluid. Trilling [15] studied the effect of a sudden temperature jump at the closed end of a semi-infinite gas-filled pipe. A linear perturbation analysis was used to show that the boundary temperature jump causes a local gas expansion which acts much like a moving piston. The resulting pressure pulse has sonic velocity with intensity proportional to the boundary temperature disturbance and inversely proportional to the fourth root of distance traveled.

Chu [16] developed a linear model of the flow field in a constant cross-section tube with heat addition at a single plane. For a moderate rate of heat release the problem is reduced to that of a moving piston. An exact solution to the nonlinear equations is given for uniform heat release at a constant rate. A solution is given for the strength of the resulting shock waves in terms of the constant rate of heat release. Chu and Ying [17,18] solved the nonlinear equations for a perfect gas in a pipe closed at both ends with a plane heater at the midsection for two special case heating conditions.

Rott [19] reviewed the theory for several thermoacoustic phenomena. Unstable configurations in which large amplitude acoustic oscillations are maintained by heated surfaces and the production of heat by sound which involves the effect of thermoacoustic streaming is investigated. The study of heat transfer by thermoacoustic streaming was continued for the case of an oscillating gas at the closed end of a tube [20].

The response of an ideal gas to moderate boundary heating and to rapid boundary heating was investigated by Kassoy [21] and Radhwan and Kassoy [22], respectively. The gas is confined by infinite parallel walls and subjected to a heat addition at both boundaries. Gravity is neglected in both cases. Solutions are developed using the singular perturbation method employing a small parameter defined by the ratio of a rapid acoustic time scale and a slow conduction time scale. On the acoustic time scale, the problem is shown to consist of thin conductive boundary layers adjacent to the walls and an isentropic core where pressure waves travel back and forth. As time progresses, the conductive boundary layers grow on the conduction time scale of the confining slot. In the case of rapid boundary heating, the transient rate of heat transfer to the compressible gas is shown to be significantly greater than for an equivalent rigid material.

Thermally driven acoustic oscillations are common in cryogenic flow lines because of the high temperature gradients that are often present. Liburdy [23] analyzed helium flowing at a constant rate through an annular gap in a constant diameter tube. The tube wall boundary temperature experiences a sudden temperature jump from the cold upstream section to the hot downstream section. The annular configuration was studied because it is known that a rod inserted into a flow line has a stabilizing effect. A first order perturbation expansion of the conservation equations was used to determine the range of stable flow. It was found that the range of hot-to-cold temperature ratios that produce unstable acoustic oscillations is decreased as the annular gap size is decreased. Heat transfer to a compressible fluid confined in a small tube, such that the induced fluid motion is parallel to the heat transfer surface, was investigated by Arp and Persichetti [24]. The complete conservation equations for one-dimensional flow in the axial direction are solved numerically.

A fluid bounded by a wall on one side and otherwise unbounded and with a gravity field and temperature gradient parallel to the wall, requiring that there be no natural convection, was considered by Luikov and Berkovsky [25]. The wall velocity and temperature are varied harmonically. It is shown that weakly damped thermal and shearing waves (waves transported solely by thermal conductivity and viscosity rather than acoustic type waves) may propagate under certain conditions. Keck and Schneider [26] considered compressibility effects in a similar problem. They showed that conditions required for the propagation of thermal and shearing waves leads to unstable conditions such that natural convection would dominate. If the fluid is bounded vertically, however, thermal and shearing waves may propagate. The interaction of thermal and shearing waves with acoustic waves was also investigated.

Larkin [27], and later Spradley, et al. [28], used a finite difference approximation to solve the governing equations for an ideal compressible gas confined between two infinite walls. The walls and the enclosed fluid are initially at a constant temperature T_0 and at time t=0 one of the walls is instantly raised to a large constant temperature T_w. Thermal fluid properties are assumed constant and gravity is neglected. In each case pressure waves propagate back and forth between the walls at the speed of sound. The transient temperature profile results are similar in the two studies and they both indicate that the acoustic motion strongly enhances the rate of heat transfer. However, the velocity amplitude obtained by Larkin is about an order of magnitude higher than that of Spradley, et al. The unexpectedly rapid fluid temperature change toward steady state predicted by these models is a key reason for interest in thermally-induced convection heat transfer processes and a primary motivation for the present study. Spradley and Churchill [29] extended the study to include gravity effects in two-dimensional flow. The zero-gravity case is equivalent to the previous one-dimensional work and produced the same results.

A scaling analysis of the problem similar to that studied by Larkin and Spradley, et al. was performed by Krane and Parang [30]. They discussed various time scales in the problem and demonstrated that the fluid temperature distribution is dominated by conduction terms long before the pressure waves dissipate. The governing equations for the problem considered by Larkin and Spradley, et al. were scaled in this study, but no attempt was made to solve the resulting normalized equations.

1.3 The Present Study

There is a considerable amount of experimental data that demonstrates the presence of thermally-induced convective motion, including limited quantitative data for certain geometries for cryogenic fluids and critical pressure water as previously discussed. However, only a few experimenters have attempted to measure the spatial and temporal temperature variation in a fluid experiencing a thermally-induced mechanical response. The investigations of Parang, Salah-Eddine and Catt [11,12,13] were limited primarily because of the impossibility of shielding thermocouples in the highly radiative environment required to produce significant thermoconvective effects. Zero-g experiments were also limited to just two seconds duration. The experimental work of Giarratano [14] was limited to less than 0.1 seconds on the ground because of the onset of natural convection and zero-g experiments aboard the KC-135 failed to produce very accurate temperature measurements because of in-flight experimental constraints.

The main focus of this investigation is to provide accurate measurements of the transient spatial temperature profile of air adjacent to a rapidly heated wall. This includes the results of ground experiments performed in support of the design of a microgravity space experiment. The experimental data can be used to develop accurate modeling of the phenomenon and help enhance the understanding of the fundamental process occurring in a fluid subjected to rapid heating at a boundary. An optical method was used to measure the air temperature. The optical system that was developed is essentially a modified Schmidt-schlieren system which is capable of measuring the fluid temperature gradient along a cross section of the test section [31]. The modification of the Schmidt-schlieren system provides a direct and accurate measurement of the spatial variation of the fluid index of refraction. This data is then used to calculate one-dimensional transient air temperature profiles. This new and unique method, to the author's knowledge, has not previously been used for fluid temperature measurement. A brief study of the feasibility of using this temperature measurement technique in other applications is given in Appendix A.

Chapter 2 describes the experimental apparatus, including the optical temperature measurement system. The experimental data, the analysis required to obtain spatial temperature profile results, and the results themselves are presented in Chapter 3. Chapter 4 contains a description of the preliminary design of a space experiment performed for NASA Lewis Research Center.

Chapter 5 includes an analysis of the numerical scheme of Spradley, et al. followed by the description of a simple numerical model based on control volume formulation which predicts thermally-induced convective heat transfer. The numerical scheme considers the bulk motion of the fluid in response to the slowly changing temperature and pressure fields, but neglects the acoustic disturbances that develop on the fast acoustic time scale of the problem. This simplification avoids some of the difficulties encountered in previous numerical modeling efforts. Comparison is made between these two numerical schemes and the present experimental data in Chapter 6.

Also, the present numerical work is compared to the results of the perturbation analysis of Radhwan and Kassoy [22]. Finally, the report is completed by concluding remarks and suggestions for further study in Chapter 7.

CHAPTER 2

EXPERIMENTAL INVESTIGATION

2.1 Overview

The main objective of the experimental investigation was to measure the transient temperature of air as a function of distance from a rapidly heated wall. Knowledge of the air temperature profile allows an assessment of the relative importance of conduction and thermally-induced convection heat transfer. The convective motion of the air consists of both acoustic-type velocity oscillations and the bulk motion of the expanding gas. A schematic of the experimental apparatus is shown in Figure 2-1.

A test section, or heated cavity, with a heated surface located on the top wall was designed and constructed for the experiment. Locating the heated wall on top minimized the effects of gravity-induced convection. The side walls were insulated to provide an approximately one-dimensional environment. The transient temperature profile of the air was measured by measuring the deflection of a laser beam caused by refraction as the beam passed through the heated air. The transient

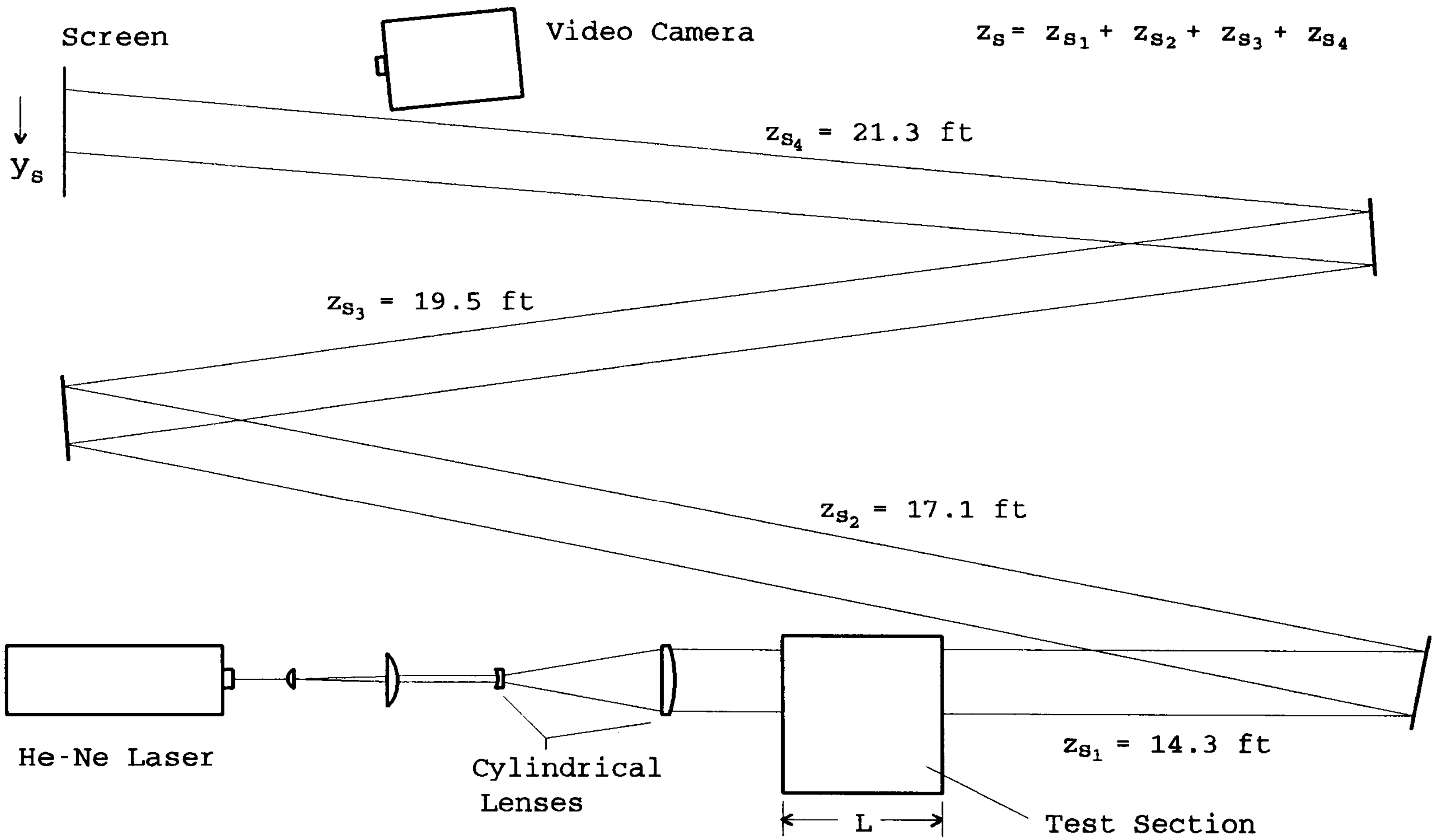


Figure 2-1. Schematic of Experimental Apparatus

deflection of the laser beam was recorded with a video camera.

The temperature of the walls of the test section, especially the heated wall, was measured by thermocouples. The temperature of the side walls was measured so that three-dimensional effects could be considered in the analysis of the experimental data. The pressure in the sealed test section, which provides an indication of the total energy transfer from the heated wall to the air, was measured with a pressure transducer.

2.2 Heated Cavity

The heated cavity apparatus provided a surface that was capable of quickly reaching high temperatures and a test section in which one-dimensional heat transfer and fluid flow was approximated. A schematic of the heated cavity apparatus is shown in Figures 2-2 and 2-3. Detailed drawings of the heated cavity apparatus are in Appendix B.

The heated cavity apparatus included a heater element (heated wall) that consists of 0.001 inch thick stainless steel foil. Electrical connections to the foil were made through copper busbars that were clamped across the full width of each end of the heater element. One of the copper busbars was free to slide on a spring assembly

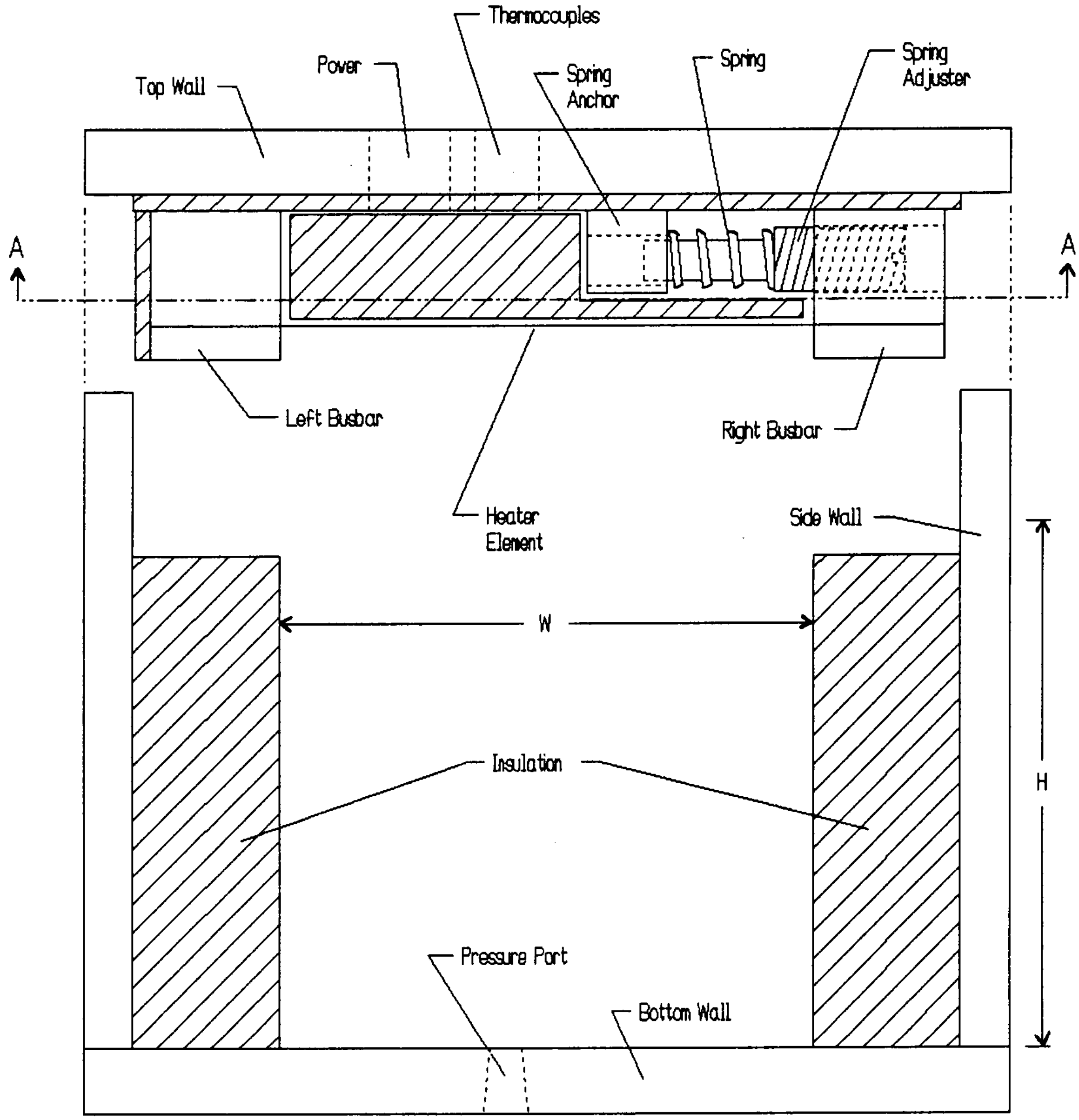
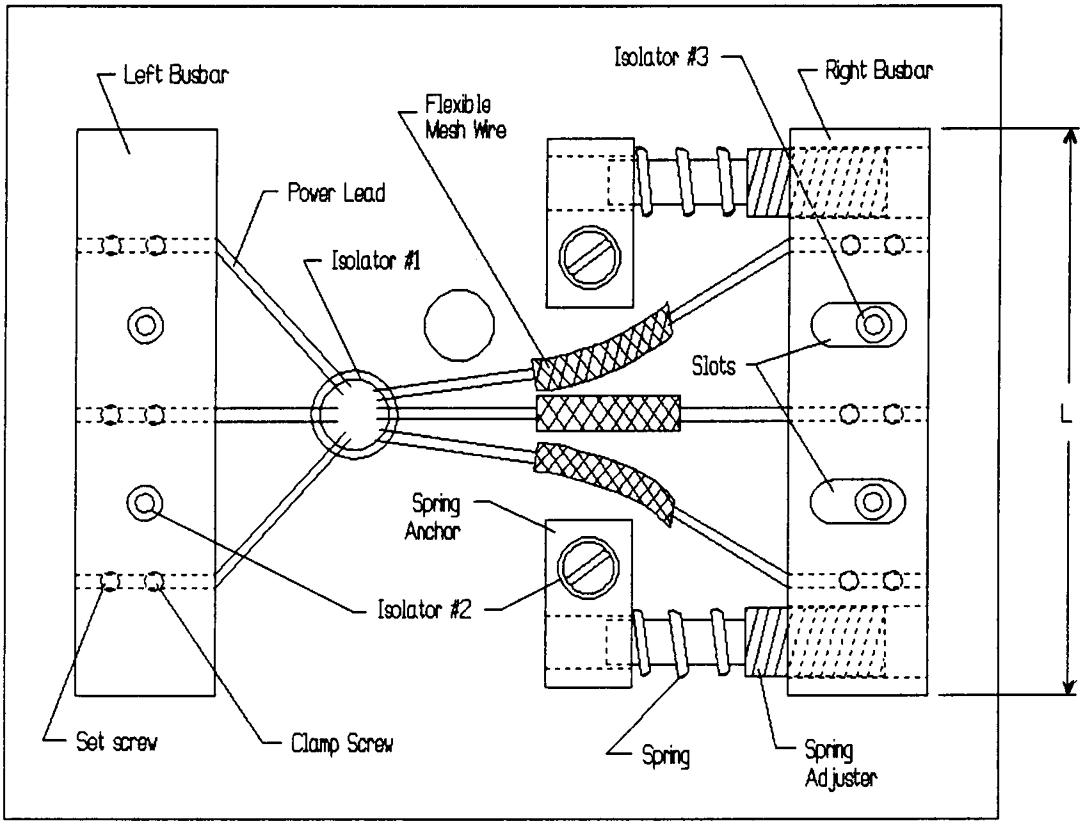


Figure 2-2. Schematic of Heated Cavity Apparatus



NOTE: Insulation not shown

Figure 2-3. Schematic of Heated Cavity Apparatus, Section A-A which pushed on the sliding busbar so that tension was maintained in the heater element. This was necessary to minimize deformation of the heater element caused by thermal expansion. The heater system was mounted to the top wall. The sides of the cubical heated cavity volume adjacent to the heated wall were 4 inches long. The top, bottom and two side walls were steel. The other two side walls were plexiglass to provide parallel transparent surfaces for optical temperature measurement. The two steel side walls were insulated with one inch of a high temperature resistant fiber material and the two plexiglass walls were insulated with 1/4 inch thick glass. The cavity was sealed to provide for accurate air pressure measurement.

Power to the heater element was supplied by two large 12-volt DC batteries connected in parallel. Variable resistance was added in the power circuit as a means of controlling and selecting the desired voltage drop across the heated wall.

A critical design element of the heated cavity apparatus was the heater element. Maintaining a flat heater element surface is essential since significant deformation would cause fluid motion that may alter or mask the thermally-induced motion. An alternative to the stainless steel foil design described above is a very thin, vacuum-deposited layer of conductive material on an insulator as described by Robben, et al. [32]. This has the potential advantage of maintaining a very flat surface throughout heating, but reliability may be reduced because of the possibility of bubbling or peeling of the conductive layer.

The size of the test section was an important variable in the experiment design. Critical factors affected by the test section size include the heat generation rate in the heater element, size and accuracy of the laser deflection measurement, and the one-dimensional flow approximation. The test section size is especially critical for the space experiment design, so further details concerning the size selection criteria are deferred to Chapter 4.

The test section was instrumented with thermocouples and a pressure transducer. Two chromel-alumel (Type K) thermocouples with a 0.015 inch bead on 0.005 inch diameter leads were spot-welded to the heated wall approximately 3/4 inch away from and on opposite sides of the center of the heated wall. It is important that a single bead be attached to the heated wall rather than two separate leads because of the voltage potential across the heated wall. An unbonded strain gage pressure transducer was also connected to the test section on the wall opposite the heated wall. The thermocouples and pressure transducer were connected to a Hewlett Packard 3497A Data Acquisition and Control unit which was connected in series with a Hewlett Packard 9826 computer. The computer was programmed to read the voltage outputs from the thermocouples and pressure transducer and record the elapsed time. The temperatures and pressure were then calculated from calibration curves.

In separate experimental runs, instrumentation was provided to measure the temperature boundary conditions. The temperature of the heated wall was measured in various locations by six thermocouples as shown in Figure 2-4. The 12 thermocouple leads were not insulated because of the high temperature environment. It was, therefore, quite difficult to spot-weld them to the surface so that none of the leads touched each other.

The temperatures of the side walls and bottom wall were measured with fifteen 0.002-inch chromel-alumel thermocouples; nine on the glass side wall, five on the insulated side wall and one on the bottom wall. The locations of the side wall thermocouples are shown in Figure 2-5. The thermocouples on the glass wall were attached with a very small amount of epoxy and shielded from radiation by a thin piece of insulation. The thermocouples on the side wall insulated with fiber material were slightly imbedded in the insulation and also shielded by a thin piece of insulation. The thermocouple on the bottom wall was spot-welded near the center of the wall.

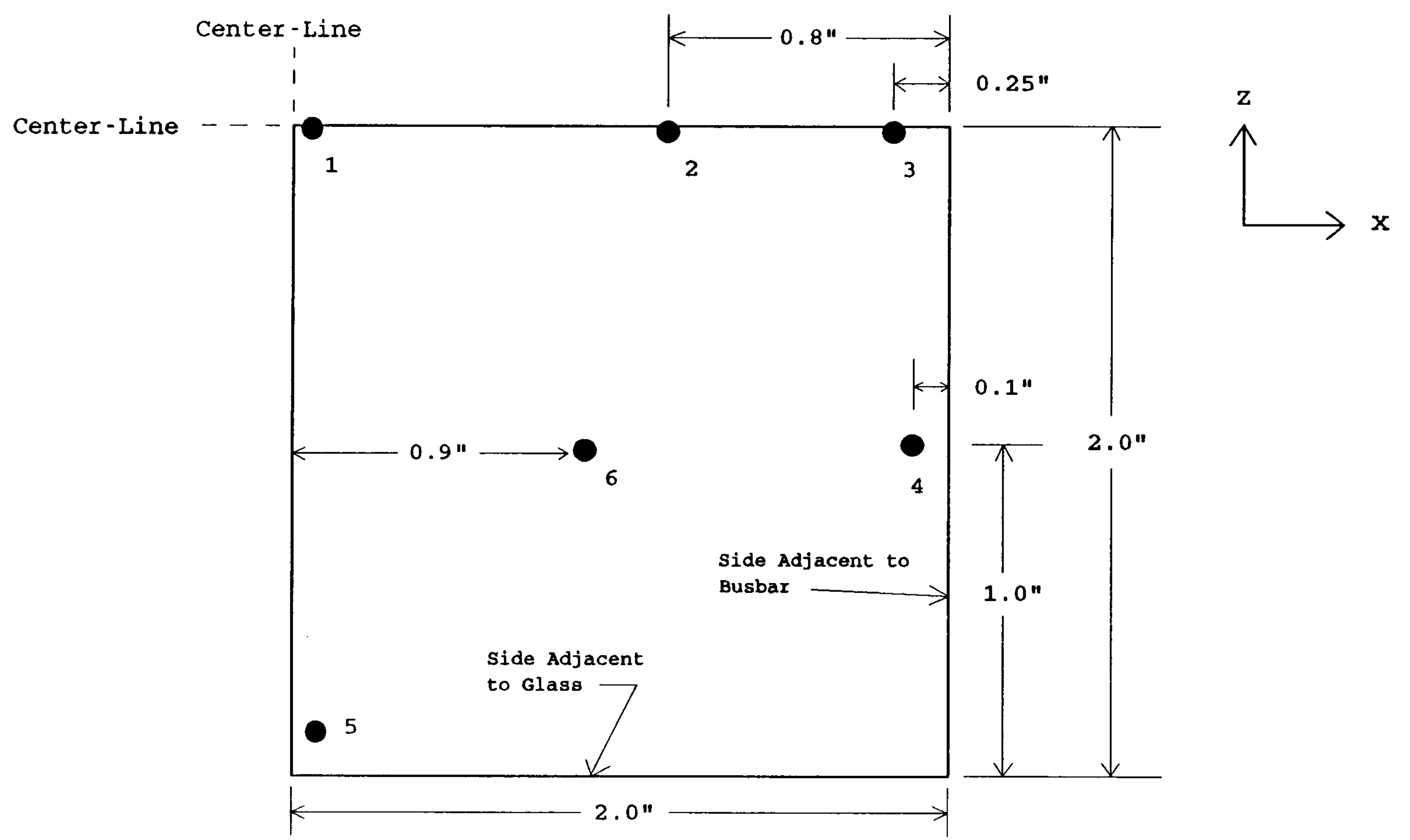


Figure 2-4. Location of Thermocouples on Heated Wall

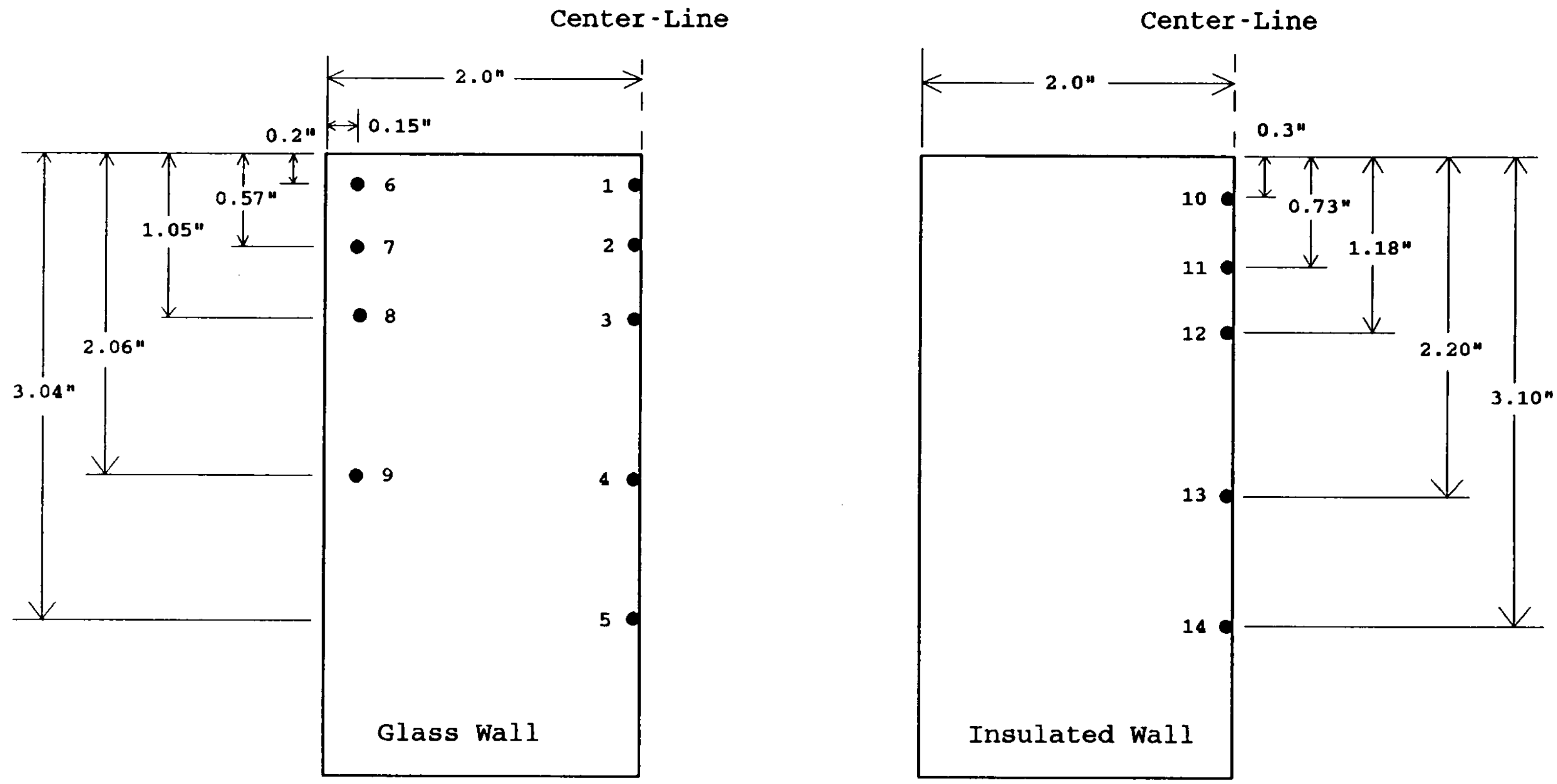


Figure 2-5. Location of Thermocouples on Side Walls

2.3 Optical Temperature Measurement Method

An optical temperature measurement method was selected in this experiment for the measurement of fluid temperature. A schematic of the experimental apparatus, including the optical system is shown in Figure 2-1. The optical system consists of a modified Schmidt-schlieren system which is capable of measuring the air temperature gradient along a cross section of the cavity.

Justification of Optical Temperature Measurement

The optical temperature measurement method is preferred over temperature measurement by thermocouples for several reasons which include:

a) Radiation Error -- Thermocouples with a view of the heated wall absorb a significant amount of radiative energy. Previous experimental work [13] has shown that thermocouples may be heated by radiation to such an extent that they may achieve a higher temperature than the surrounding air. Therefore, large errors may be introduced in the measurement of fluid temperature. Thermocouples could be shielded from incoming radiation, but this would severely interfere with the induced flow convection.

b) Time Response -- The time response of even very small thermocouples (0.0005 inch bead) in the relatively low velocity air flow may introduce large errors in the acoustic time scales of the experiment. Also, very small thermocouples are generally difficult to install and handle especially for fluid temperature measurement. The time response of optical methods is essentially instantaneous.

c) Interference -- thermocouples and thermocouple wiring are, of course, intrusive.

d) Simplicity -- Thermocouples are easier to use except for possible difficulties with mounting very small thermocouples.

Generation of the Laser Sheet

The laser sheet was slanted at an angle from the heated surface in order to measure the laser deflection as a function of distance from the heated wall. The laser sheet passed through the test section as shown in Figure 2-6. With this incident angle and orientation, the beam deflection could be measured at any distance from the heated wall. Assuming that the temperature gradients parallel to the heated wall were zero, the deflection of the beam was away from and perpendicular to the heated wall.

The angle θ of the laser sheet, measured from a perpendicular to the heated wall, was chosen based on the following considerations:

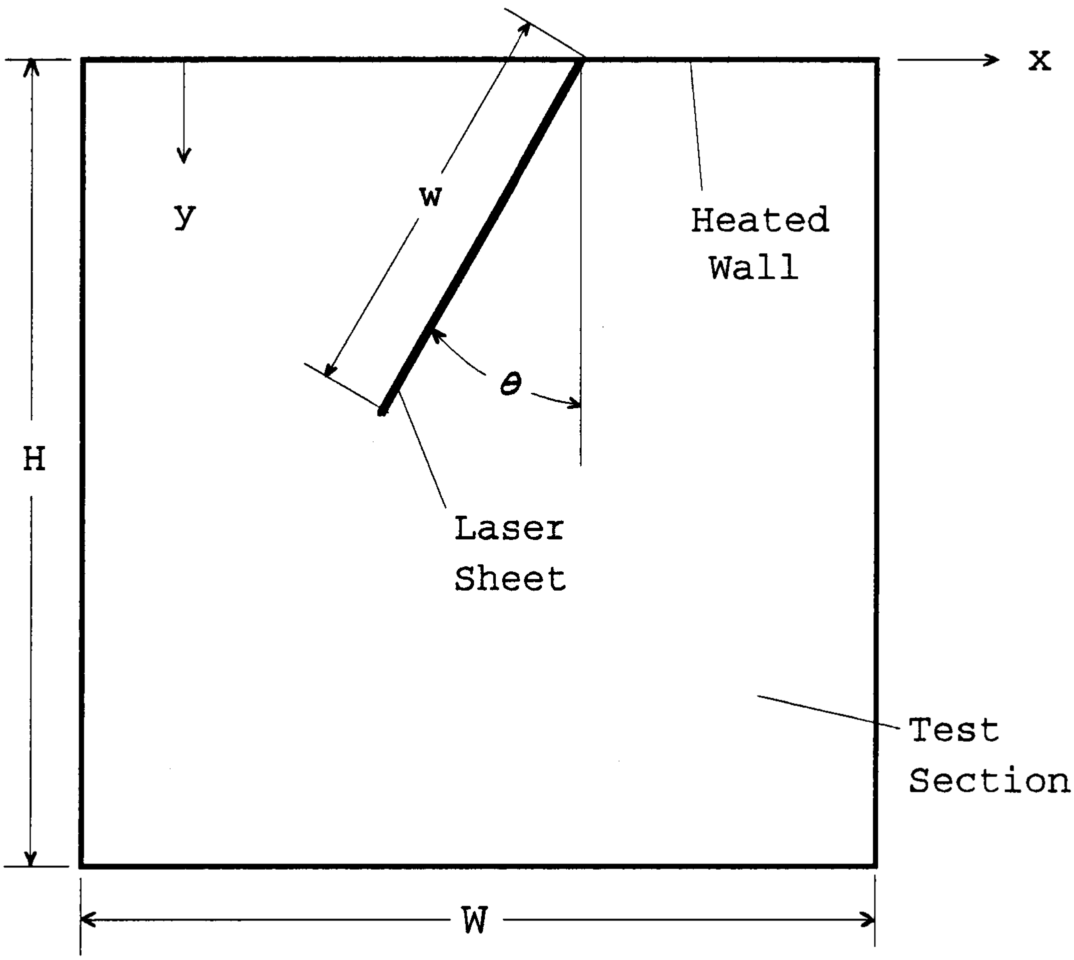


Figure 2-6. Laser Sheet Orientation in Test Section

a) The resolution of distance from the heated wall is proportional to $\tan\theta$. (If $\theta=0$, it would not be possible to determine the amount of deflection at a given distance from the wall at all.)

b) The maximum range of distance from the heated wall that can be measured is $w(\cos\theta)$, where w is the expanded width of the laser sheet.

c) Minimizing three dimensional (wall) effects requires the laser to be at a distance of $w(\sin\theta)/2$ from the centerline or at $(W - w(\sin\theta))/2$ from the side walls (W is defined in Figures 2-2 and 2-6).

Clearly, (a) improves with larger θ and (b) and (c) improve with smaller θ. The actual size of θ was determined by balancing the above considerations with regard to the size of the test section and the width, w, of the laser sheet. For a 4x4 inch test section and w=2.0 inches, the value of $\theta=37°$ was found suitable and selected.

The laser sheet was produced as shown in Figure 2-7 using a 5 milliwatt red He-Ne laser with a beam diameter of 0.83 mm and divergence of 0.97 mrad. The beam expander/collimator consisted of two spherical lenses with focal lengths of 36 mm and 125 mm. The focal point of each lens was at the same location, so the resulting beam was parallel with a diameter of

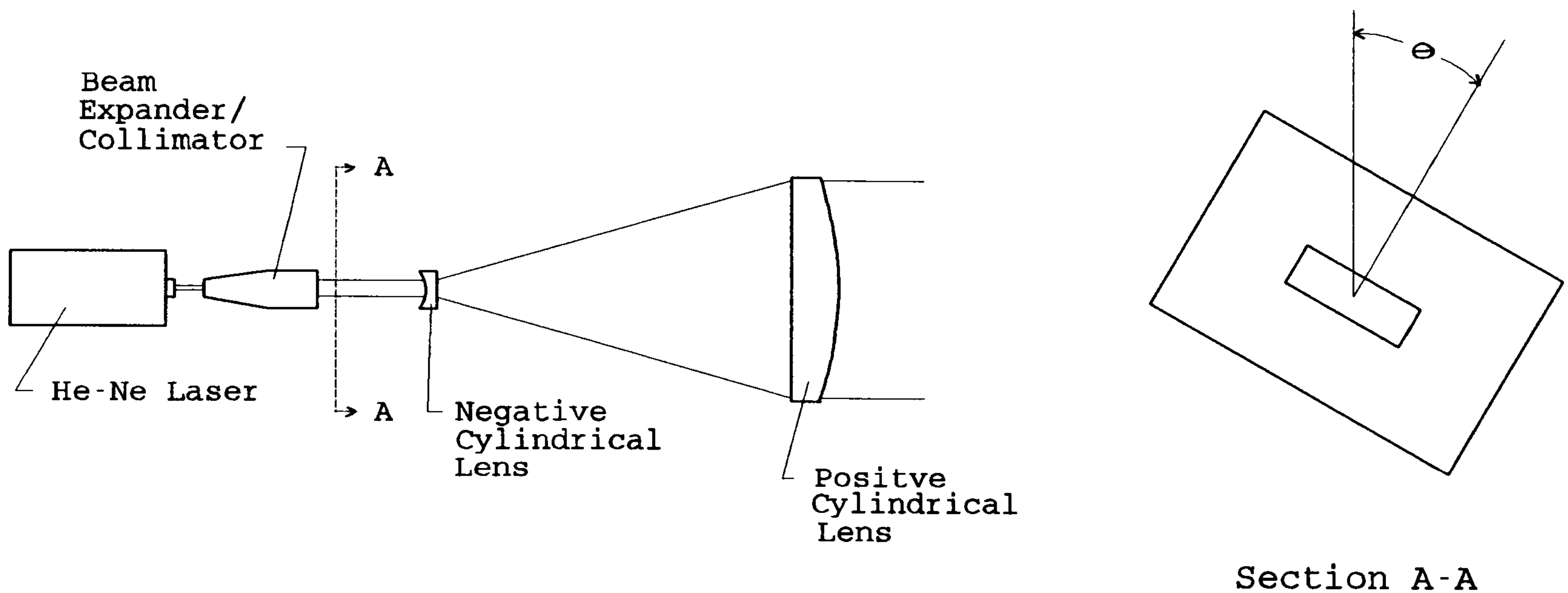


Figure 2-7. Schematic of Optical System for Generating Laser Sheet

$$d_2 = d_1 \left| \frac{f_{s2}}{f_{s1}} \right| \tag{2-1}$$

where d_1 is the original laser beam diameter and d_2 is the diameter after the lenses. The focal lengths f_{s1} and f_{s2} are for the first and second spherical lenses, respectively. This results in d_2=2.88 mm. Diffraction effects cause an initially parallel beam to diverge. The effect decreases with increasing diameter, so expanding the beam reduces its divergence. The net result is a narrower beam image at the screen.

The collimated, circular beam passes through a negative (concave) cylindrical lens with a focal length of -6.35 mm. This lens causes the beam to diverge in one direction only so that an expanding laser sheet is created. This expanding sheet then passes through a positive (convex) cylindrical lens with a focal length of 300 mm. The focal point of the second (positive) cylindrical lens was slightly farther from the test section than that of the first (negative) cylindrical lens so that the resulting laser sheet expanded slightly after exiting the lens system with a width, w, given by

$$w = d_2 \left| \frac{f_{c4}}{f_{c3}} \right| \tag{2-2}$$

where f_{c3} and f_{c4} are the focal lengths of the third and fourth (the two cylindrical) lenses, respectively. This results in w=136 mm, however, since the height of the second cylindrical lens was only 50 mm, effectively, w=50 mm. The thickness t of the laser sheet is $t=d_2$=2.88mm. The slight expansion of the width of the sheet had a negligible effect on w at the test section, but provided an elongated image at the screen for better resolution of distance from the heated wall. Both of the cylindrical lenses were tilted to an angle of $\theta=37°$ relative to the test section to produce the laser sheet oriented as shown in Figure 2-6.

After the test section the beam was reflected off a series of three plane mirrors, as shown in Figure 2-1, so that the total length z_s from the test section to the screen was 72.2 feet. This long path length past the test section was necessary to provide the needed sensitivity in the measurement of beam deflection. A video camera was used to record the image of the laser sheet on a white screen. The video camera took 30 images per second for a time least count of 0.033 seconds. Fixed reference markers were used on the screen to make it easier to identify the undeflected position of the laser image during video image processing.

The Relationship between Laser Deflection and Temperature

If measuring the deflection of the laser sheet is to be useful, its relationship to the temperature of the air must be known. The deflection of the beam is the result of refraction, so it is the index of refraction (or its derivative) that is actually measured. For a homogeneous transparent medium, the index of refraction n is primarily a function of density as given by the Lorenz-Lorentz relation:

$$\frac{1}{\rho}\left(\frac{n^2-1}{n^2+2}\right) = C \qquad (2\text{-}3)$$

where C is a constant. When n is close to 1 (which is true for air), this reduces to the Gladstone-Dale equation

$$C = \frac{n-1}{\rho} = \frac{n_o - 1}{\rho_o} \qquad (2\text{-}4)$$

where n_o and ρ_o are the index of refraction and density, respectively, at standard temperature. The Gladstone-Dale constant C depends on the particular gas and the wavelength of the light source used in the optical system. The wavelength dependence is, however, small.

The refraction of light is actually a result of changes in the index of refraction, so from Equation (2-4)

$$\frac{\partial n}{\partial y} = \frac{(n_o - 1)}{\rho_o} \frac{\partial \rho}{\partial y} \tag{2-5}$$

where y is in the direction perpendicular to the light source as shown in Figure 2-8. For an ideal gas $\rho = P/RT$, and if P is not a function of y (note that acoustic pressure disturbances are small compared to the mean pressure as shown in Appendix C)

$$\frac{\partial \rho}{\partial y} = -\frac{P}{RT^2} \frac{\partial T}{\partial y} \tag{2-6}$$

Combining (2-5) and (2-6) gives

$$\frac{\partial n}{\partial y} = -\frac{(n_o - 1)}{\rho_o} \frac{P}{RT^2} \frac{\partial T}{\partial y} \tag{2-7}$$

If the deflections are small and $\partial n/\partial z = \partial n/\partial x = 0$ and $n \approx 1$, the path of a light ray through the test section is given by [31]

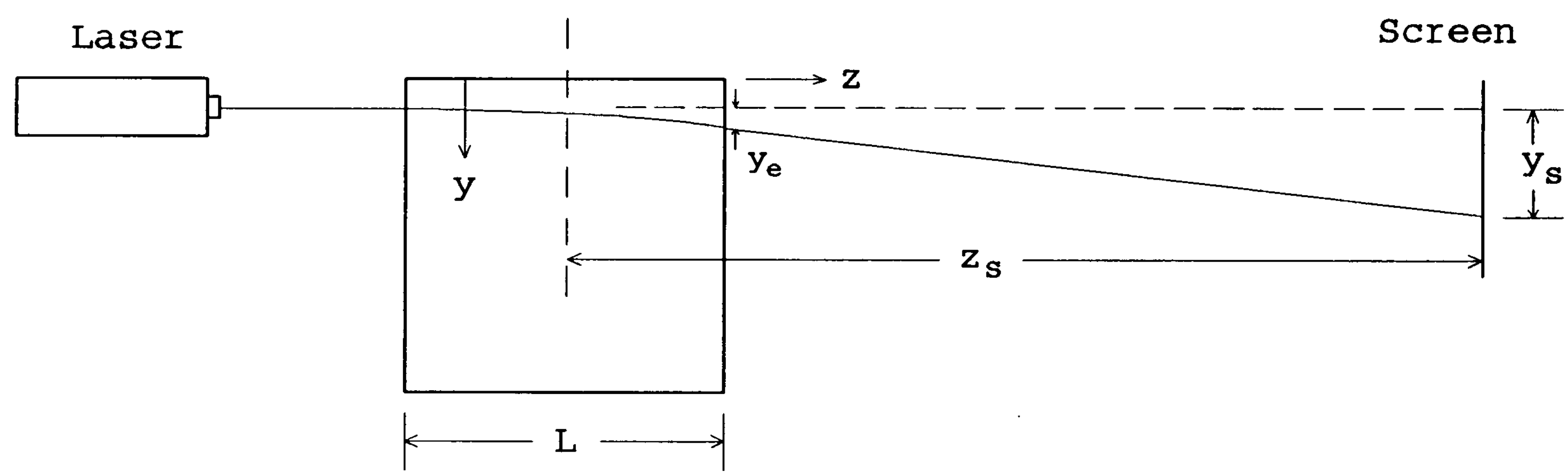


Figure 2-8. Schematic of Laser Ray Passing Through Test Section

$$\frac{\partial^2 y}{\partial z^2} = \frac{\partial n}{\partial y} \tag{2-8}$$

These assumptions can be easily justified for air and with the selection of L (length of the test section in the z direction) equal to 4 inches. If in addition the deflection is small, $\partial n/\partial y$ is not a function of z, so integration over the length of the test section gives the slope of the beam at the exit of the test section:

$$\left.\frac{\partial y}{\partial z}\right)_e = \frac{\partial n}{\partial y} L \tag{2-9}$$

Integrating again gives

$$y_e = \frac{\partial n}{\partial y} \frac{L^2}{2} \tag{2-10}$$

After exiting the test section the light travels in a straight line, so with the slope given in Equation (2-9) one obtains

$$y_s = y_e + \frac{\partial n}{\partial y} L\left(z_s - \frac{L}{2}\right) \tag{2-11}$$

where y_e is the laser beam deflection at the exit of the test section, z_s is the length of the path to the screen, and y_s is the deflection at the screen as shown in Figure 2-8. Equations (2-7), (2-10), and (2-11) can now be combined to give

$$y_s = -\frac{(n_o - 1)}{\rho_o} \frac{PLz_s}{RT^2} \frac{\partial T}{\partial y} \tag{2-12}$$

or

$$\frac{\partial T}{\partial y} = -\frac{\rho_o}{(n_o - 1)} \frac{RT^2 y_s}{PLz_s} \tag{2-13}$$

Equation (2-13) relates the temperature and its derivative to the deflection at the screen in terms of other known quantities (recall that the mean pressure, P, is directly measured as a function of time). With the wall temperature known from thermocouple measurements, Equation (2-13) can be used along with measured laser deflection varying with y to determine temperature as a function of y at any given time. The numerical integration of Equation (2-13) is discussed in Chapter 3.

2.4 Procedure

Experiments were run with four different levels of power applied to the heated wall which are referred to as (a) **low**, (b) **medium low**, (c) **medium high**, and (d) **high** power. Three runs were made for each power level, and the results were averaged before presentation. Quantitative data for the power and heated wall temperatures are presented in Chapter 3.

The data acquisition computer program provided the thermocouple temperatures, the pressure and the elapsed time since the beginning of data acquisition. The actual start time of the run could be determined from the beginning of the rise in temperature. The video camera also provided an on-tape elapsed time from the instant the camera was turned on. Again, the actual start time could easily be determined from the initiation of the deflection of the laser.

The next task was to determine the laser deflection from the video tape. Using an IBM AT compatible PC with a PCVision Plus image processing card and Werner Frei Associates image processing software along with a high resolution television monitor, it was possible to obtain a series of digitized images at various selected times from the video tape. These images were then enhanced to improve the visibility of the laser sheet image, and the location of the laser sheet was then measured with a mouse pointer. Deflection measurements were taken at times of 0.5, 1.0, 2.0, 5.0, and 10.0 seconds after initiation of heating. The following steps were taken to get the detailed measurements:

a) The number of pixels per length in each direction was determined directly from marks on the screen that were spaced six inches apart.
b) The image was zoomed to include only the deflected beam. With the ratio of the size of the original to the zoomed image known, the number of pixels per length was determined in each direction. There were approximately 80 pixels/inch in the z direction (perpendicular to the deflection direction) and approximately 60 pixels/inch in the y direction (along the deflection direction).
c) The coordinates of the top edge of the undeflected laser sheet image were established before heating. The top edge of the image was the edge adjacent to the heated wall.
d) The frames were counted at 30 frames/second to locate the frame at the selected time of interest.
e) The y and z pixel location of the top edge of the deflected laser sheet was recorded for each selected time of interest.

From this recorded set of information, the deflection versus distance from the heated wall for a given time was determined as described in Chapter 3.

Separate runs were made to get temperature boundary conditions. One run at each power level was made to measure the temperature of the heated wall at various locations. These runs also measured the voltage drop across the heated wall and the current to obtain the power dissipation of the heated wall. Another run at each power level was made to measure the side wall temperatures at various locations. These runs, of course, did not involve the laser or video camera.

CHAPTER 3

EXPERIMENTAL ANALYSIS AND RESULTS

This chapter describes the results of the experimental investigation. The data that is presented includes power consumption, heated wall temperature, pressure, and laser deflection; all of these were measured directly in the experiment. These directly measured quantities are analyzed and the data is processed to obtain transient temperature profiles for the enclosed air. The experimental analysis and the resultant air temperature profiles along with error analysis are presented in this chapter. Finally, the chapter is concluded by presenting the results of flow visualization and transient one-dimensional air velocity measurement within the cavity.

3.1 Power Consumption

The voltage drop and current through the heater element were measured for each of the four power level runs performed in the experiment. The current was highest at the beginning of each run before leveling off after about 1 second. This is the result of increasing resistance of the heated wall and the external resistor with temperature. Figure 3-1 shows a plot of current as a function of time for each power level. The voltage drop across the heater element decreased only slightly (±5%) with time. The nominal voltage for each power level was

(a) Low power 3.7 volts

(b) Medium low power 5.7 volts

(c) Medium high power 7.0 volts

(d) High power 10.4 volts.

The resulting power consumption at each level is shown in Figure 3-2. After an initial drop in power (a maximum of approximately 15%) that occurred within one second, the power remained steady with values that ranged from approximately 300 Watts for the low power case to 2000 Watts for the high power case.

3.2 Heated Wall Temperature and Measured Pressure

The HP computer was programmed to read the voltage output from the two thermocouples attached to the heated wall and the pressure transducer. The output consisted of the two thermocouple temperatures, pressure and the time for each reading. Each reading, however, required approximately 0.15 seconds, so the actual time associated with each value varied slightly from the recorded time. The actual time for each value was determined by linear

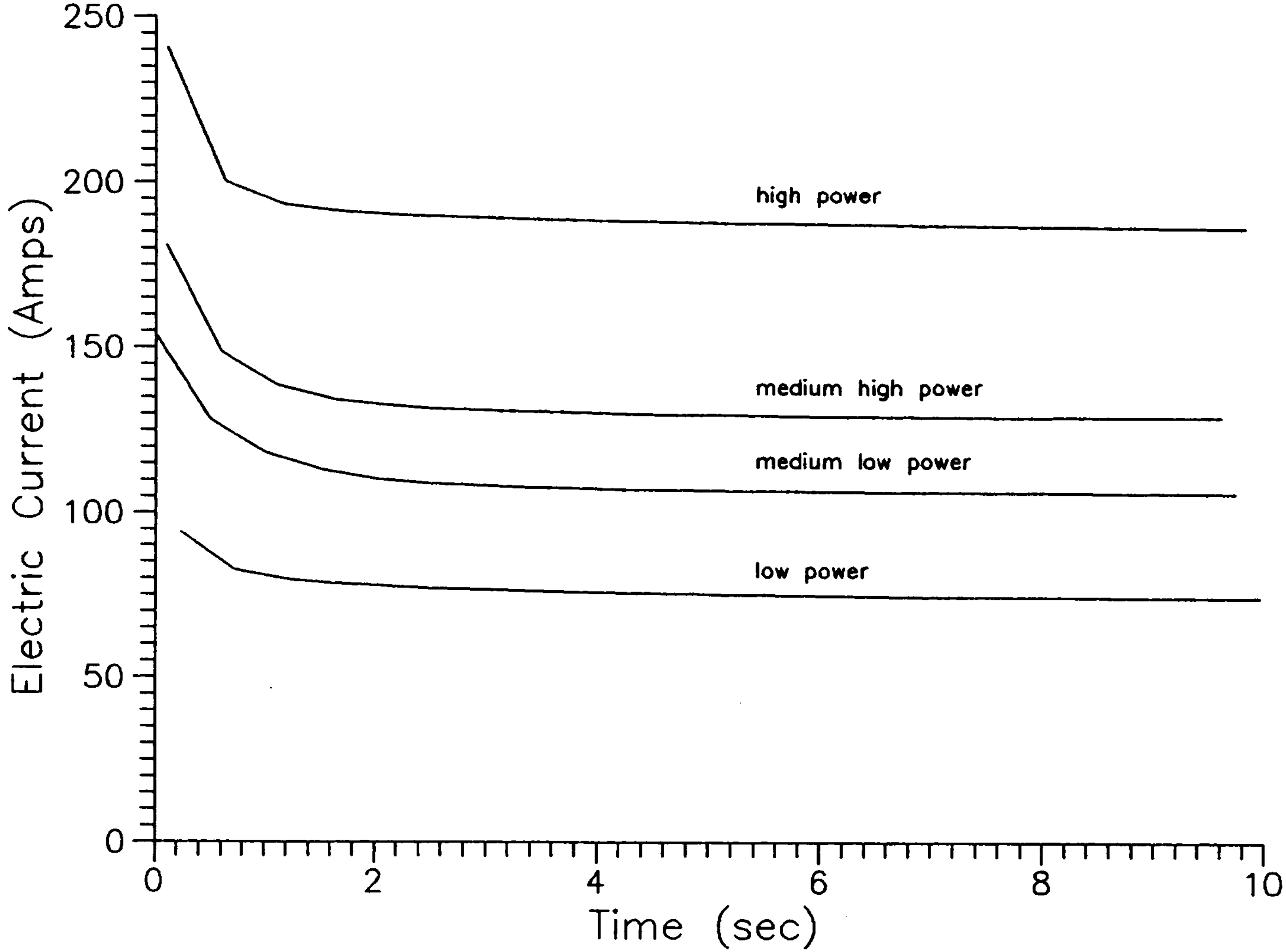


Figure 3-1. Current as a Function of Time for Each Power Level

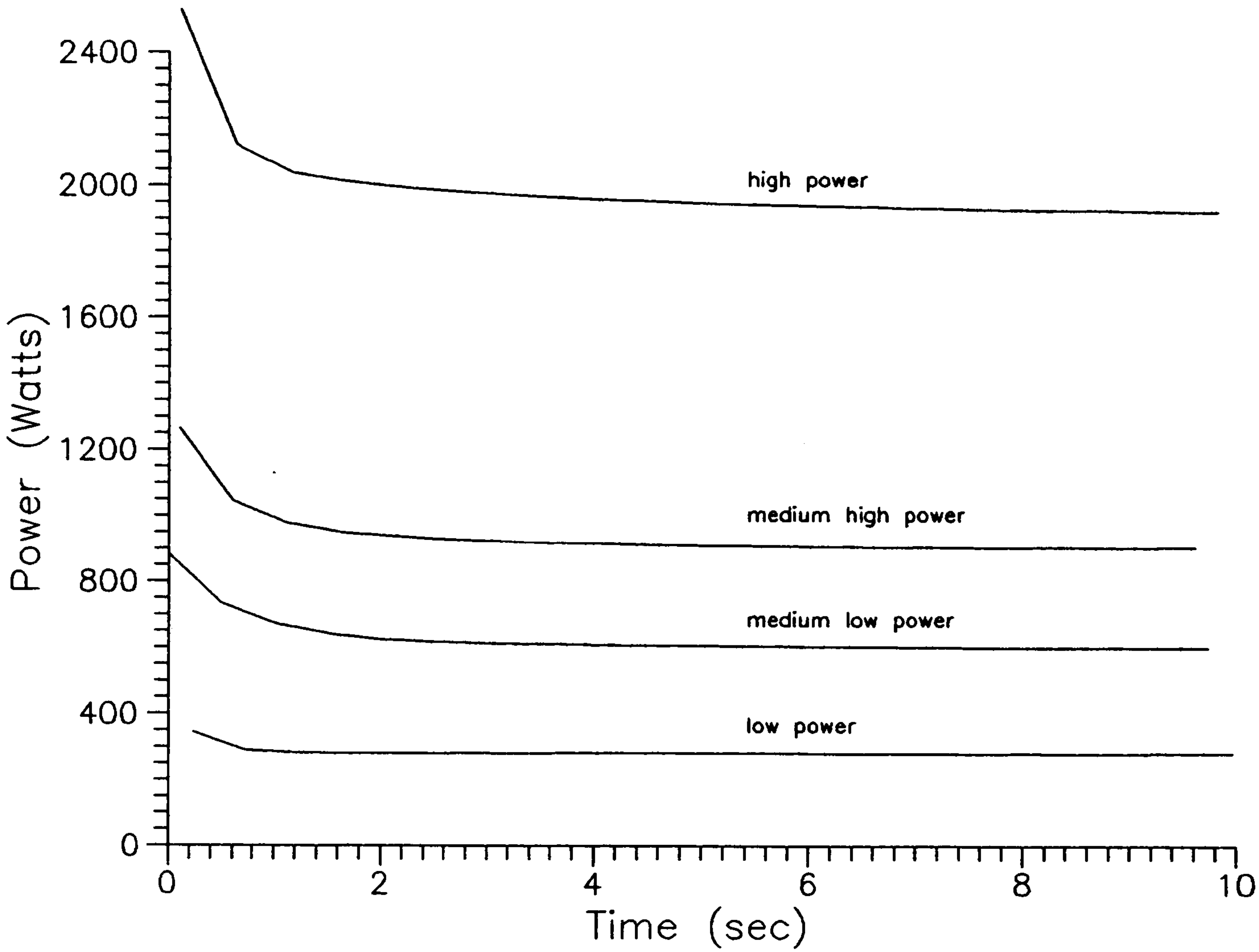


Figure 3-2. Power Consumption as a Function of Time for Each Power Level

interpolation between the recorded times. The data for each run at a given power level and for a given thermocouple were very repeatable (maximum difference in temperature rise less than 2%), however, one of the thermocouples consistently read a temperature 50 to 60°F above the other after 2 or 3 seconds. This is less than 9% of the total temperature rise for the low power case and a smaller percentage for higher power levels. It can be shown that the difference between the thermocouple temperature and the local heater element temperature is negligible. The average of the two thermocouple readings was taken as the temperature of the center of the heated wall.

Since the laser sheet traverses the length of the test section between the glass side walls, it essentially averages conditions along its path. Therefore, it is necessary to determine the average temperature of the heated wall between the glass walls. Figures 3-3 through 3-6 show the temperatures of the heated wall at each power level in the locations shown in Figure 2-4. These temperatures were normalized by dividing by the center point temperature and curve-fitted to a two dimensional quadratic polynomial. The average temperature along the centerline between the glass side walls was then determined and is shown in Figure 3-7. The initial rate of temperature rise of the heated wall was approximately 1700°F per second for the high power case.

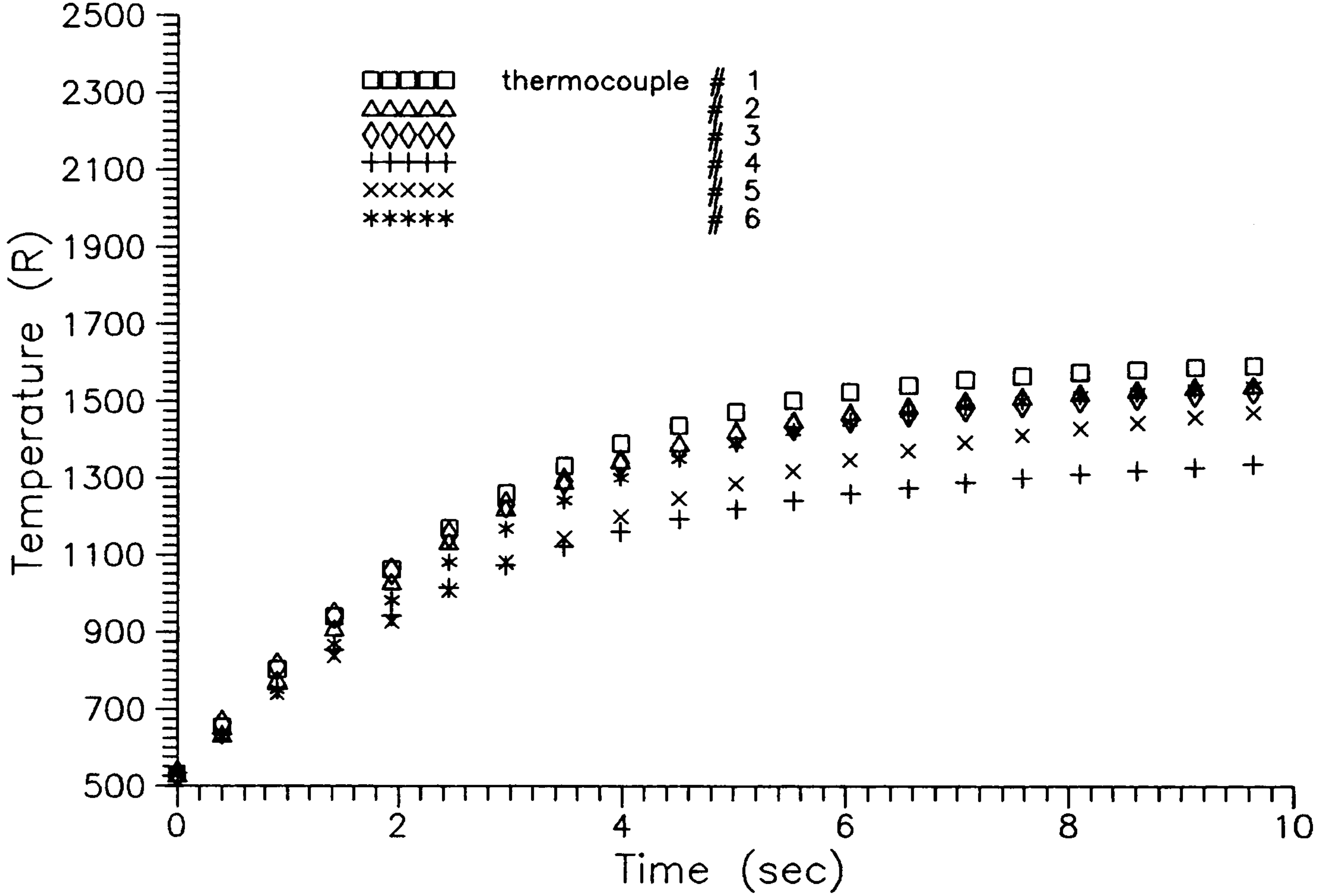


Figure 3-3. Heated Wall Temperature at Various Locations as a Function of Time for Low Power

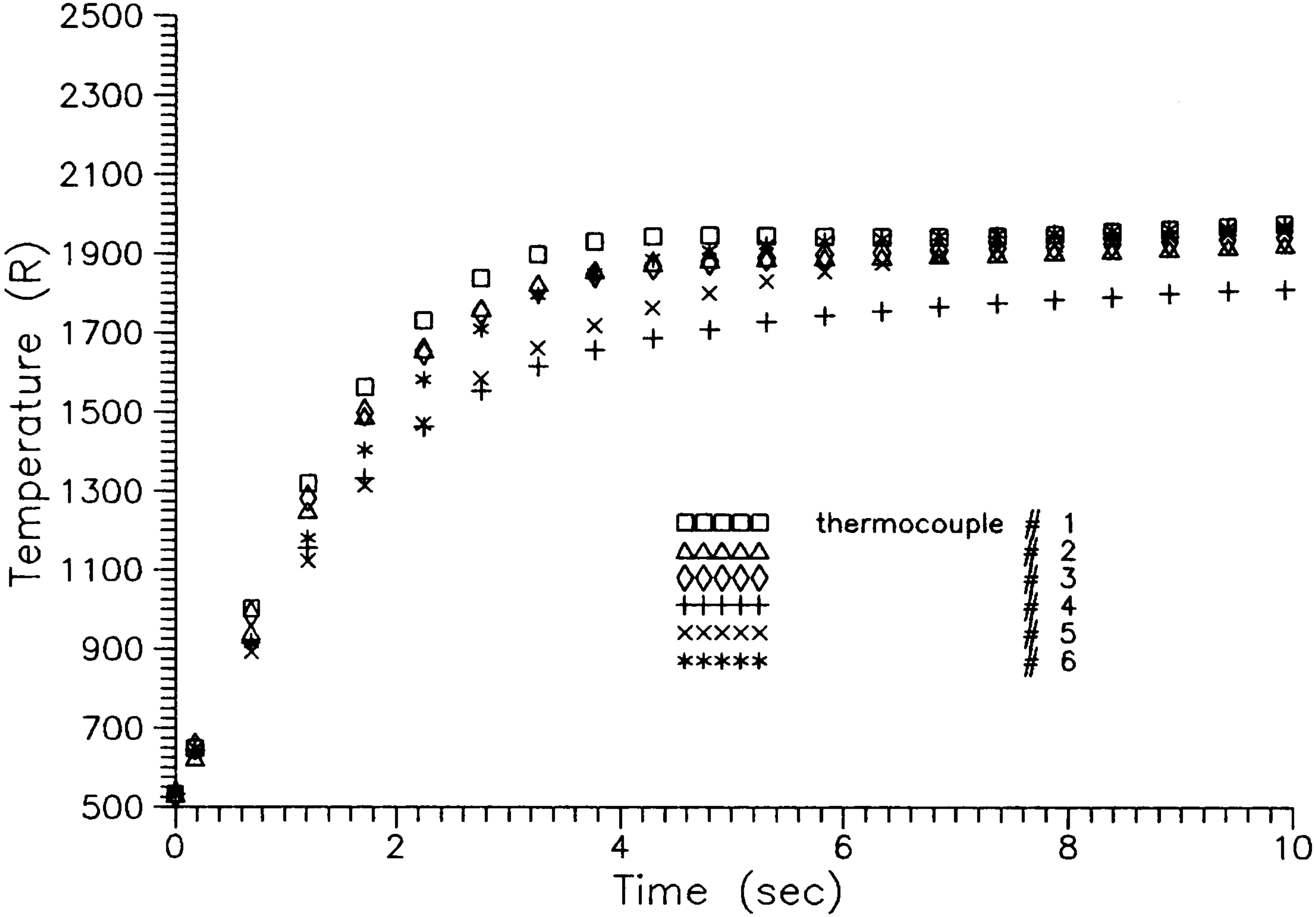


Figure 3-4. Heated Wall Temperature at Various Locations as a Function of Time for Medium Low Power

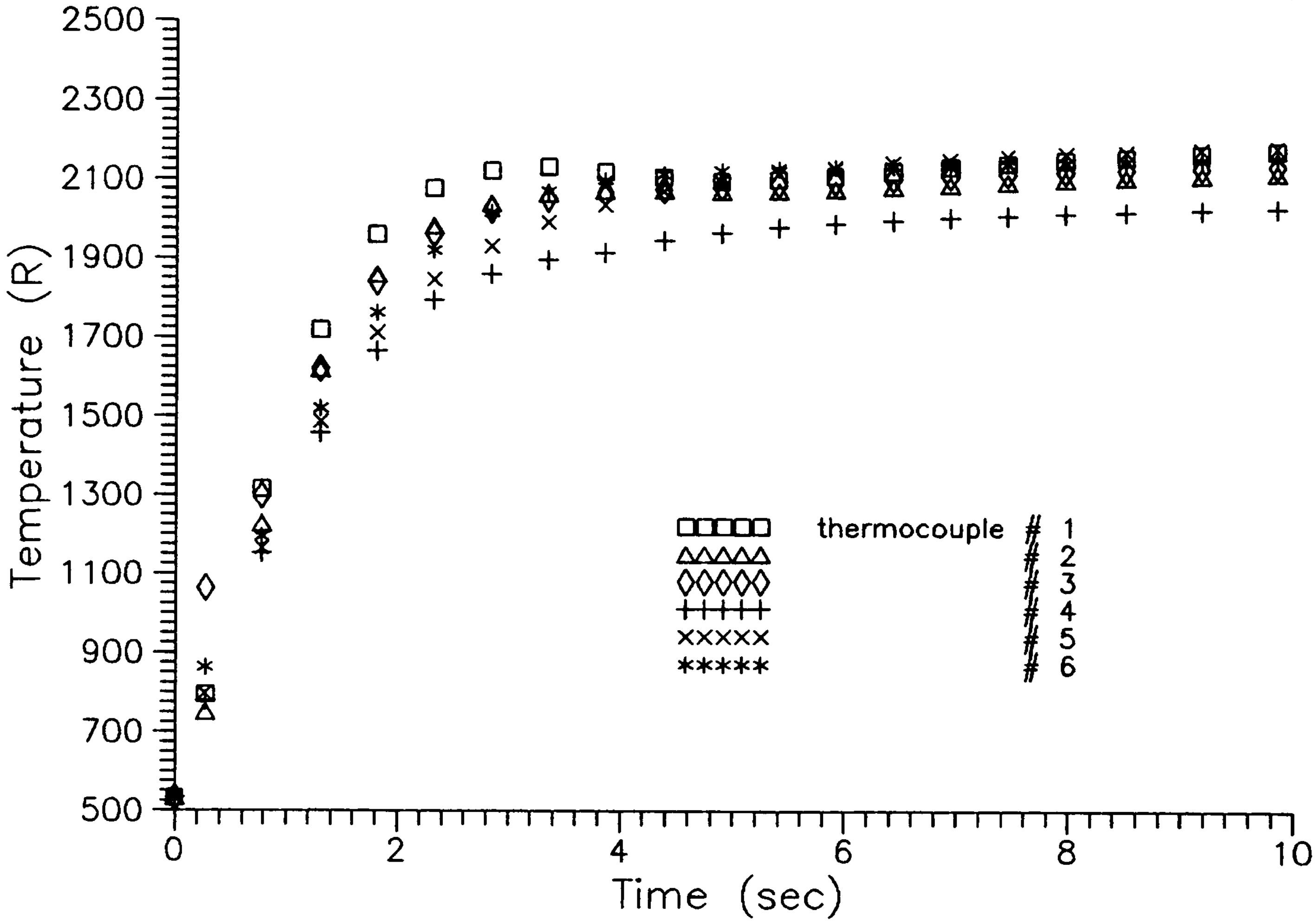


Figure 3-5. Heated Wall Temperature at Various Locations as a Function of Time for Medium High Power

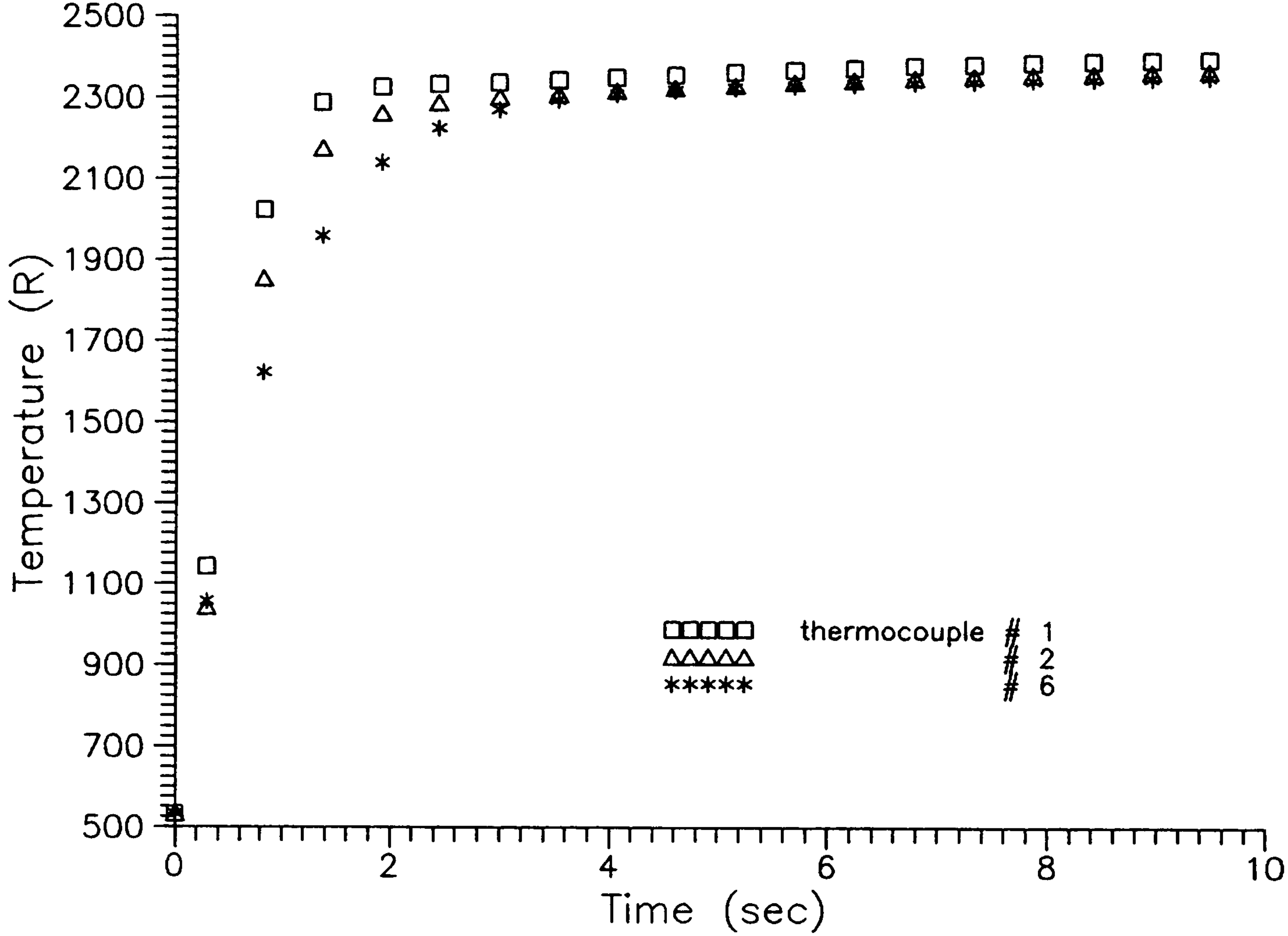


Figure 3-6. Heated Wall Temperature at Various Locations as a Function of Time for High Power

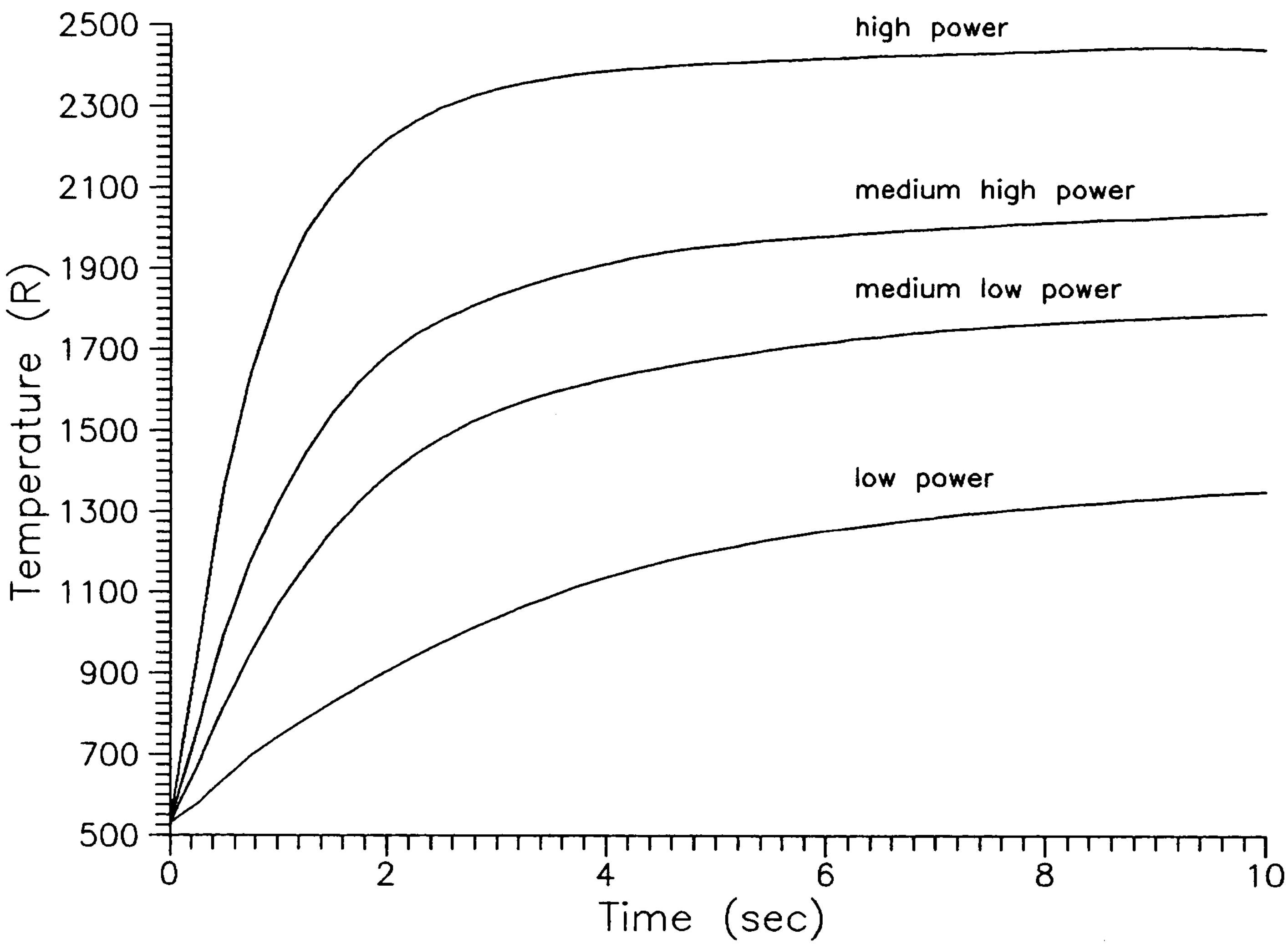


Figure 3-7. Heated Wall Temperature Averaged Along the Centerline for Each Power Level

The measured pressure for each run at a given power level varied by less than 1% of the maximum pressure rise. The measured pressure for each power level is shown in Figure 3-8. These pressure values are used in the integration of Equation (2-13) for the determination of the temperature profiles.

3.3 Laser Deflection

The raw data taken manually from the image processor consisted of the coordinates (y,z) of pixels that formed the contour of the laser sheet profile on the screen. A schematic of the deflected beam shapes at various times for the medium high power case is shown in Figure 3-9. The coordinate data was first converted from pixels to inches before any further manipulations.

The shape of the undeflected beam was saved for comparison with the deflected beam shapes at the selected time slices. Though the undeflected beam would ideally be a straight line, some curvature was present due to errors in lens alignment and imperfect mirrors. This was, however, not a problem since the actual shape of the undeflected beam was used. The deflection at a given point is simply $y_d - y_u$, where y_d and y_u are the deflected and undeflected points, respectively, for a given z. The

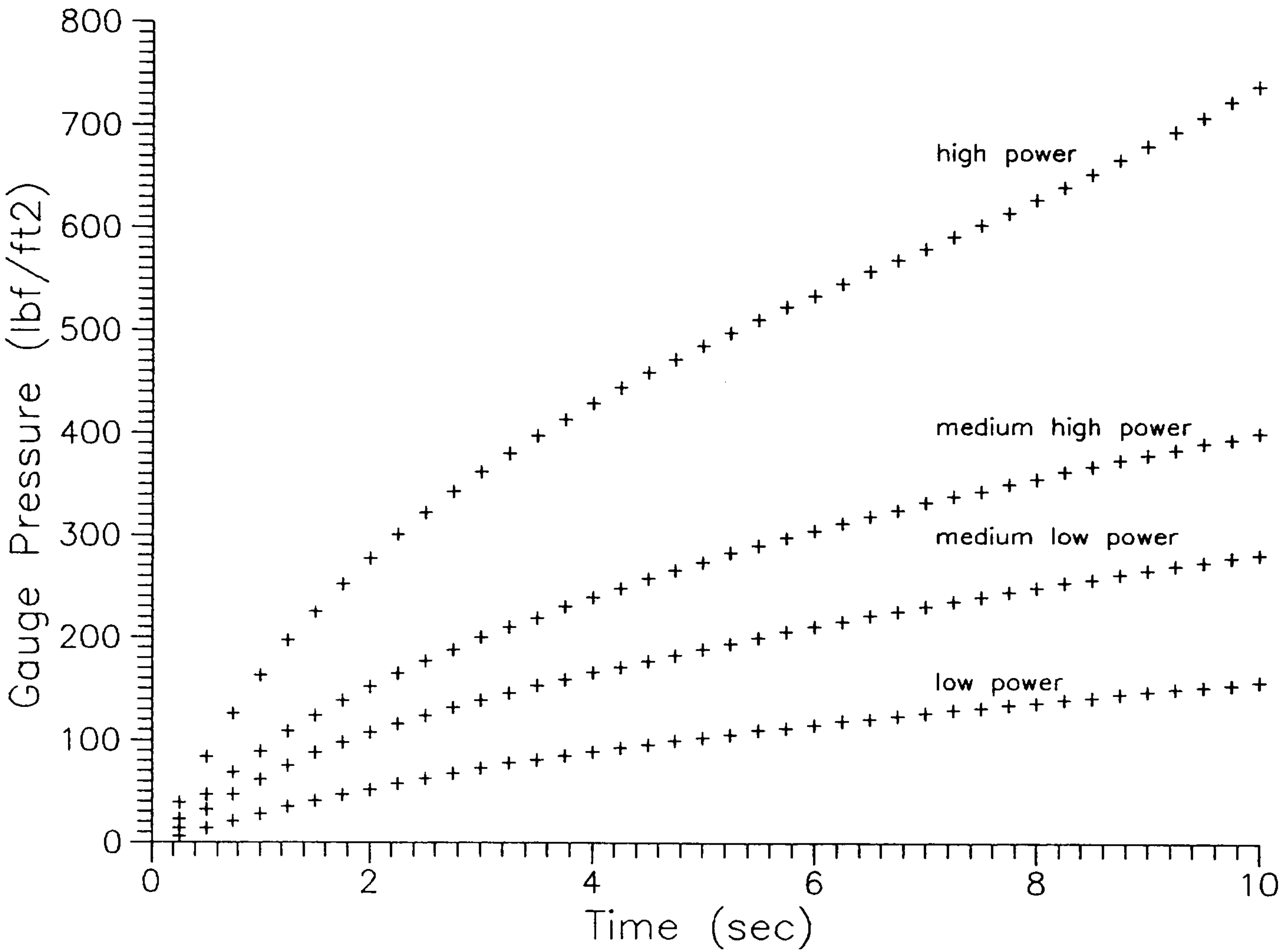


Figure 3-8. Measured Pressure as a Function of Time for Each Power Level

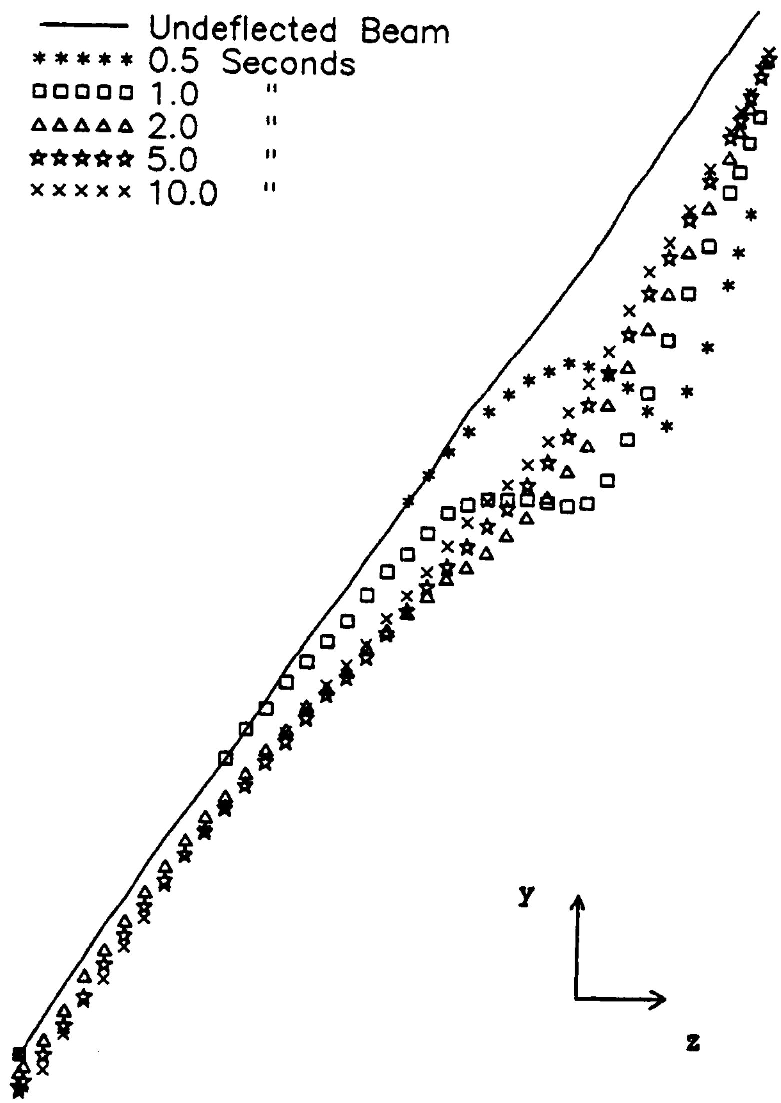


Figure 3-9. Deflected Laser Beam Shape at Various Times for Medium High Power

distance from the heated wall associated with each point is

$$y = \frac{w}{w_s} \sqrt{(y - y_o)^2 + (z - z_o)^2} \cos\theta \qquad (3\text{-}1)$$

where w and w_s are the laser sheet width at the test section and at the screen, respectively, and θ is the angle of inclination of the laser sheet (see Figure 2-6). The square-root quantity represents the distance along the undeflected beam image on the screen. The laser deflection at times 0.5, 1.0, 2.0, 5.0, and 10.0 seconds are presented in Figures 3-10 through 3-13 as a function of the distance from the heated wall at each power level. The maximum deflection is observed to decrease in magnitude and to occur at a greater distance from the heated wall as time increases.

3.4 Temperature Profile

As suggested by Equation (2-12), the deflection is proportional to the temperature gradient (heat flux) and inversely proportional to the square of the local air temperature. Equation (2-13) can be used to relate the laser deflection to the temperature of the air. All factors in Equation (2-13), except T and $\partial T/\partial y$, are known

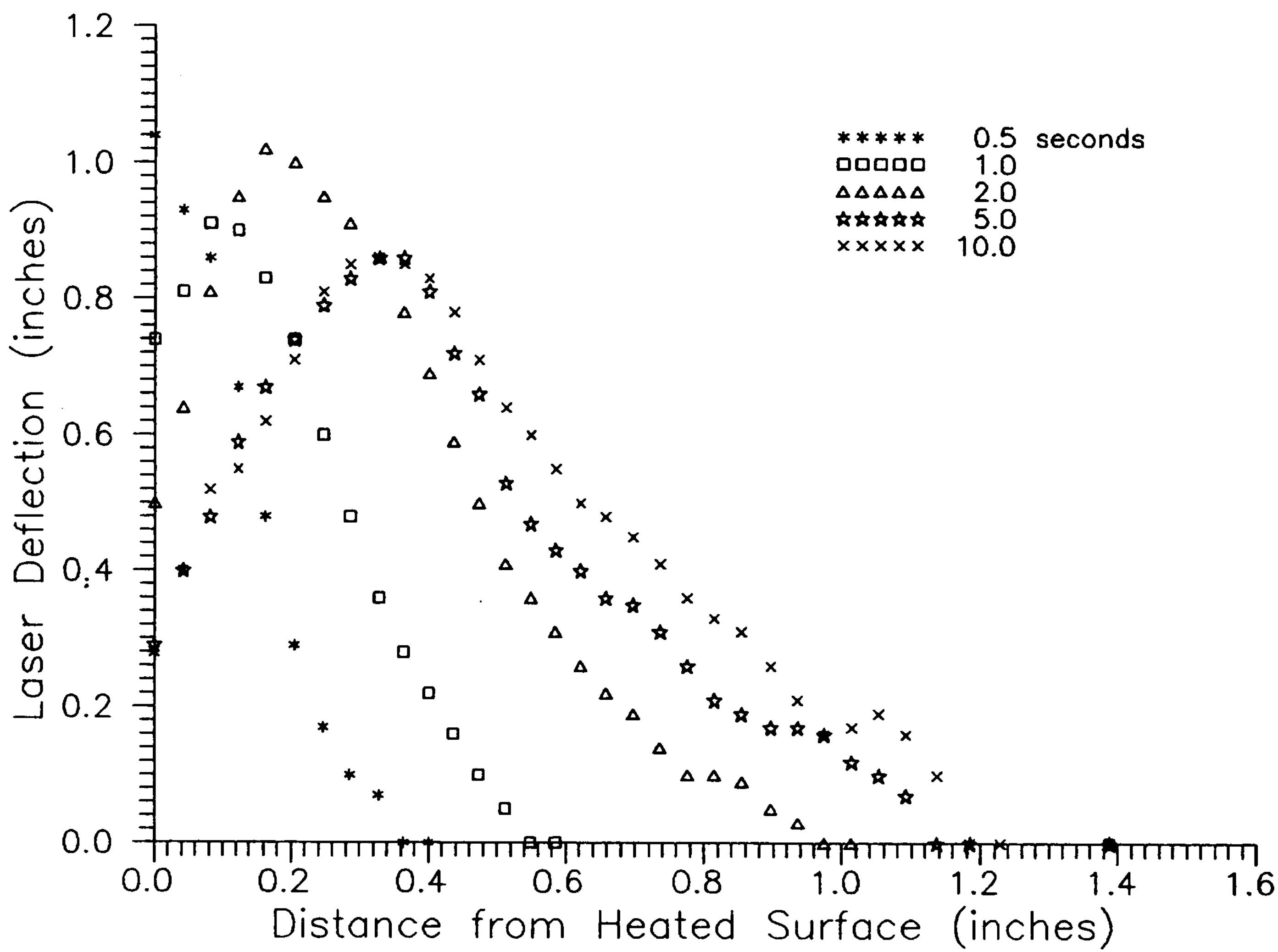


Figure 3-10. Laser Deflection as a Function of Distance from the Heated Wall for Low Power

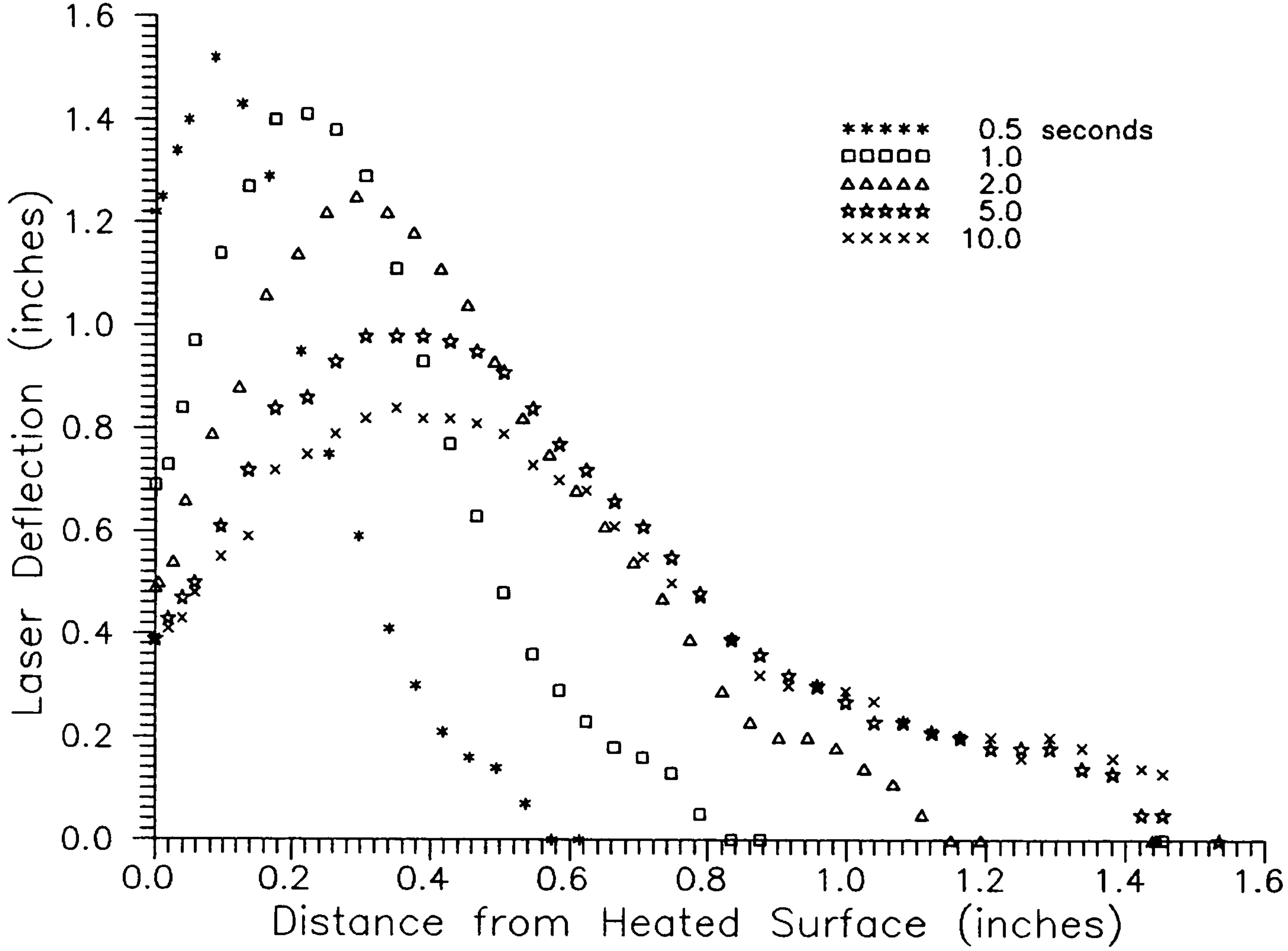


Figure 3-11. Laser Deflection as a Function of Distance from the Heated Wall for Medium Low Power

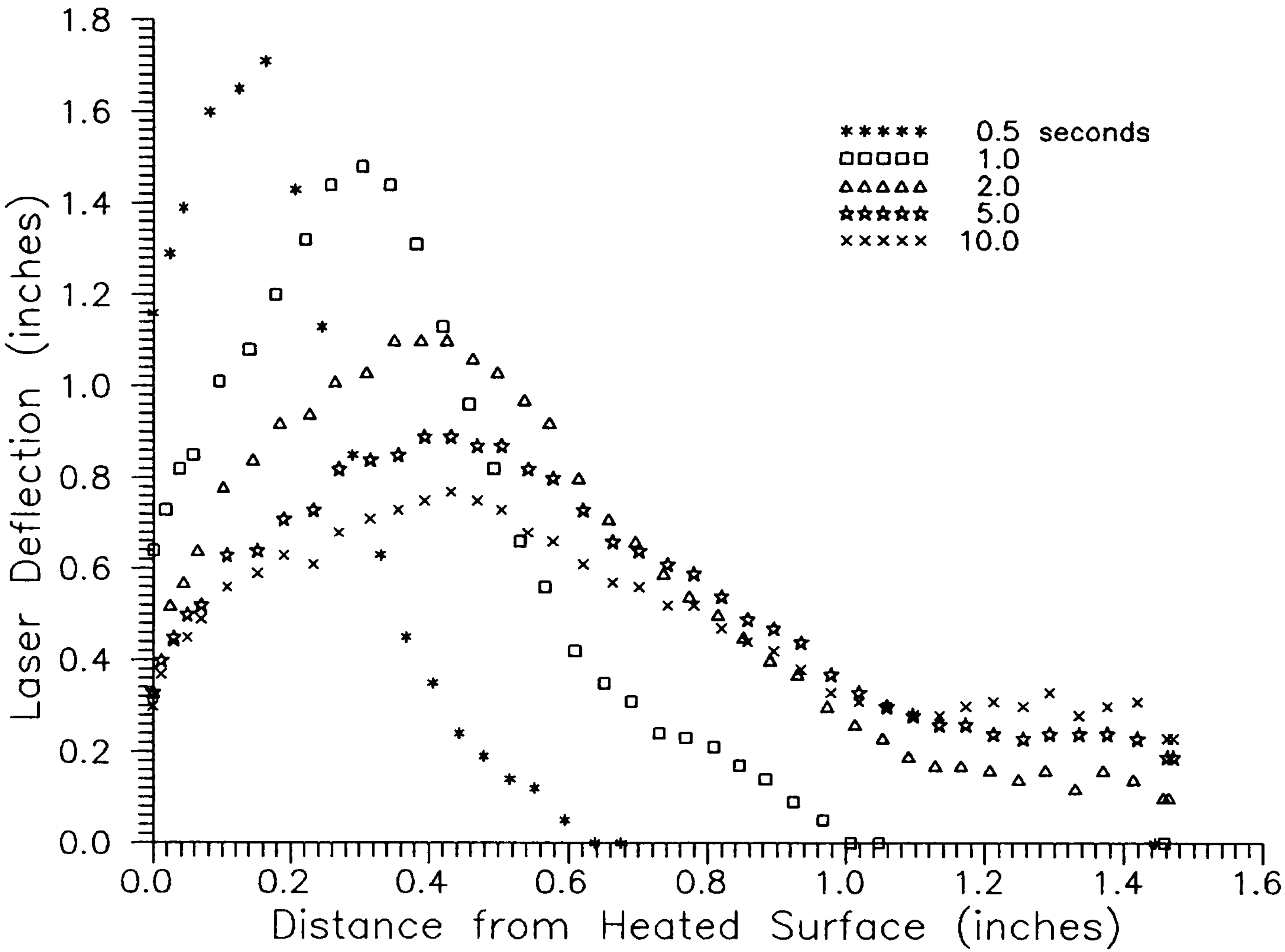


Figure 3-12. Laser Deflection as a Function of Distance from the Heated Wall for Medium High Power

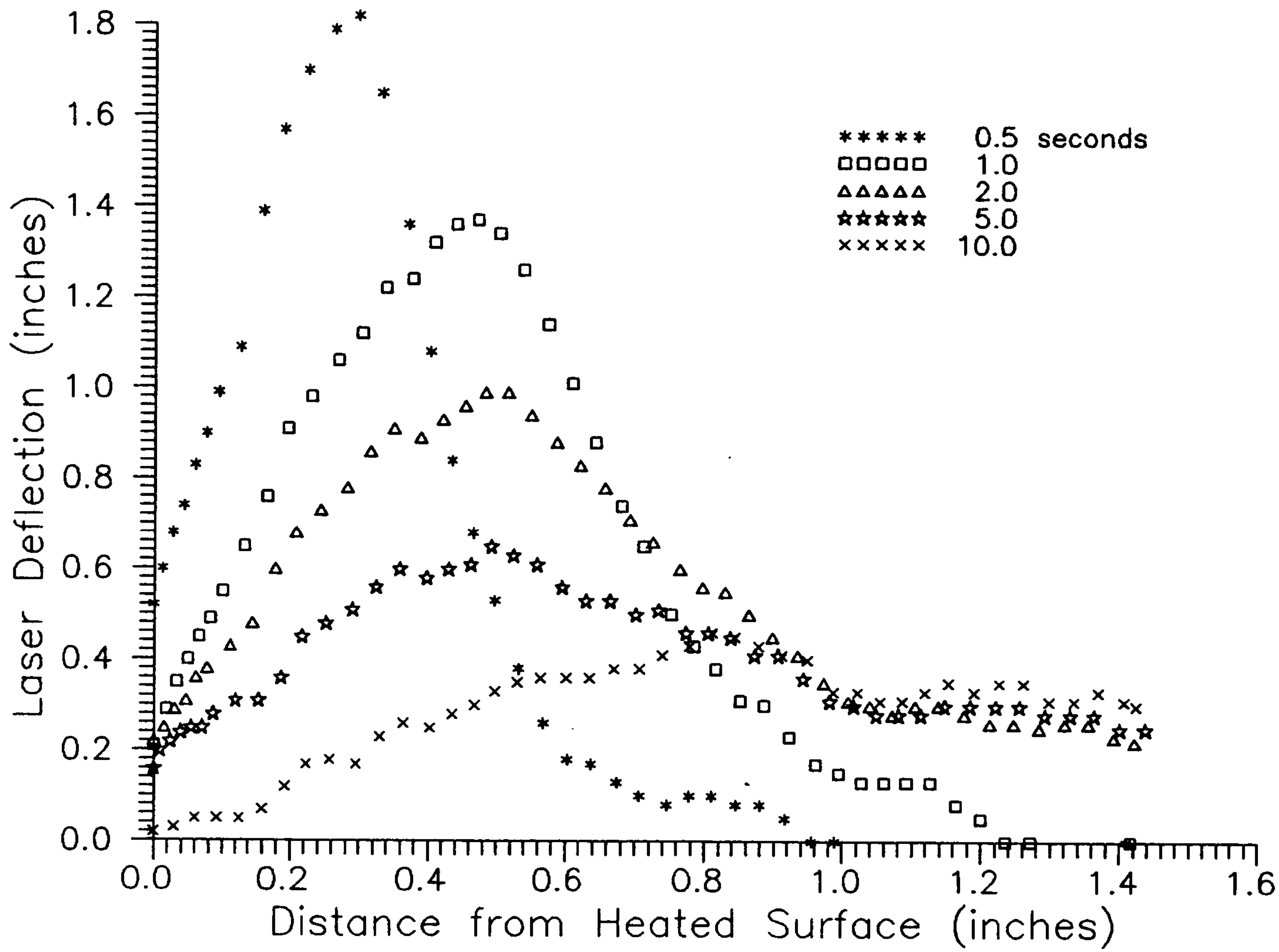


Figure 3-13. Laser Deflection as a Function of Distance from the Heated Wall for High Power

quantities, including the pressure. The latter was determined as a function of time and the measured laser beam deflection $y_s(y,t)$. The temperature of the heated wall T_w is required as a boundary condition for Equation (2-13) and has been measured as a function of time. Equation (2-13) can be written as

$$\frac{dT}{dy} = KT^2 y_s \tag{3-2}$$

since, for a given time, T is assumed to be only a function of y. From Equation (2-13), K is

$$K = -\frac{\rho_o R}{(n_o - 1) PLz_s} \tag{3-3}$$

The temperature can be found from a Taylor series expansion of the temperature profile. The following equations can be written directly

$$T_i = T_{i-1} + \Delta x_i T'_{i-1} + \frac{\Delta x_i^2}{2} T''_{i-1} + \cdots$$

$$T_{i-1} = T_i - \Delta x_i T'_i + \frac{\Delta x_i^2}{2} T''_i + \cdots \tag{3-4}$$

$$T_i = T_{i+1} - \Delta x_{i+1} T'_{i+1} + \frac{\Delta x_{i+1}^2}{2} T''_{i+1} + \cdots$$

$$T_{i+1} = T_i + \Delta x_{i+1} T'_i + \frac{\Delta x_{i+1}^2}{2} T''_i + \cdots$$

where each i represents a discrete location along y, and primes indicate derivatives with respect to y.
Equations (3-4) can be combined to give

$$T_i = \frac{1}{2}(T_{i-1} + T_{i+1}) + \frac{\Delta x_i}{4}(T'_{i-1} + T'_i) - \frac{\Delta x_{i+1}}{4}(T'_i + T_{i+1}) + O(\Delta x^2) \tag{3-5}$$

Equation (3-5), along with Equations (3-2) and (3-3) can be solved for each T_i. Since $T' = f(T^2)$, the solution must be found iteratively. The $O(\Delta x^2)$ terms were found to be negligibly small and were dropped from the expansion. The above formulation (Equations (3-4) and (3-5)) includes derivative approximations from both sides of T_i. This prevents accumulated error which may occur with a solution that marches in one direction from the known derivative at the wall.

The measured temperature profile extends only as far from the wall as the laser beam; a distance of approximately 1.4 inches. If dT/dy was not zero at the "end" of the measured temperature profile, the profile was extended. The point where dT/dy=0 is

$$y_b = y_e + \left[\frac{dT/dy}{d^2T/dy^2} \right] \Bigg)_e \qquad (3\text{-}6)$$

where the subscript b indicates the edge of the temperature boundary and the subscript e indicates the end of the laser beam. The temperature is extended linearly so that

$$T_b = T(y_b) = T_e + (y_b - y_e)\left(\frac{1}{2}\right)\frac{dT}{dy}\Bigg)_e \qquad (3\text{-}7)$$

The temperature profiles at various specified times are shown for each power level in Figures 3-14 through 3-17. It is important to note that neither T_c nor T_b are specified in the solution of the temperature profile. The temperature T_c far from the wall, or core temperature, is the initial temperature T_o plus a small temperature rise due to pressure work. The core temperature should increase slightly with increasing time. It is observed that the experimentally measured temperature T_b does increase with time. The proximity of T_b to T_c is an indication of the accuracy of the results. The apparently excessive increase in T_b for the high power case (Figure 3-17) may be the result of slight charring of the fiber insulation material since a small amount of smoke was observed in the test section.

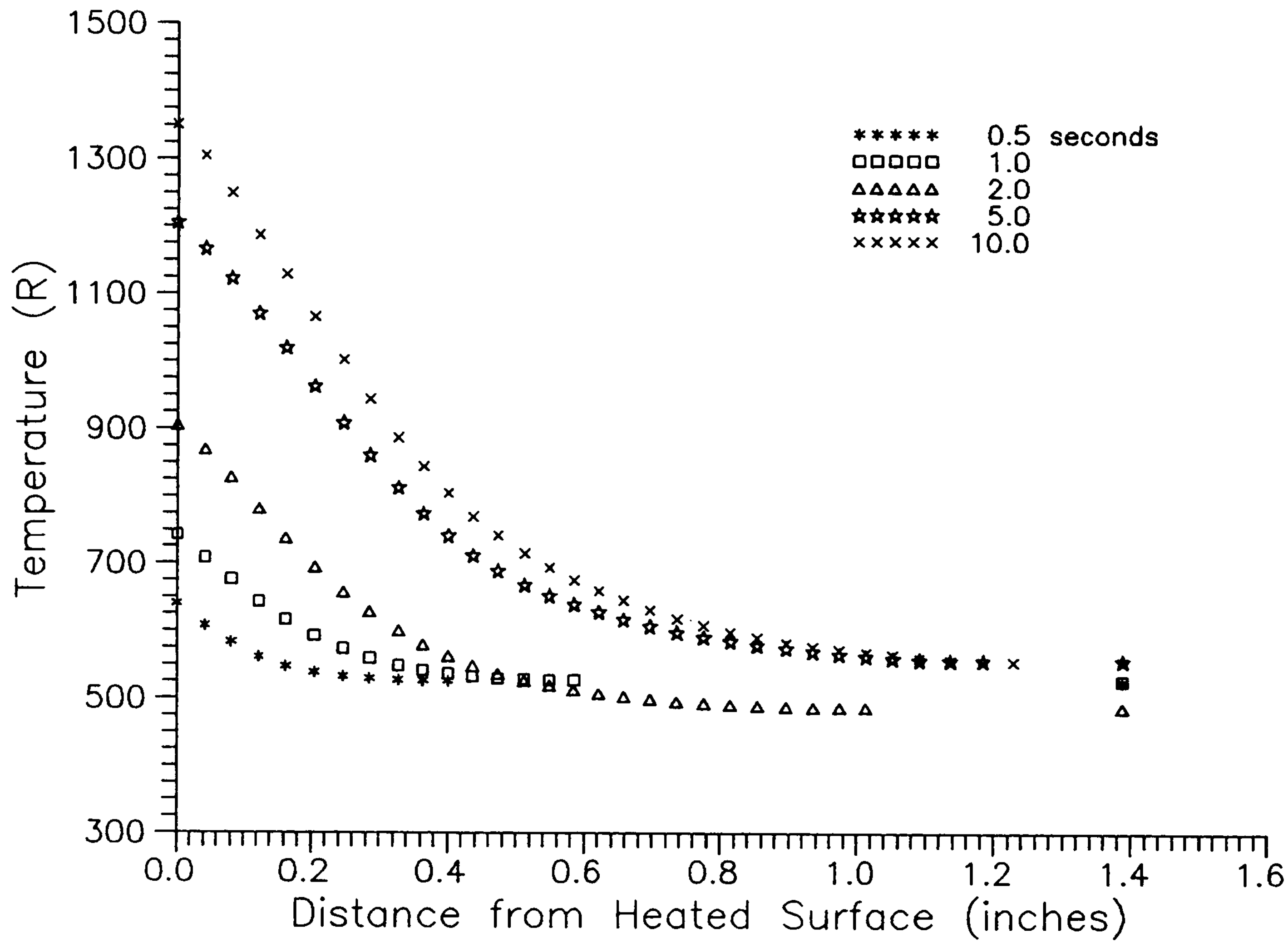


Figure 3-14. Temperature as a Function of Distance from the Heated Wall for Low Power

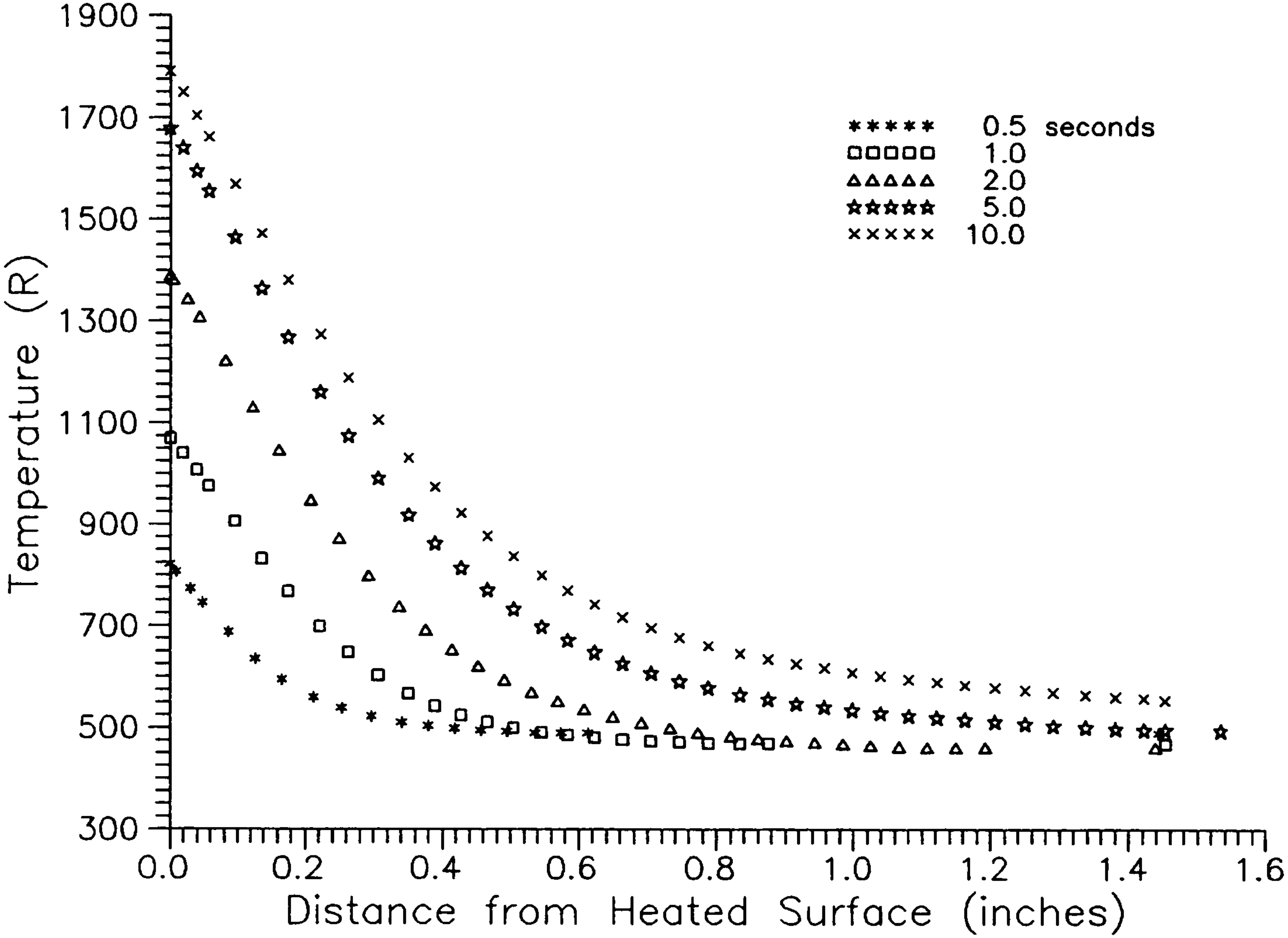


Figure 3-15. Temperature as a Function of Distance from the Heated Wall for Medium Low Power

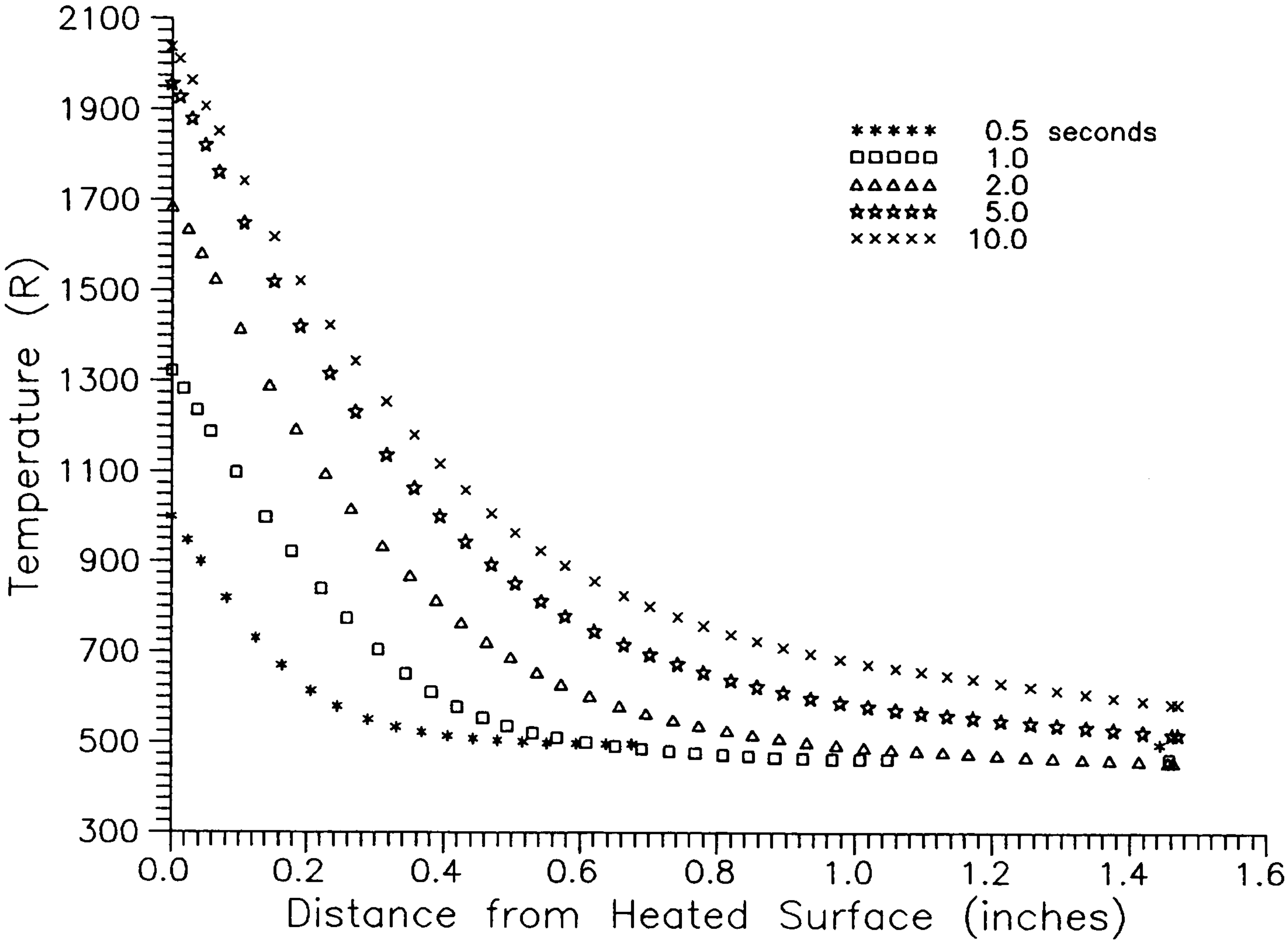


Figure 3-16. Temperature as a Function of Distance from the Heated Wall for Medium High Power

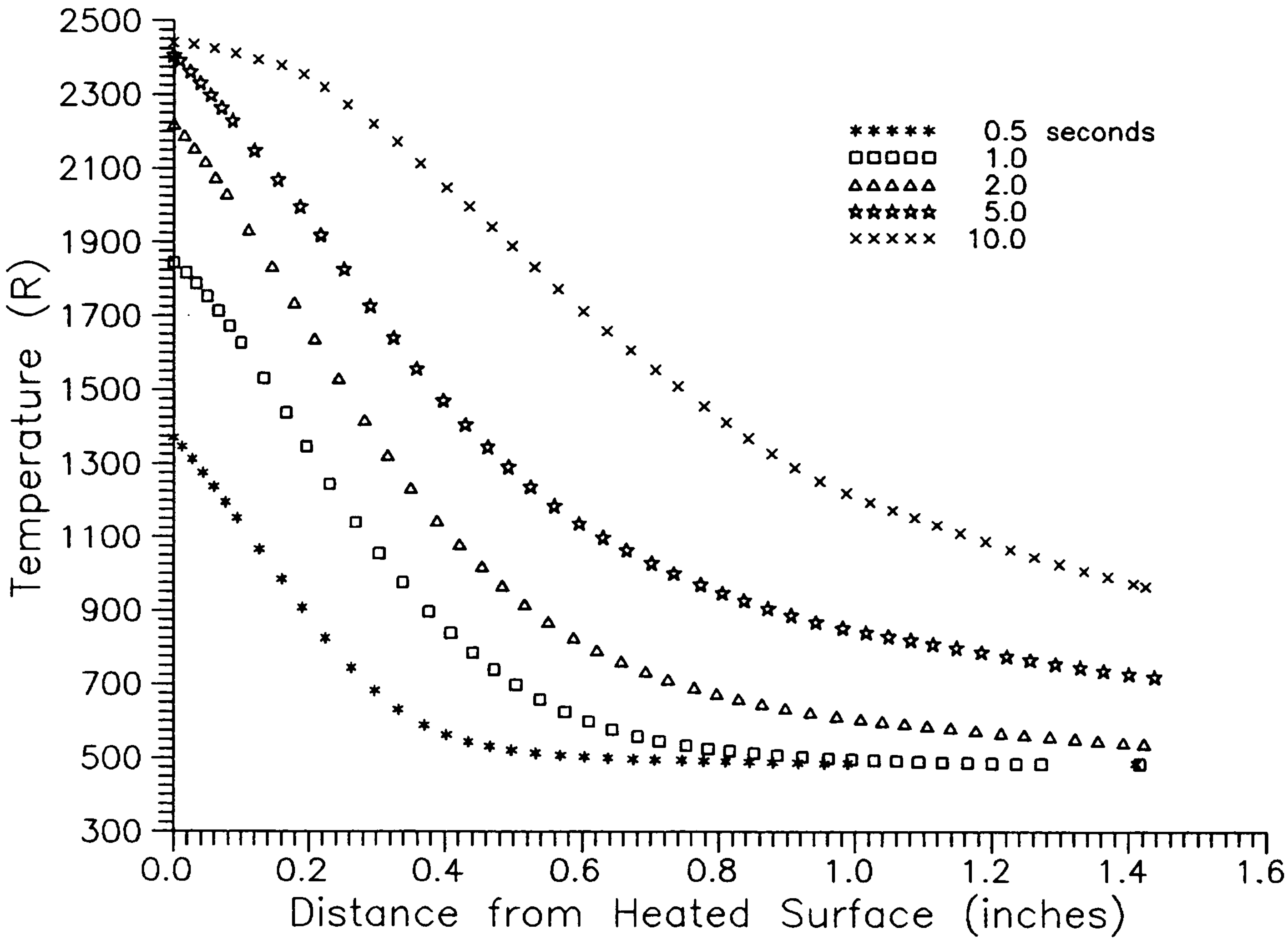


Figure 3-17. Temperature as a Function of Distance from the Heated Wall for High Power

An analysis was performed to determine the maximum error in the measured temperature profile. The details of the analysis are presented in Appendix D. Several sources of potential error have been identified including measurement of the laser sheet deflection, heated wall temperature, test section length, and heated wall flatness. The uncertainty for the above quantities was determined and the effects on the temperature profile were calculated. The worst-case uncertainty in the measurement of the fluid temperature was found to be approximately 8%. This experimental uncertainty is primarily due to the uncertainty in the flatness of the heated wall.

3.5 Temperature Correction

Since T_b cannot be expected to be exactly equal to T_c, the temperature profile was adjusted as follows

$$T_{a_i} = T_i - (T_b - T_c)\left[\frac{(T_w - T_i)}{(T_w - T_b)}\right] \qquad (3\text{-}8)$$

where T_a is the adjusted temperature. The core temperature can be chosen such that the measured gage pressure equals the gage pressure calculated from the temperature profile.

The absolute pressure can be determined from the temperature profile as

$$P = \frac{R}{(H + H_{ex})} \int_0^{(H+H_{ex})} \rho T dy \tag{3-9}$$

Equation (3-9) includes the height H of the test section which has been modified to include an "extra" volume (insulation porosity, etc.) enclosed in the heated cavity apparatus but not part of the actual test section (a 4x4x4 inch cube). The total free volume in the heated cavity apparatus was measured by measuring the volume of water it held. The "extra" volume is represented by an equivalent height, H_{ex}=0.79H.

It can be shown that the cavity pressure is only a function of time. This point is elaborated in Appendix C. If P is independent of the space coordinate y, Equation (3-9) reduces to the ideal gas state equation

$$\rho = \frac{P}{RT} \tag{3-10}$$

The pressure can be determined numerically by first assuming a pressure to calculate the density profile from Equation (3-10) and then iterating until continuity is satisfied, i.e.,

$$\frac{1}{(H + H_{ex})} \int_0^{(H+H_{ex})} \rho \, dy = 1 \qquad (3\text{-}11)$$

The adjusted temperature T_a and the core temperature T_c can now be determined by solving Equations (3-8), (3-10) and (3-11) iteratively. Considering the "extra" volume in Equation (3-11) has the effect of increasing the core temperature slightly, but has very little impact on the resulting adjusted temperature profile. The computational procedure is summarized in the following paragraph.

A computer program was developed which computes the adjusted spatial temperature profile for a given time based on the laser beam deflection data. The steps required for this computation can be summarized as follows:

a) Calculate the spatial temperature gradient from Equations (3-2) and (3-3) using the measured heated wall temperature and an assumed temperature profile elsewhere.

b) Solve for a new temperature profile using Equation (3-5) and the results of step (a).

c) Return to step (a) and use the new temperature profile calculated in step (b). Repeat until convergence is achieved for the unadjusted spatial temperature profile.

d) Calculate an adjusted temperature profile from Equation (3-8) using an assumed air core temperature T_c.

e) Determine the density profile from Equation (3-10) using the results of step (d) and an assumed pressure.

f) Check to see if continuity is satisfied using Equation (3-11). If not, revise the pressure appropriately and return to step (e). Repeat until convergence is achieved for pressure.

g) Compare the numerically determined pressure from step (f) to the experimentally measured pressure. If the calculated pressure is high, reduce the core temperature guess (or vice versa) and return to step (d). Repeat until convergence is achieved for the adjusted temperature profile T_{a_i} and the core temperature T_c.

The adjusted temperature profiles are shown in Figures 3-18 through 3-21. The profiles are observed to asymptotically approach the core temperature T_c away from the wall. The core temperature rises slightly over the experimental time by slow compression of the core region.

It also appears from the experimental results and comparison with numerical calculations, that the thermal boundary layer penetrates into the core region less than expected for the relatively long times of 5.0 and 10.0

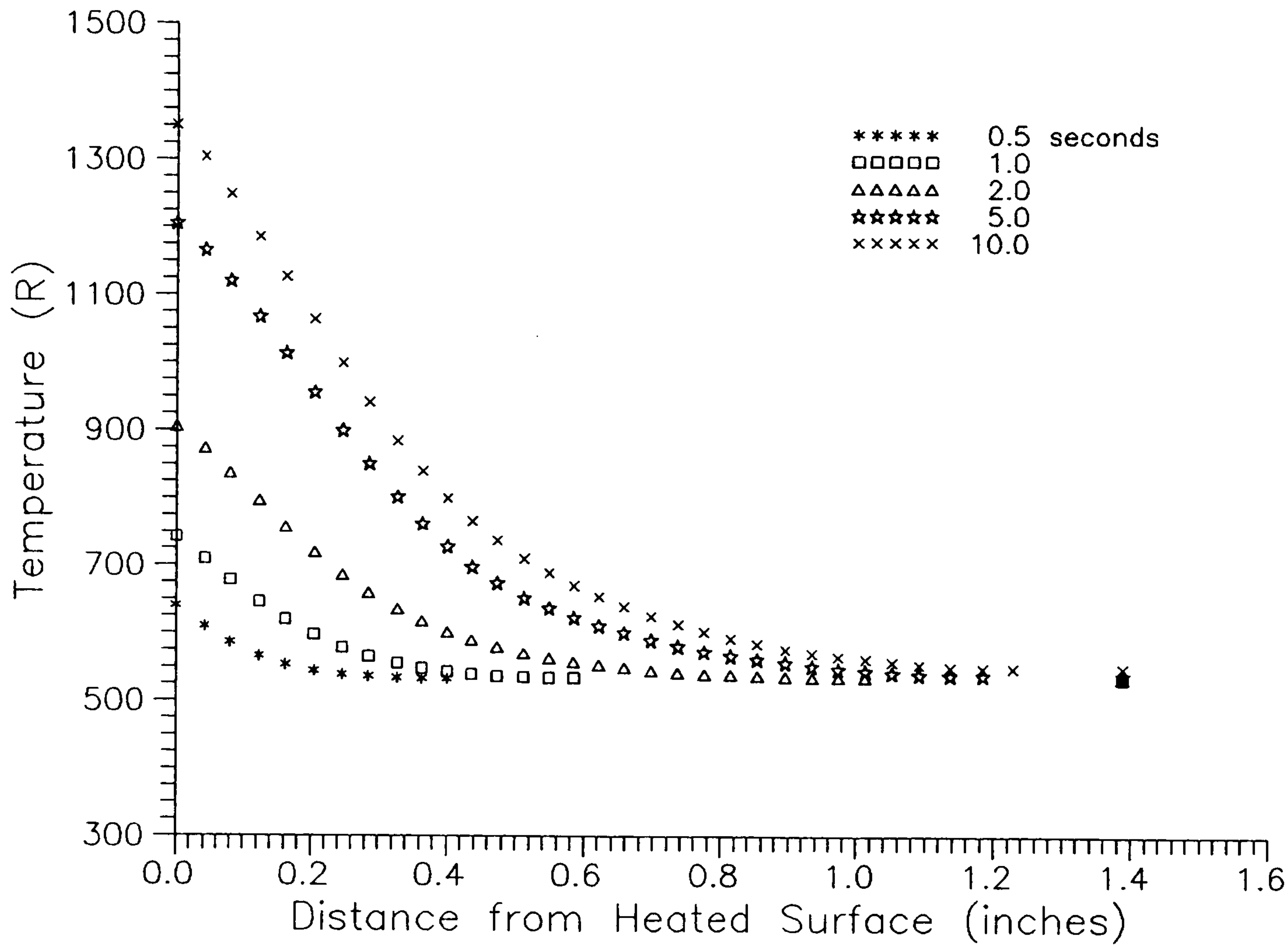


Figure 3-18. Corrected Temperature as a Function of Distance from the Heated Wall for Low Power

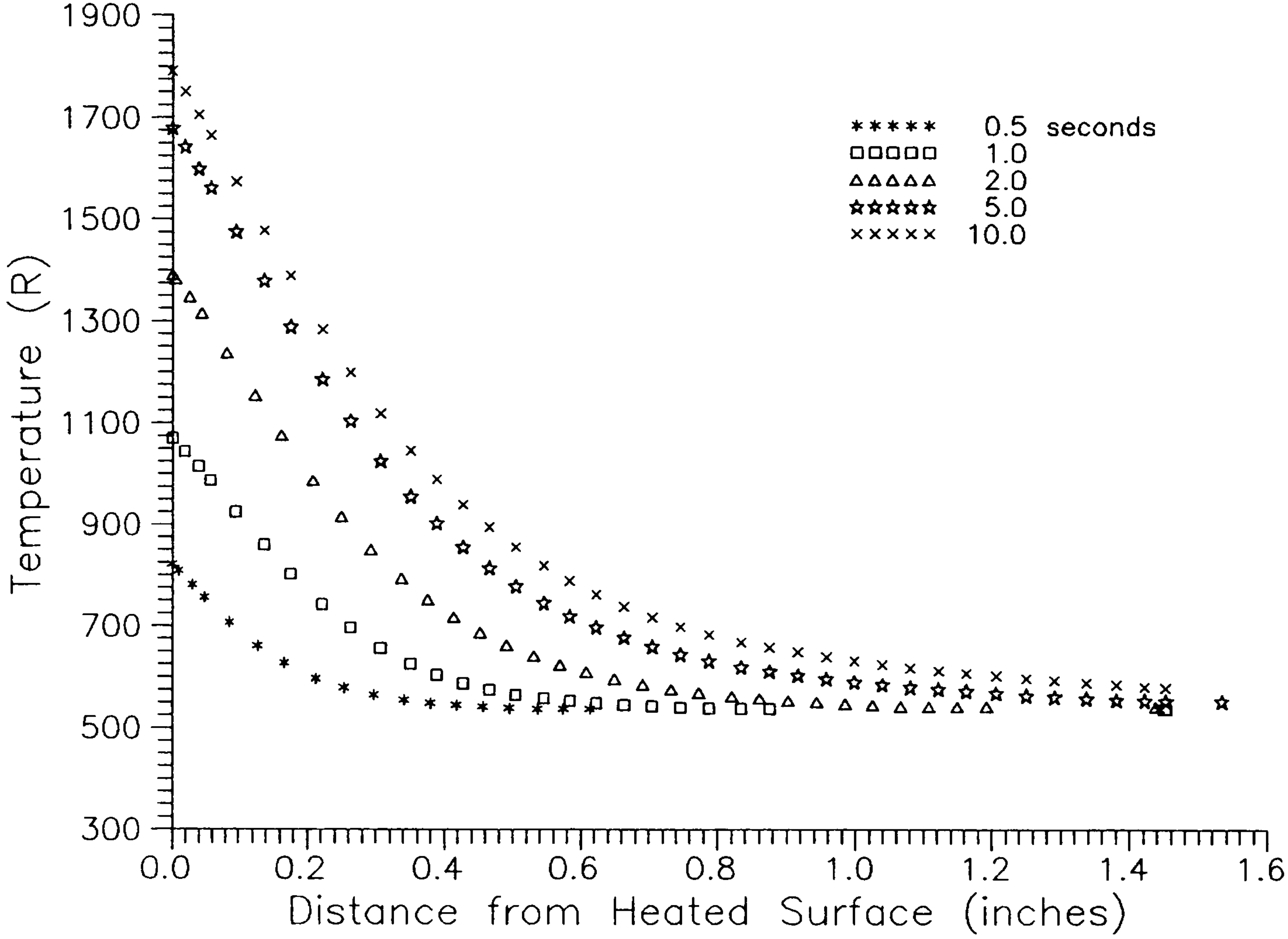


Figure 3-19. Corrected Temperature as a Function of Distance from the Heated Wall for Medium Low Power

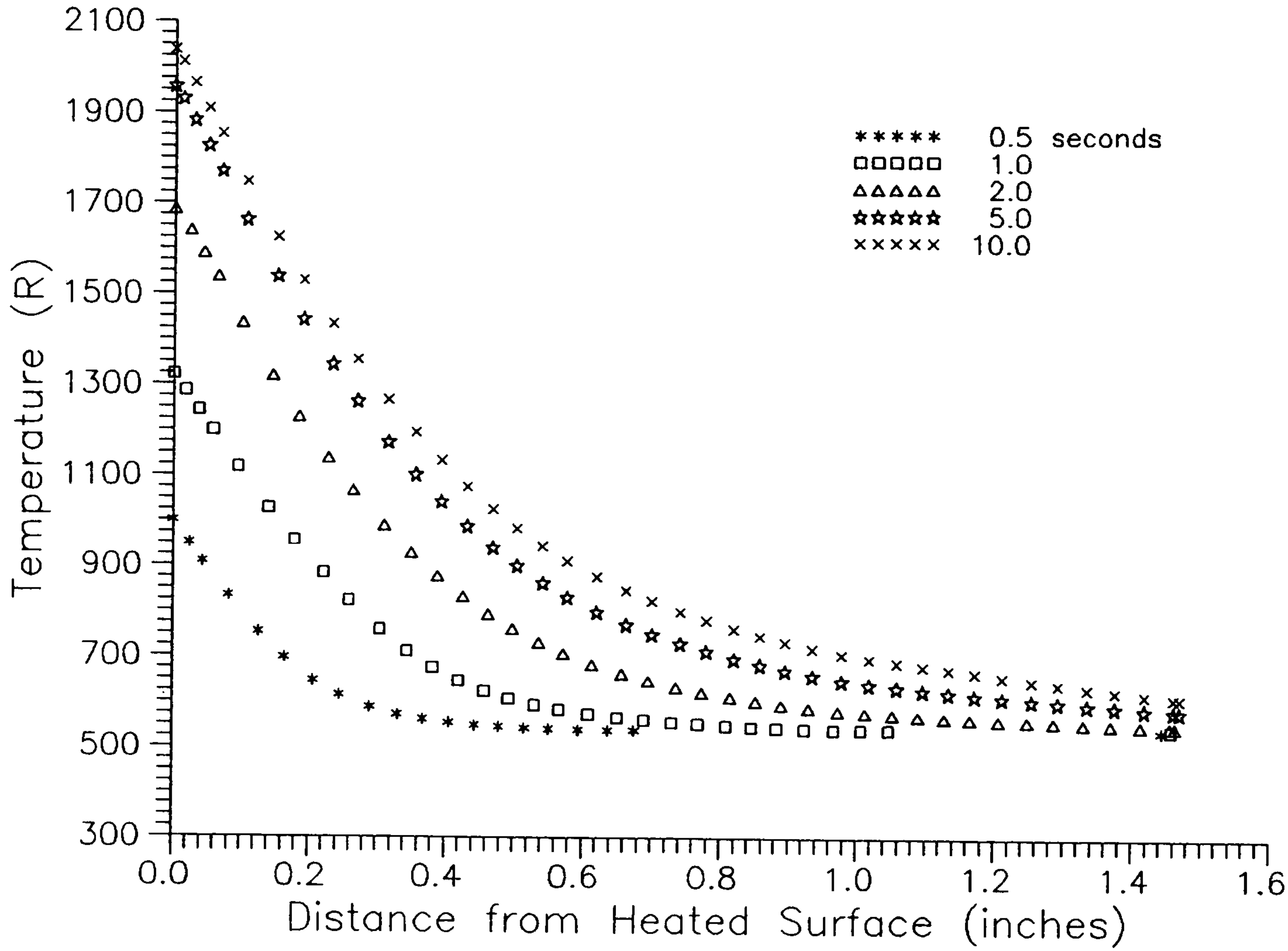


Figure 3-20. Corrected Temperature as a Function of Distance from the Heated Wall for Medium High Power

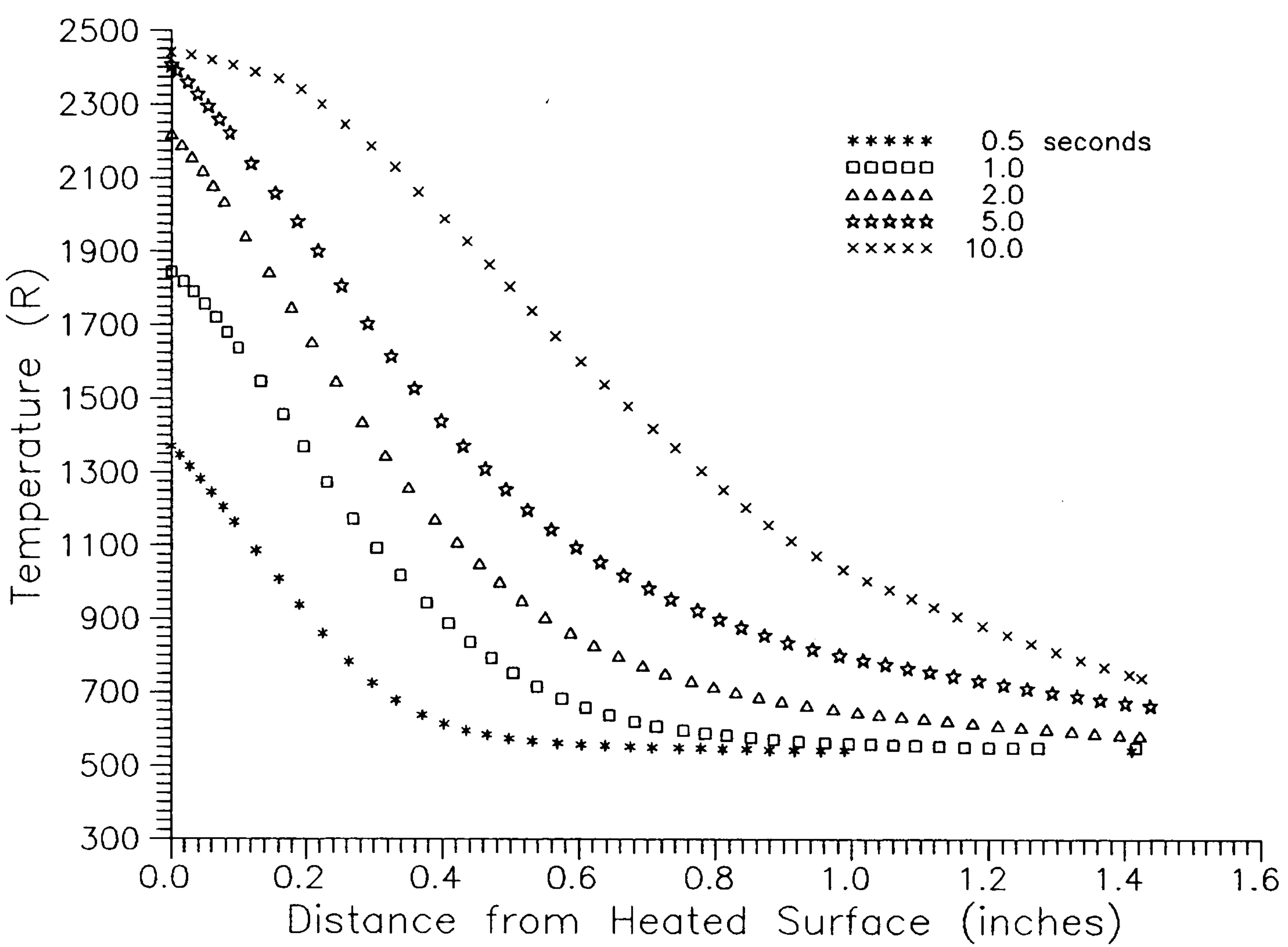


Figure 3-21. Corrected Temperature as a Function of Distance from the Heated Wall for High Power

seconds. This is discussed in more detail in Chapter 6. It was proposed that flow through the porous side wall insulation (a fiber material) may have a significant effect on the flow and thermal response of the fluid in the test section. To test this proposal, experiments were run in which the porous insulation side walls were replaced with glass side walls. These experiments showed that eliminating the porous side walls had no measurable effect on the temperature profile results.

The method that was described and selected for fluid temperature correction is not the only available method. Another way to correct the temperature profile is to determine T_c by performing an energy balance between the thermal boundary layer and the core region. This is discussed in Appendix E.

3.6 Flow Visualization and Air Velocity Measurement

Separate flow visualization experiments were performed using the same heated cavity apparatus that was described in Chapter 2. The wall opposite the heated wall included an open hole, so the test section was not pressurized. Tobacco smoke was injected into the test section to allow visualization. The experiment was performed for the high power case only. The results, recorded on movie film, indicate a uniform flow, or expansion, of the air away from the heated wall. A smoke-free space adjacent to the wall that was approximately one inch thick developed in about one second. However, the effect of the motion of the air may have been obscured by the effect of thermophoresis of the smoke particles in the high temperature gradient [33].

The average velocity of the air in the test section during expansion away from the heated wall was approximated by measuring the exit velocity of air through a hole in the bottom wall (opposite the heated wall) of the heated cavity. The velocity in the test section was estimated by multiplying the velocity through the hole by the ratio of the hole area to the area of the test section normal to the heated wall. Measurements were made with a hot-film velocity meter. The results are shown in Figure 3-22. The accuracy of the measurements provide an indication of the order of magnitude of the flow velocity and the results are presented only to demonstrate that the flow velocities are similar in magnitude to flow velocities that occur in natural convection. Note that Figure 3-22 provides spatially-averaged velocity, so the velocity in the region adjacent to the wall may be significantly higher.

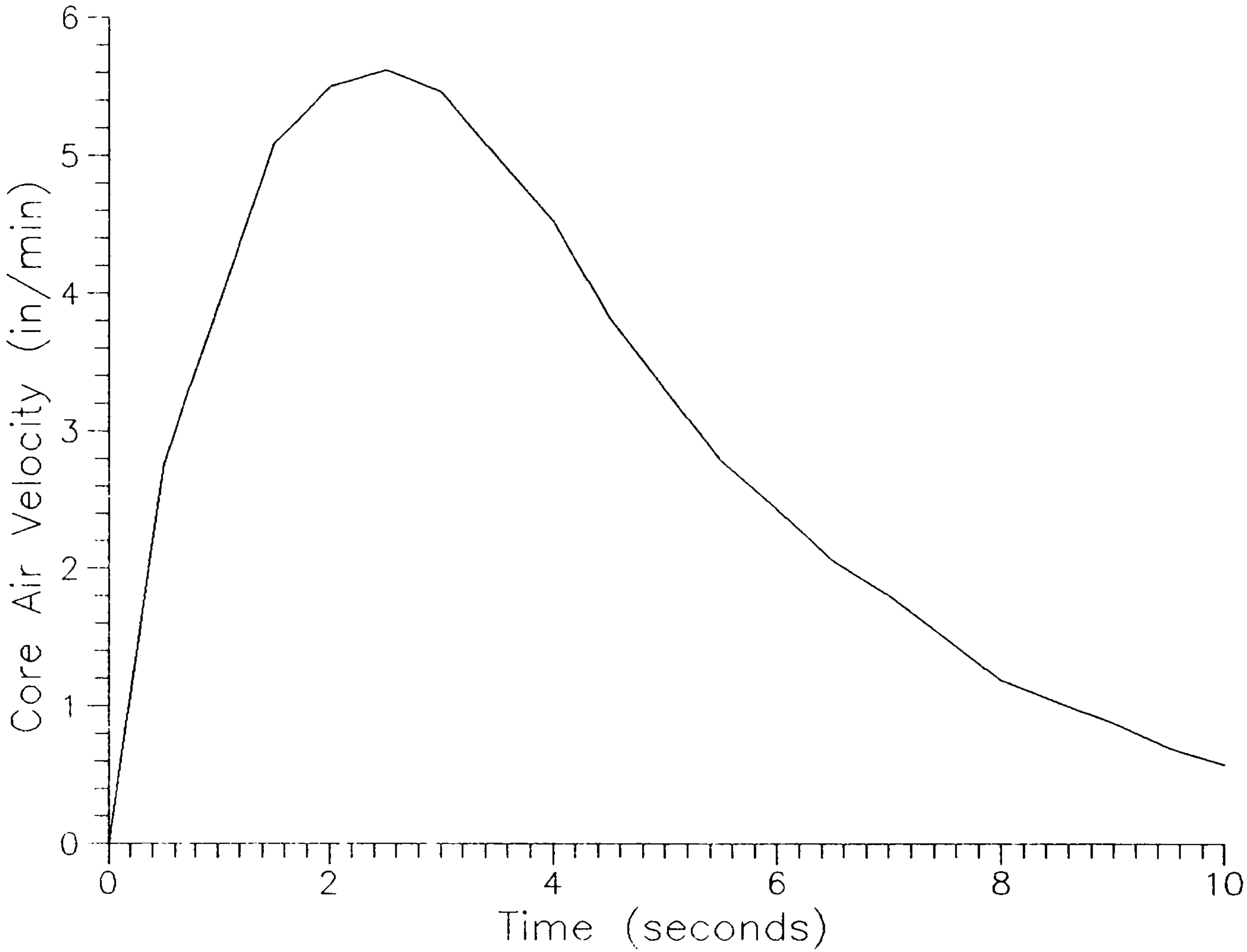


Figure 3-22. Average Air Velocity in the Test Section Core as a Function of Time for High Power

CHAPTER 4

SPACE EXPERIMENT CONCEPT DEFINITION

In this chapter the need for an in-space experiment to study thermally-induced convection heat transfer is discussed and a preliminary design for a space experiment is presented. The concept definition phase for an in-space experiment was performed under the sponsorship of NASA and was presented at NASA Lewis Research Center. One of the goals of the ground experiments described in Chapters 2 and 3 was to establish the feasibility of performing a useful and reliable space experiment. For more detailed results of the concept definition study, the reader is referred to Parang and Crocker [34].

4.1 Reasons for Experimentation in Space

The role and importance of thermally-induced convection heat transfer in a microgravity environment is unclear because of a lack of reliable analytical solution methods and meager experimental information. Thermally-induced convection may play a significant role in a variety of technological processes involving heat transfer. The effect has been observed in numerous gravity environment applications, however, it is in the microgravity environment, and in the absence of natural convection, that thermally-induced convection may acquire significance. This is because the only remaining mode of heat transfer to a transparent fluid is the slow molecular conduction process. Space applications where thermally-induced convection may play an important role include cryogenic fluid storage and handling systems, space manufacturing processes, and the stability of combustion systems.

The study of thermally-induced convection heat transfer in a gravity environment is made difficult by the interference of natural convection heat transfer. While the phenomenon can be studied with the heat flux vector parallel to the gravity vector (e.g., the experimental study described in Chapters 2 and 3), the presence of the gravity field may still mask important thermally-induced motion. For example, in heating a fluid from above, the stabilizing role of gravity may dampen important features of the induced motion.

Ground-based, zero-gravity experiments are also inadequate for providing a clear picture of the nature of heat transfer processes in this phenomenon. There are two time scales associated with the thermally-induced pressure waves. One is a fast time scale which is described by the characteristic length divided by the sound speed. A second time scale is associated with the motion induced by the expansion of the fluid. The latter motion is of the same order of magnitude as natural convection in enclosures and involves a relatively slower time scale. Therefore, to understand this aspect of the flow and compare with conduction heat transfer requires experimental times of the order of conduction time, i.e., tens of minutes or more. This length of time is longer than the available time for earth-bound microgravity experiments.

For these reasons, collection of experimental data on thermally-induced convection effects in microgravity will require the design and fabrication of an in-space experiment. An apparatus capable of producing and measuring thermally-induced convection heat transfer effects needs to be designed to take advantage of the longer experimental time available in the microgravity space environment. It is anticipated that the data obtained from the space experiment will help determine the importance of thermally-induced convective motion and will help provide design guidelines for space technologies in which this phenomenon is encountered. The information on flow and temperature fields obtained in these experiments will also improve thermal control and prediction of flow processes in which thermally-induced convective effects may usually be small relative to forced convection flows, but may still play an important role in near critical conditions.

4.2 Baseline Space Experiment Design

The primary objective of the space experiment is to measure the transient temperature of air as a function of distance from a rapidly heated wall. An isometric view of the preliminary experiment layout is shown in Figure 4-1. The design includes the heated cavity, laser system, optical temperature measurement system, data recording system and power supply. The preliminary experiment is designed for deployment on the middeck of a Space Transportation System Orbiter. The experiment will be mounted on two adjacent single adapter plates. The dimensions of the experimental envelope are shown in Figure 4-1. The Orbiter provides up to 115 watts of continuous power for up to eight hours, or 200 watts peak power for 10 seconds or less. Heat rejection is also limited to 115 watts continuous dissipation. The maximum payload weight is 62.8 pounds per single adapter plate. Detailed descriptions of the flight hardware and the optical system layout are given in Appendix F. Many aspects of the space experiment design are similar to the design of the ground experiments described in Chapters 2 and 3. Details of the space experiment are described below with the exception of elements that are in common with the ground experiments and were described earlier.

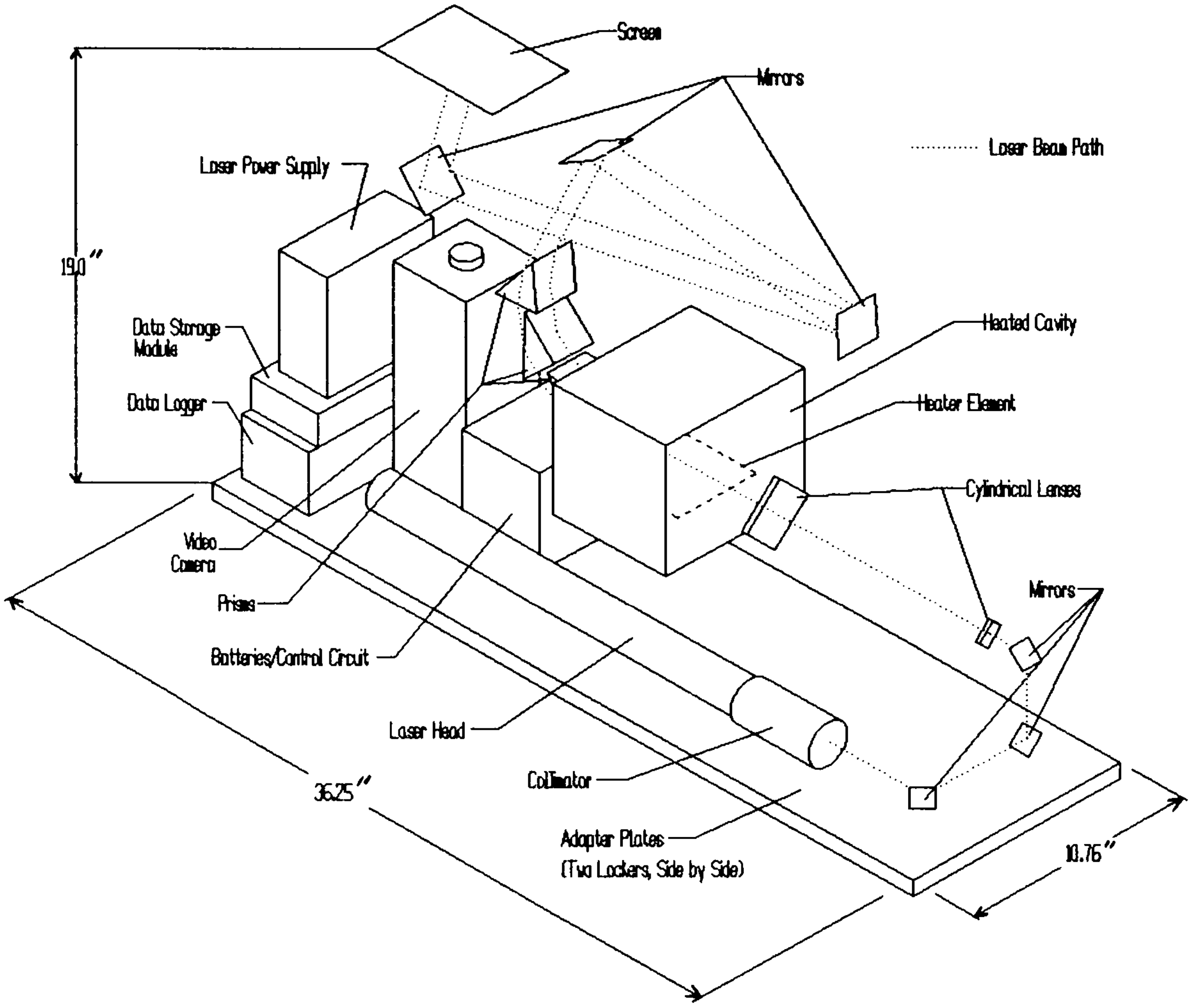


Figure 4-1. Isometric View of the Space Experiment Hardware Layout

Heated Cavity

The baseline design for the heated cavity is the same as that described in Chapter 2. However, the dimensions of the test section are critical to the space experiment design and are subject to change. Criteria for selecting the test section size are discussed below.

The test section volume is V=LWH where L is the laser path length, W is the distance between the busbars, and H is the distance from the heated wall to the cold wall, as shown in Figures 2-2 and 2-3. The choice of the test section size should be based on the following considerations:

a) Power Dissipation -- The power dissipation of the heater element can be expressed as

$$\text{Power} = \frac{V^2 L t}{k W} \tag{4-1}$$

where V is the voltage drop across the heater element, t is the thickness of the heater element, and k is the electrical resistance of the heater element material.

b) Heat Generation Rate in the Heater Element --

$$q''' = \frac{Power}{LWt} \tag{4-2}$$

and since Power=VI=$(V^2Lt)/(kW)$, we have

$$q''' = \frac{V^2}{kW^2} \tag{4-3}$$

c) One-Dimensional Approximation -- Larger L and W clearly improves the one-dimensional approximation and L=W maintains symmetry. The layout of the baseline design limits W to about 5 inches.

d) Laser Deflection -- Laser deflection is proportional to L.

e) Space and Weight -- Smaller dimensions are obviously advantageous. H is flexible since it does not affect the above considerations.

f) Ease of Construction -- There may be difficulty if the test section is too small. A reasonable lower limit for the size of the test section is L=W=2 inches.

Test section volume dimensions of 4x4x4 inches were selected for the baseline experiment design.

Fluid Medium

The fluid selected for the baseline space experiment design is air. The choice of fluid medium depends on the following:

a) Compressibility -- High compressibility is required to emphasize the effects of thermally-induced convection.

b) Index of Refraction -- A high index of refraction is required to maximize the deflection of the laser beam.

c) Ease of Handling -- The fluid must not be flammable, toxic or corrosive. Air is clearly the choice since leakage would not be a concern.

d) Near-Critical Condition -- Compressibility effects are greater near the critical point. Important space fluid heat transfer problems involve cryogenic fluids near the critical point.

e) Thermal Diffusivity -- A high thermal diffusivity results in a higher heat transfer rate which minimizes the required experiment run time and the power requirements.

Air is the logical choice to satisfy (a) and (c). The use of cryogenic fluids will satisfy (d), however, their use would introduce complications (e.g., handling) that are difficult to control or resolve in the space experiment. Table 4-1 shows the refractive properties (see Equation (2-12)) and thermal diffusivity of several gases. CO_2 is slightly better than air for refraction, but has thermal diffusivity less than half that of air. Helium has a very high thermal diffusivity, but very low refractive index. Since there is no clear winner considering refractive and thermal properties, air is chosen for the baseline design because of its clear superiority to satisfy (c).

Optical Temperature Measurement System

The optical method is essentially the same as that described in Chapter 2. Hardware specifications and design details for the space experiment optical system are described in Appendix F. In the ground experiment, the deflection of the laser beam in the test section was amplified by providing for a long laser beam path length past the test section (see Figure 2-1). Since space is limited for the in-space experiment, adequate beam deflection at the screen must be achieved by other means.

The deflection of the laser beam is proportional to the distance from the test section to the screen, and in the limited space available (approximately 3 feet) for the in-space experiment, the deflection would have to be amplified for accurate measurement. This can be achieved by using either mirrors or prisms. In the first case, a series of mirrors is employed reflect the beam several

Table 4-1. Refractive and Thermal Properties of Gases

Gas	$(n_o-1)/\rho_o R$ ($(°R \cdot ft^3)/Btu$)	Thermal Diffusivity (ft^2/hr)
Air	0.060	1.28
CO_2	0.078	0.59
Helium	0.0074	12.09
Argon	0.058	1.44

times effectively increasing z_s. However, since z_s must be about 70 feet to get a maximum deflection of approximately 1 inch, this method is not practical.

The preferred and selected method is to use one or more right angle prisms to amplify the laser beam deflection. A wide range of deflection angle amplification can be achieved by varying the number of prisms in series and the initial (undeflected) angle of incidence of the laser beam. Figure 4-2 is a schematic of the path of a laser ray through a series of three prisms.

The incident angle α_i of the laser ray entering and the exiting angle α_e for each prism can be found by applying Snell's law

$$\frac{\sin\alpha_a}{n_a} = \frac{\sin\alpha_p}{n_p} \tag{4-4}$$

where the subscript a indicates the air side and the subscript p indicates the prism side of the air-prism interface. Since $n_a \approx 1$,

$$\alpha_{e_k} = \sin^{-1}\left[n_p \sin\left(45^\circ - \sin^{-1}\left(\frac{\sin\alpha_{i_k}}{n_p}\right)\right)\right] \tag{4-5}$$

where k indicates the k^{th} prism in a series of prisms. The prisms are oriented relative to each other by

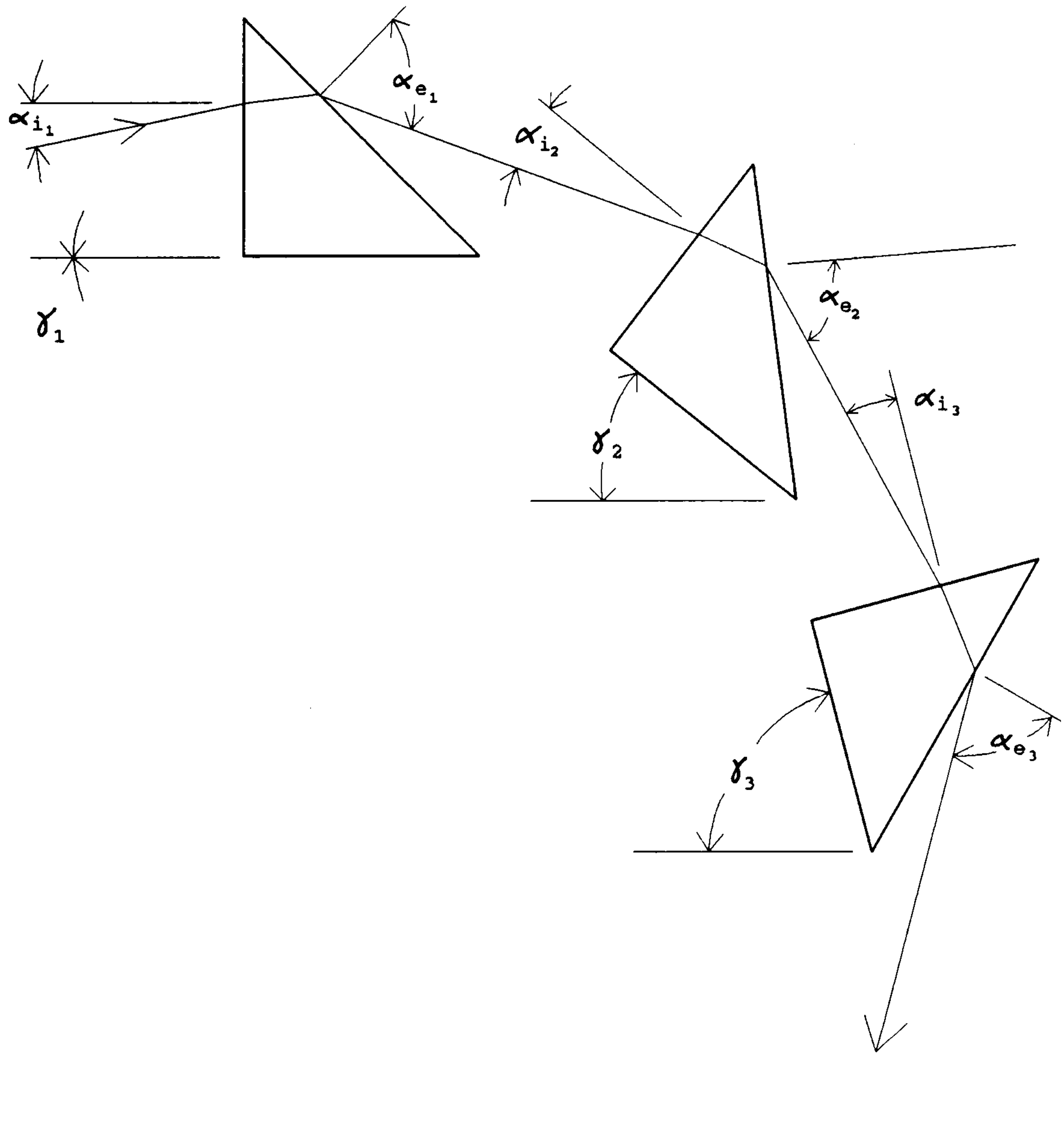


Figure 4-2. Schematic of a Laser Ray Path through a Series of Three Prisms. (γ_1 is arbitrary and is set to zero.)

$$\gamma_k = \gamma_{k-1} + \alpha_{e_{k-1}} + \alpha_{i_k} - 45^\circ \tag{4-6}$$

where angle γ_k is defined in Figure 4-2.

The above equations describe the path of an undeflected laser ray where α_{i_k} is predetermined. Given this fixed arrangement of the prisms, it is necessary to determine incidence angles for a deflected ray. This can be achieved by using the above equations and the relation

$$\alpha'_{i_k} = \alpha_{i_k} + \left(\alpha_{e_{k-1}} - \alpha'_{e_{k-1}}\right) \tag{4-7}$$

where primes indicate incident angles for a deflected ray. Figure 4-3 shows the ratio r_p of the change in exit angle to the change in entrance angle as a function of the change in the entrance angle, that is, $r_p = \left(\alpha_{e_3} - \alpha'_{e_3}\right)/\left(\alpha_{i_1} - \alpha'_{i_1}\right)$ as a function of $\left(\alpha_{i_1} - \alpha'_{i_1}\right)$, for a series of three prisms. The relationship is nonlinear, but for the small angle changes considered (maximum change in entrance angle expected is 0.1°), the deviation from linearity is small for $\alpha_{i_k} > 7^\circ$. The flight experiment has been designed for $\alpha_{i_k} = 8^\circ$ which provides an angle change ratio of approximately 25. Thus, using only three prisms and locating the screen just 4 feet past the prism array provides an effective z_s of approximately 100 feet.

Unfortunately, amplification of laser beam deflections is not the only effect of the prism array.

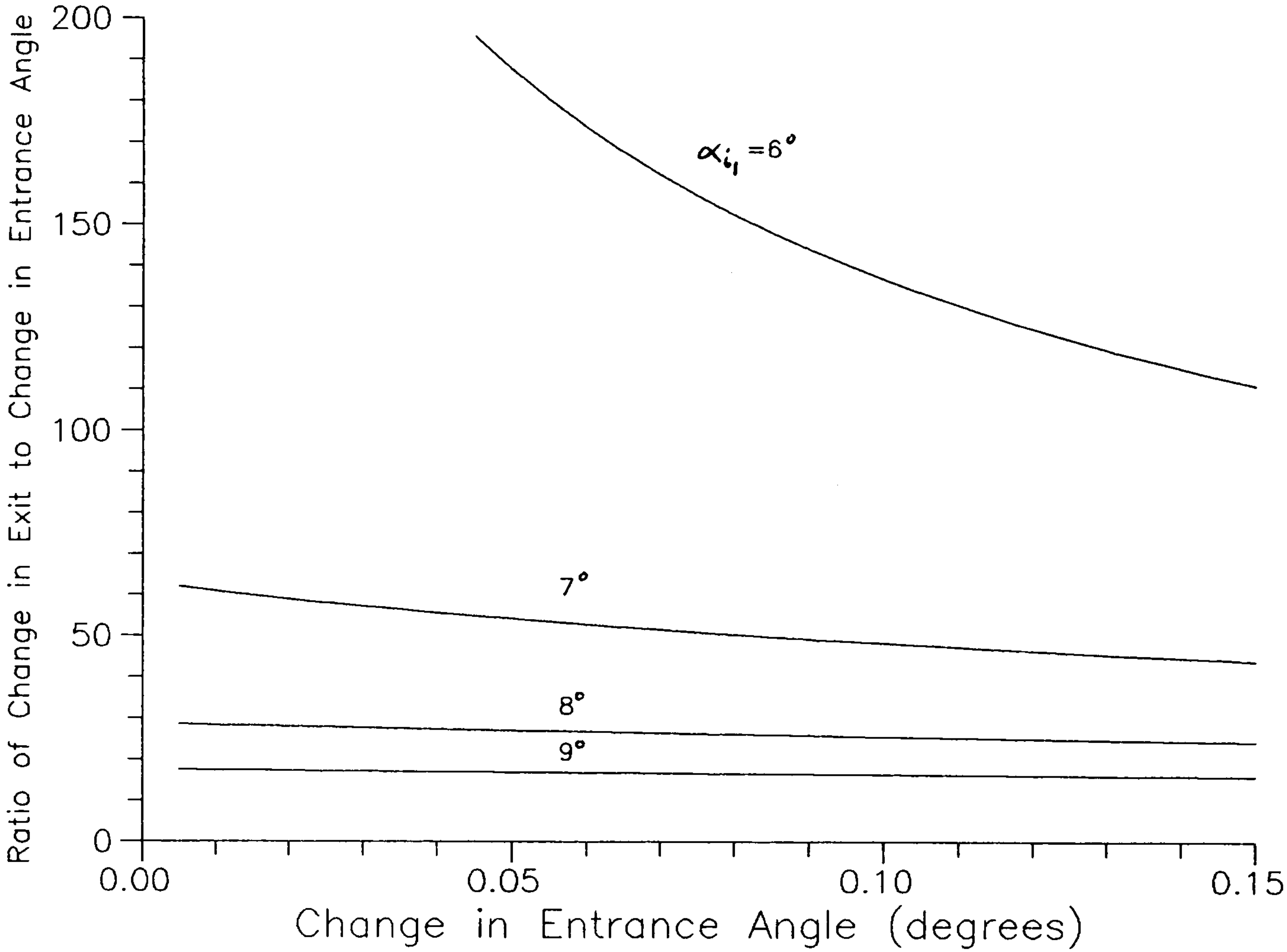


Figure 4-3. Ratio of Change in Exit Angle to Change in Entrance Angle as a Function of Change in Entrance Angle for Three Prisms

Also, because the experiment requires a laser sheet rather than a narrow beam, the following additional comments should be considered:

a) The laser sheet will be constricted in the direction of deflection. The prisms must be oriented so that deflections are perpendicular to the prism base as shown in Figure 4-4. (Refer also to Figure 2-6 for comparison of the laser sheet orientation in the test section.) This constriction is simply a result of the turning of each parallel ray of the laser sheet as shown in Figure 4-5. The effect is compounded by each prism. The constriction is not a problem however, because $w(\sin\theta)$ remains constant, and there is still a linear relationship between the distance along the laser sheet profile and the distance from the heated wall.

b) The above constriction effect also results in a thin image on the screen if the incoming rays are parallel and remain parallel after the prism array. If they are not perfectly parallel, however, the divergence will be amplified. The prism system is, of course, designed to amplify angle changes. Even if the incoming rays are perfectly parallel, divergence due to diffraction effects on the very thin sheet is unavoidable. The result is a thicker line at the screen. A well collimated beam should minimize this

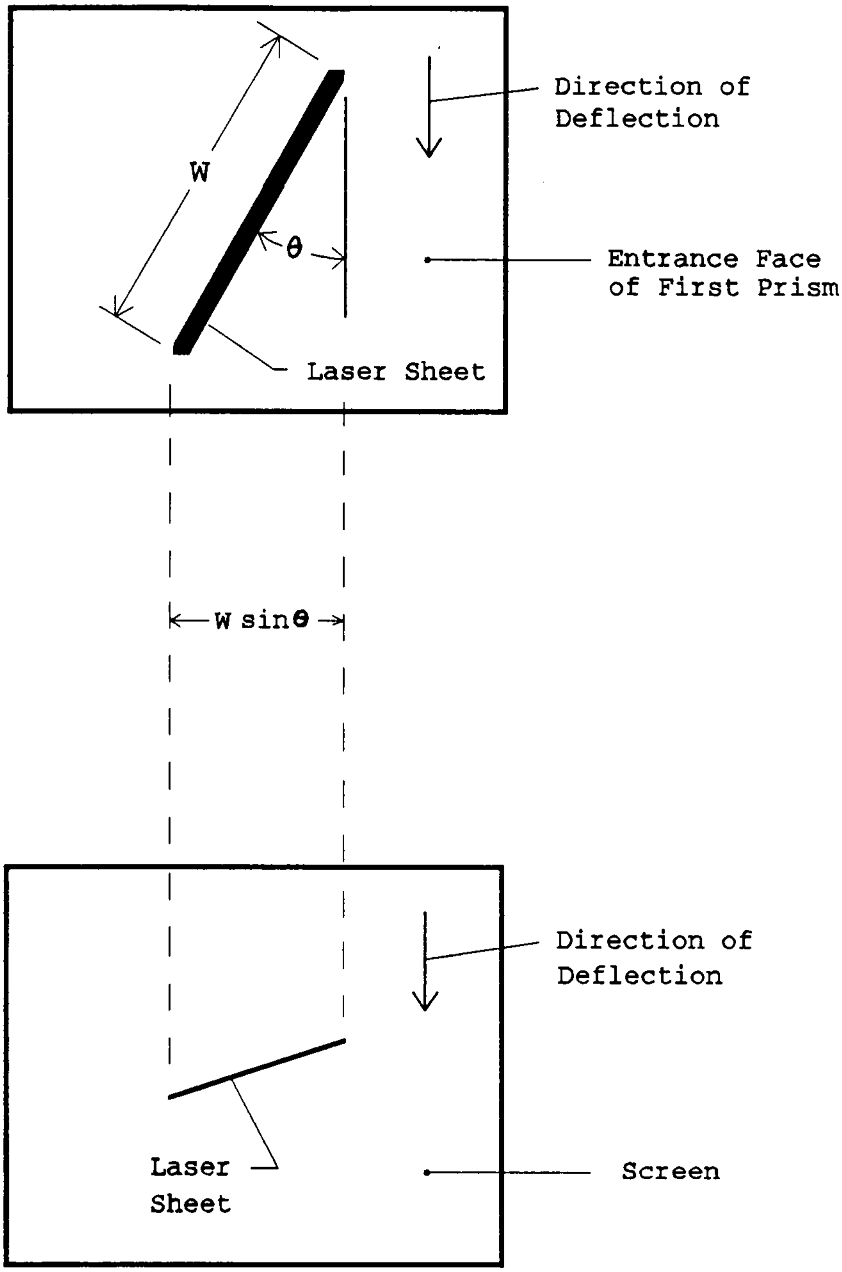


Figure 4-4. Undeflected Laser Sheet Orientation at the Entrance Face of the First Prism and at the Screen

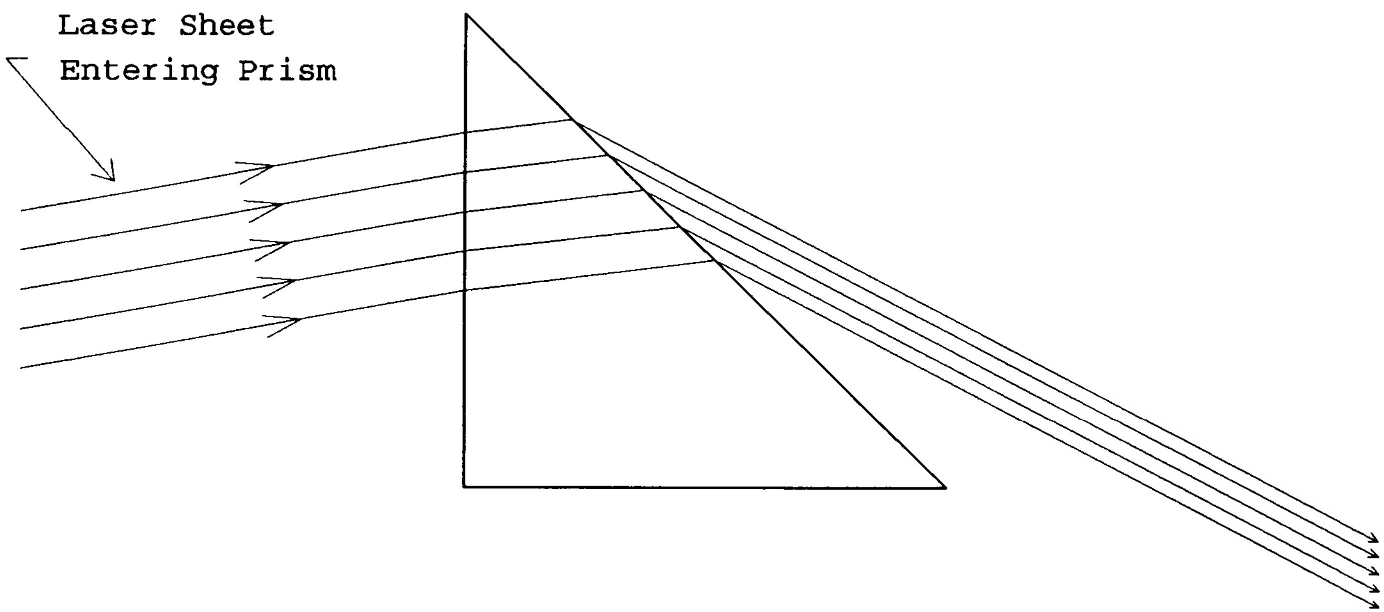


Figure 4-5. Constriction of Laser Sheet by One Prism

effect, and furthermore, even a relatively thick image at the screen will influence the accuracy of deflection measurements only minimally since only beam displacement from a reference position (e.g., the edge of the beam) is measured.

c) Reflections of the beam at each glass/air interface will reduce the intensity of the beam at the screen. The relatively high incidence angle as the beam exits each prism results in high reflection. Preliminary calculations indicate that approximately 39% of the original beam intensity would be transmitted for the present design with no anti-reflection coating. This lower bound is sufficient if a 10 milliwatt He-Ne laser is used. Also, a MgF_2 anti-reflection coating will significantly improve transmittance.

Equation (2-13) provides the relationship between the temperature gradient and the deflection and can be used to obtain the spatial air temperature profile as previously described in Chapter 3. This equation, however, does not consider the amplification of the deflection by the prism system, so an effective length z'_s past the test section must be determined. z'_s can be approximated, assuming small angle deflections, by

$$z'_s = z_b + r_p z_a \qquad (4\text{-}8)$$

where z_b and z_a are the laser path distance before and after the prism system, respectively, and r_p is the angle ratio as shown in Figure 4-3. This equation neglects the deflection that occurs within the prism system. The accuracy of the equation is also limited because r_p is highly sensitive to the orientation of the prism system and is a function of the deflection angle. Therefore, z'_s probably needs to be determined in terms of y_s by calibration. This is accomplished by turning the incoming beam by a known angle β and noting y_s, then

$$z'_s = \frac{y_s}{\sin\beta} \qquad (4\text{-}9)$$

Equation (2-13) can now be used (with z'_s rather than z_s) to determine the temperature gradient $\partial T/\partial y$. The gradient can then be used with the heated wall boundary condition measured with thermocouples to determine the temperature profile, T(y). This calculation was described in detail in Chapter 3.

Power Supply

The main power for the heater element in the space experiment can be supplied in either of two ways. One way is to use rechargeable batteries. The batteries would supply the large power requirement for the relatively short duration of each experiment run and then be recharged between experiment runs by STS Orbiter power. Each battery is size D (about 2.5 inches high by 1.25 inches in diameter) with 4 amp-hrs and 1.2 volts and weighs about 0.29 pounds. The baseline design provides space for 18 batteries, but more could easily be added if necessary. The batteries must be configured in a combination of series and parallel so that sufficient power can be supplied at various voltage levels. The voltage level determines the heating rate of the heater element. A simple circuit diagram corresponding to the battery power supply method is shown in Figure 4-6.

Another way to supply power is to directly use Orbiter power. The elimination of batteries has several advantages including improved safety, reduction of weight, more available space, and simplification of the design. The disadvantage is the limitation of the power supply. The current limitation of 115 watts for middeck experiments would require much smaller heated wall dimensions to achieve a high heating rate. It may be possible to use the Middeck Utility Panel (MUP) with a power supply of 1 kW to resolve this problem, however, the MUP has limited availability.

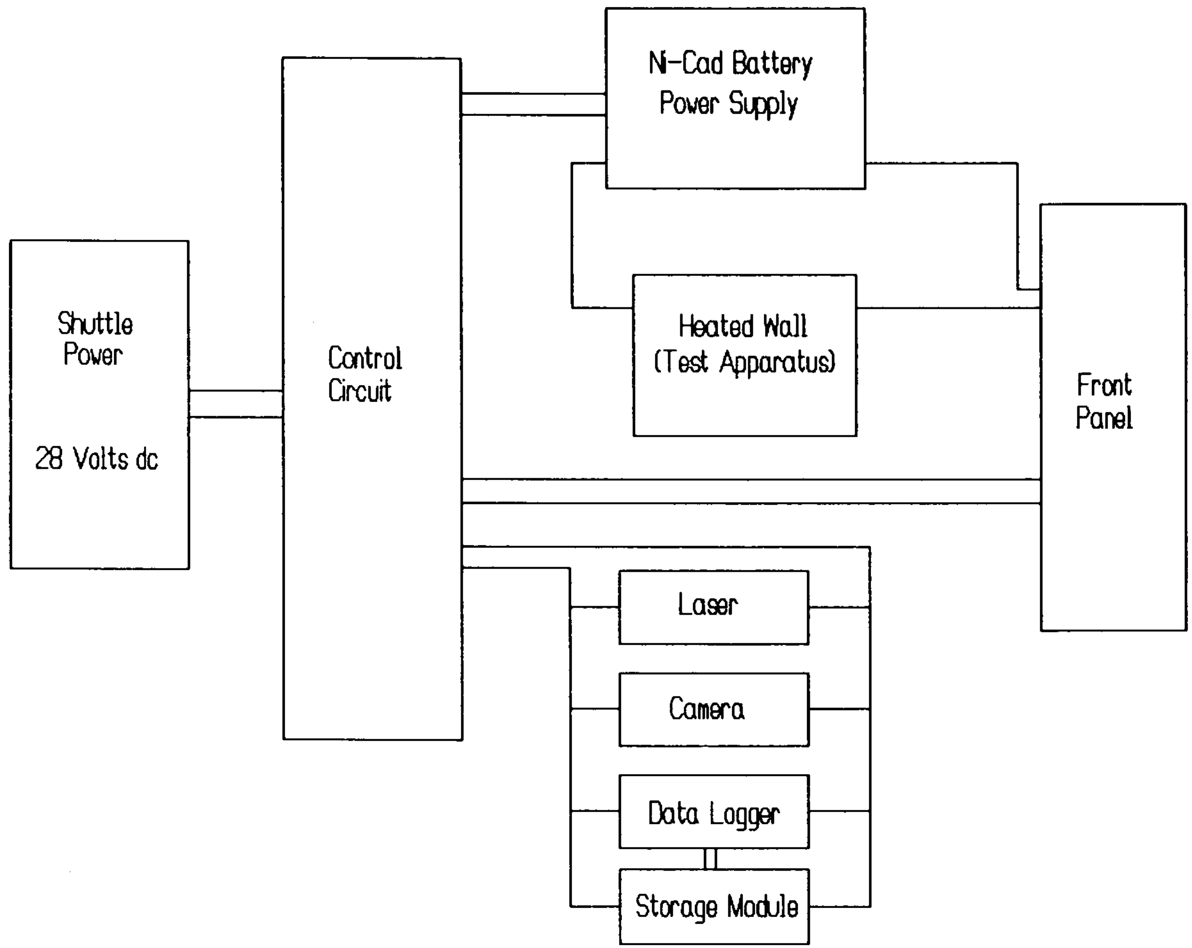


Figure 4-6. Space Experiment Circuit Diagram

Procedure

Most of the in-space experimental steps can be executed automatically so that crew interaction with the experiment is minimized. The basic steps required to perform one experiment run are as follows:

a) Turn on front panel switch to begin run.
b) Turn on laser, video camera, data logger and storage module.
c) Adjust voltage level to be applied to heated wall.
d) Turn on power to heated wall for duration of run.
e) Turn off power to heated wall.
f) Turn off laser, video camera, data logger and storage module.
g) Turn off front panel switch to end run.
h) Begin recharge of battery.
i) End recharge of battery.

It is probably necessary for the crew to perform steps (d) and (e) since the relatively high current to the heated wall would require a large, heavy relay for automatic switching. The duration of each run depends on available power supply.

CHAPTER 5

NUMERICAL INVESTIGATION

5.1 Overview

This chapter examines the solution of the governing equations of thermally-induced convection heat transfer by various numerical or analytical methods. First, the numerical scheme of Spradley [28,29] is analyzed and its shortcomings discussed. Several changes are introduced into Spradley's numerical scheme, the inaccuracy of that model is discussed and the difficulties involved in obtaining an accurate numerical model are illustrated. Second, a finite volume numerical model that improves the prediction of the transient spatial temperature profile is described. This model is essentially a conduction model which includes the convective effects of the bulk velocity generated as the heated fluid expands and contracts in response to the evolving temperature and pressure fields. The model is therefore termed an "expanding conduction" model. Later, in Chapter 6, the results of these two numerical schemes, the perturbation analysis of Radhwan and Kassoy [22] and the experimental data presented in Chapter 3 are compared and discussed.

The discussion is primarily limited to a one-dimensional model of the problem for several reasons. First, comparison of the results with previous works will be simple and convenient since they also use a one-dimensional model. Also, a one-dimensional model enables easier identification of the important differences in the modeling of the physical process. Finally, the important physical features of the problem can be represented in a one-dimensional flow and it should be easy to extend to a multidimensional domain. In fact, as will be discussed later, the expanding conduction model has been extended to three dimensions for the heat conduction terms in the energy equation.

5.2 The Governing Equations

The governing equations as developed by Spradley, et al. [28] are briefly reviewed here in preparation for a detailed discussion of the numerical scheme developed in Reference 28. Later, the discussion will involve some changes in the equations which will be appropriately noted.

Consider a compressible fluid enclosed by two infinitely-long flat plates. The plates and the enclosed fluid are assumed to be at an initial temperature T_o', where primes indicate dimensional quantities. At time t=0, the temperature of one of the plates is suddenly raised to a large and constant temperature T_w'. The fluid is assumed to be an ideal gas with constant thermal properties k', c_v', μ' and γ. The second coefficient of viscosity λ' follows Stokes viscosity hypothesis, $\lambda' = -(2/3)\mu'$. Assume no radiation, no internal heat sources, zero gravity, and neglect viscous dissipation of energy. The latter can be shown to be justified by appropriate scaling of equations [30]. Space, time, temperature, velocity, pressure and density are normalized with H', $H'/\sqrt{R'T_o'}$, T_o', $\sqrt{R'T_o'}$, P_o' and ρ_o', respectively, where H' is the distance between the plates and R' is the specific gas constant. With the above assumptions the normalized equations become:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{5-1}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho v v)}{\partial y} = -\frac{\partial P}{\partial y} + \frac{4}{3}\left(\frac{1}{Re}\right)\frac{\partial^2 v}{\partial y^2} \tag{5-2}$$

$$\rho\frac{\partial T}{\partial t} + \rho v\frac{\partial T}{\partial y} = -(\gamma-1)P\frac{\partial v}{\partial y} + \left(\frac{\gamma}{Pe}\right)\frac{\partial^2 T}{\partial y^2} \tag{5-3}$$

$$P = \rho T \tag{5-4}$$

The dimensionless initial and boundary conditions are:

$$\begin{aligned} u(y,t=0) &= 0, \\ T(y,t=0) &= 1, \\ \rho(y,t=0) &= 1, \\ v(y=0,t) &= v(y=1,t) = 0, \\ T(y=0,t) &= T_w'/T_o', \\ T(y=1,t) &= 1 \end{aligned} \tag{5-5}$$

The dimensionless groups which appear in the equations are:

Prandtl number	$Pr = \mu' c_p'/k'$
Reynolds number	$Re = (\rho' H' \sqrt{R' T_o'})/\mu'$
Peclet number	$Pe = RePr$
Specific heat ratio	$\gamma = c_p'/c_v'$.

5.3 Analysis and Modification of Previous Numerical Work

Description of Previous Numerical Work

The numerical model of Spradley, et al. [28] is used here as a basis for discussion, modification and comparison in the modeling of thermally-induced convection. The above set of dimensionless equations were solved by Spradley, et al. using an explicit finite difference scheme. Forward time differences were used for the unsteady terms and all space operators except the convection terms were expressed in space-centered difference form. The convection terms in both momentum and energy equations used an upwind difference scheme. The numerical scheme first computed (ρv_i^{n+1}) from Newton's Second Law. The spatial first derivative of (ρv_i^{n+1}) was then evaluated and substituted in the continuity equation to determine the new density profile. The velocity at all node centers was then calculated at the new time level and substituted in the energy equation to compute (T_i^{n+1}). Finally, the new pressure distribution was determined using the equation of state.

The results obtained by this numerical method are presented in Figures 5-1 and 5-2 for helium initially at standard conditions, $H'=15.3$ cm and $T(y=0,t)=2$. Figure 5-1 shows the calculated dimensionless velocity profile at the midpoint between the walls as a function of time. This figure shows an oscillatory behavior for the fluid velocity and illustrates the acoustic nature of the fluid motion. The calculated spatial temperature profile at $t'=0.2$ seconds shown in Figure 5-2 is much closer to steady state values than a solution corresponding to the pure conduction heat transfer mode. While thermally-induced convection may be an effective mechanism for enhancement of heat transfer, no reasonable physical argument can explain the short transient times computed by this numerical model. The solution for the temperature field obtained by Larkin [27] is also shown in Figure 5-2. The temperature profiles obtained by each method agree

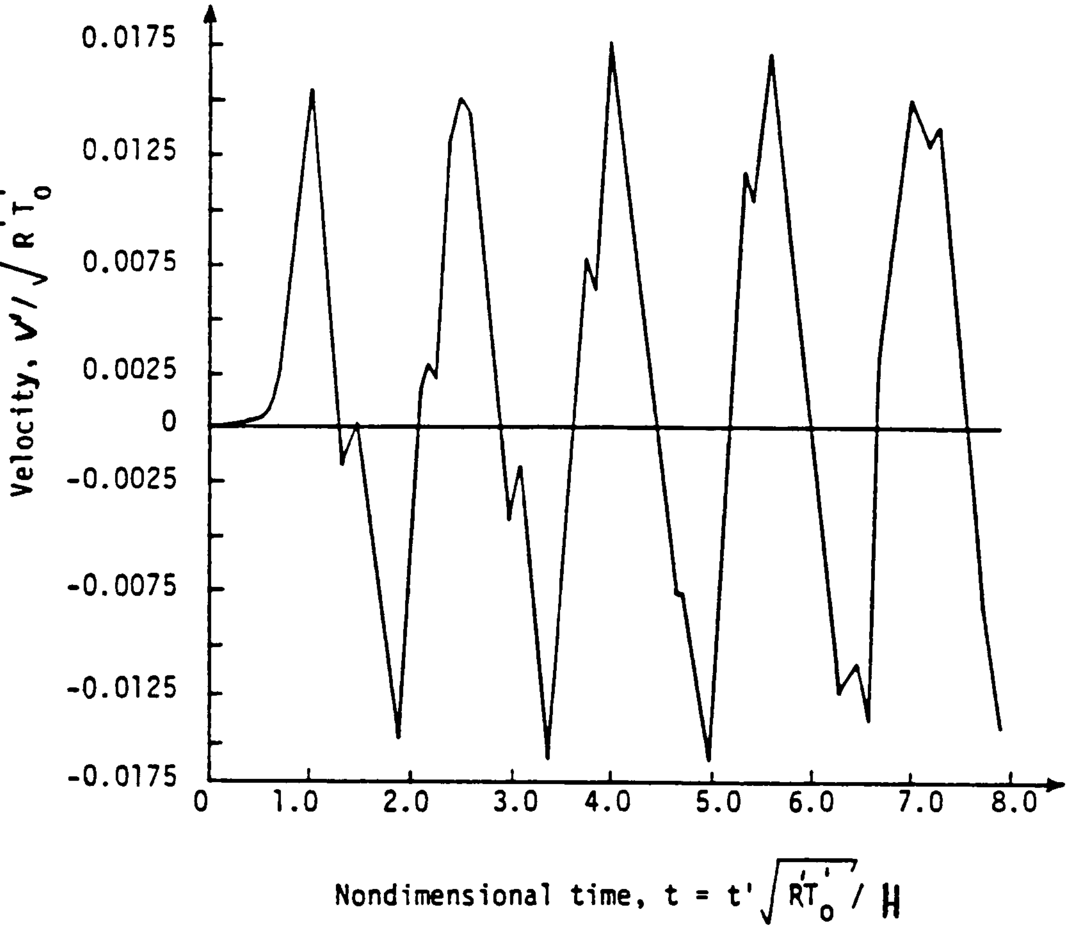


Figure 5-1. Unsteady Velocity at the Midpoint Between Plates, Reference 28

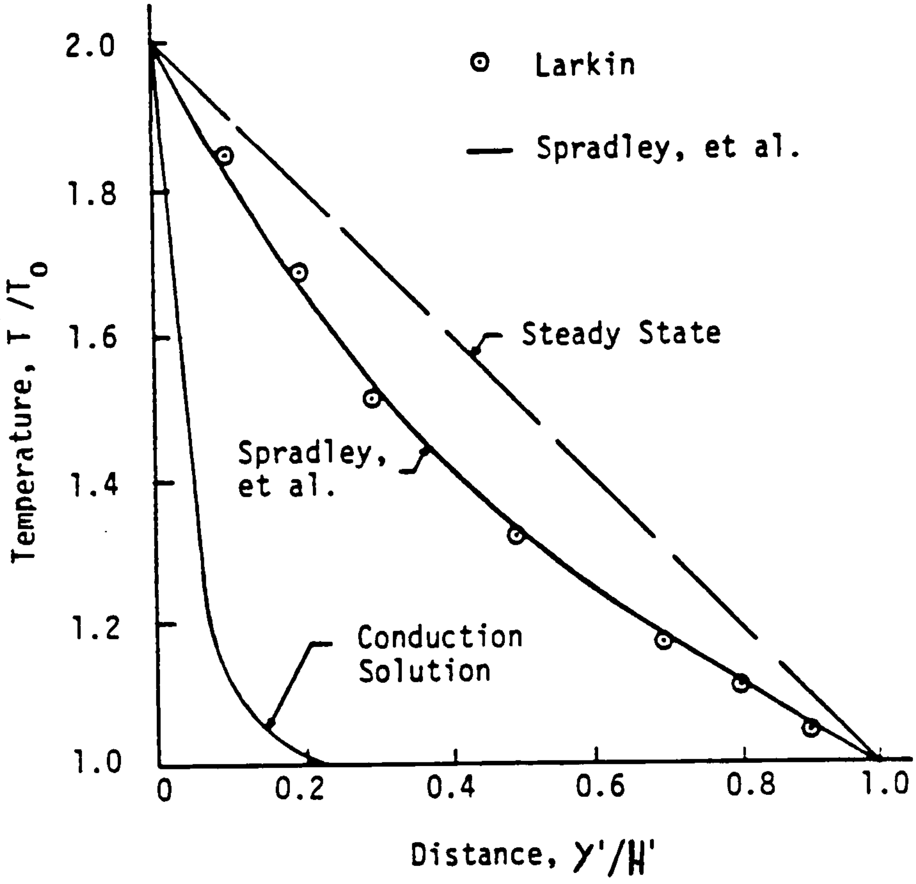


Figure 5-2. Temperature Distribution with Time at 0.2 Seconds, Reference 28

almost exactly even though the velocity amplitude computed by Larkin was approximately an order of magnitude higher than that of Spradley, et al.

The numerical scheme used by Spradley, et al. has been reproduced and then modified in several ways to ascertain the accuracy of the computational methods used in that study. The dimensional quantities included in the following results are for air initially at standard conditions and H'=30.5 cm.

Grid Refinement Study

A series of spatial and temporal grid refinements was implemented to investigate the effect of grid size on the numerical results. The results of an accurate numerical model will converge to a unique solution as the grid size is further refined.

A range of time and space increments was chosen and velocity and temperature results were computed and compared. Table 5-1 includes a partial list of space and time increments used in the numerical computations. The original numerical work of Spradley, et al. used a space grid with 22 nodes and a nondimensional time increment of 0.015 (case (a)). For comparative purposes it was found convenient to vary the space and time increments by multiples of 1/22 and 0.015, respectively.

The effect of decreasing the space increment for a fixed time increment is illustrated in Figure 5-3 for a

Table 5-1. Space and Time Increments Used in Numerical Computations

Case	Δy	Δt	$\Delta t/\Delta y$	$\Delta t/(\Delta y)^2$
(a)	1/22	0.015	0.33	7.26
(b)	1/22	0.005	0.11	2.42
(c)	1/44	0.005	0.22	9.68
(d)	1/44	0.00375	0.165	7.26
(e)	1/44	0.0015	0.066	2.90
(f)	1/88	0.005	0.44	38.72
(g)	1/88	0.00375	0.33	29.04
(h)	1/88	0.001875	0.17	14.52
(i)	1/88	0.0015	0.132	11.62
(j)	1/88	0.0009375	0.0825	7.26
(k)	1/110	0.003	0.33	36.30
(l)	1/176	0.0025	0.44	77.44
(m)	1/220	0.0015	0.33	72.60

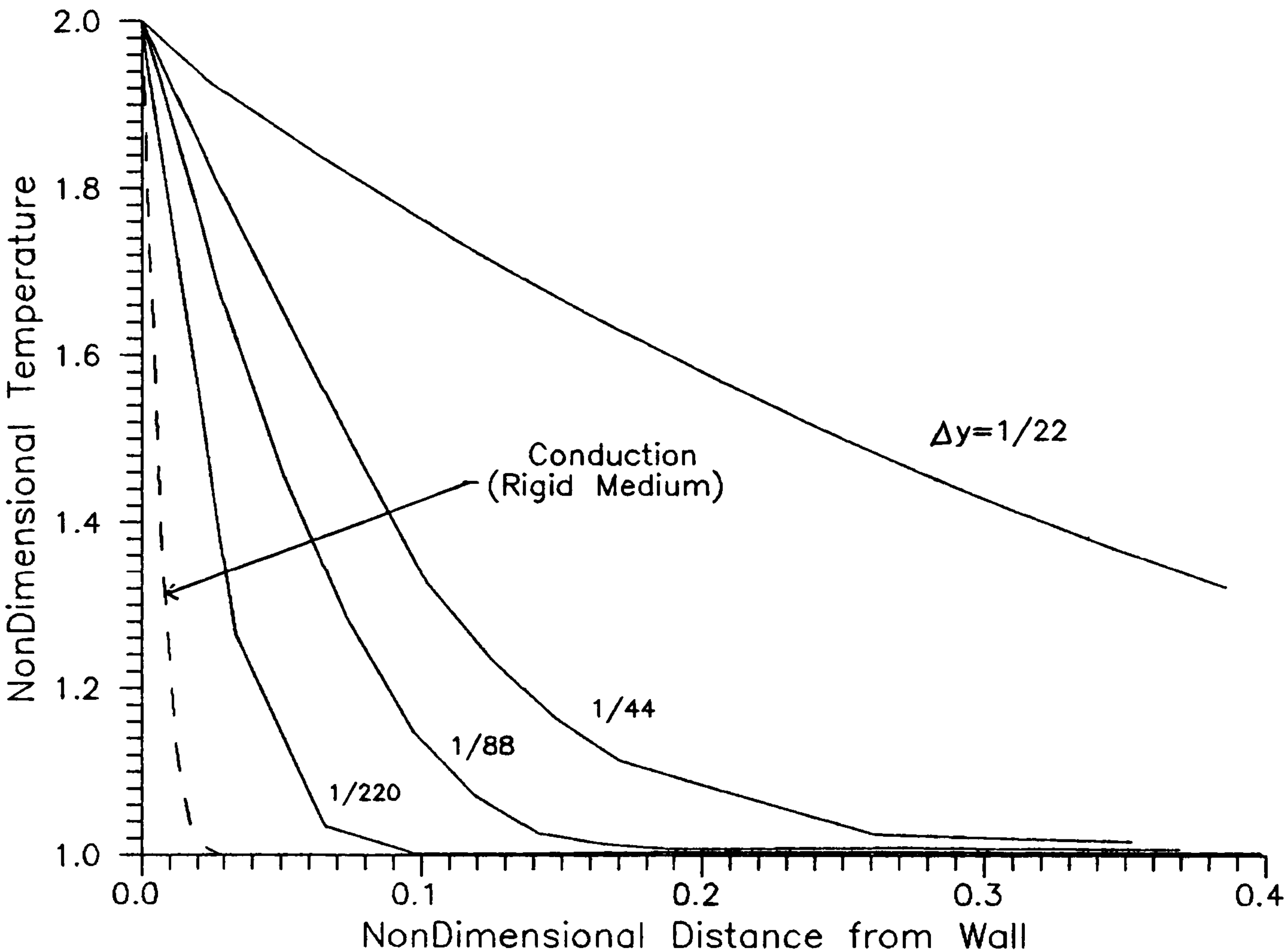


Figure 5-3. Temperature Distribution for Various Mesh Sizes, t'=0.1 Seconds, $\Delta t=0.0015$

total computation time of 0.1 seconds. The spatial temperature distribution shows significant change with decreasing space increment. The profile approaches the results for pure conduction in an equivalent rigid material and indicates that with more grid refinement the importance of convective effects diminishes. This suggests that the large difference between the computational results of Spradley, et al. and the pure conduction model may be largely due to numerical error. It appears that the relatively coarse grid is not able to accurately model the combined conduction and convection phenomena occurring near the wall. The dependence on Δy may be explained by considering the manner in which the governing equations are discretized adjacent to the wall. When the velocity at node 1, adjacent to the wall, is positive (moving away from the wall) the convective term in the energy equation $v(\partial T/\partial y)$ is modeled as

$$\frac{v_1 (T_w - T_1)}{(\Delta y/2)} \tag{5-6}$$

where the subscripts w and 1 indicate the wall and node 1, respectively. This term provides a positive convective transfer of energy (in addition to conduction) from the wall to node 1 even though the velocity at the wall is known to be zero. The heat transfer from the wall, therefore, is overestimated by this model. Furthermore, the overestimation of the heat transfer to node 1 results in an overestimation of the velocity at node 1 which amplifies the error in the convective term. Reducing the grid size minimizes the error, but the grid size required for sufficient accuracy is not known since reducing Δy below 1/220 requires a prohibitive amount of computer time.

The effect of decreasing the time step for a fixed space increment was also investigated. Typical results of temperature profiles for different time steps are shown in Figure 5-4. It is observed that the results are also dependent on the selected time step, however, it will be shown that this dependence can be eliminated successfully through modifications of the numerical scheme.

The effect on the velocity profile produced by variation in the spatial and temporal mesh size can not be clearly defined. While there are small changes in the pattern of the velocity profile, the amplitude of the velocity remains unchanged. A typical velocity profile (for $\Delta y = 1/88$, $\Delta t = 0.009375$) at the midpoint between the walls is shown in Figure 5-5. The unstable jittery behavior of the velocity profile seems to be characteristic of the explicit numerical scheme employed by Spradley, et al. This will be demonstrated by comparison with an implicit scheme.

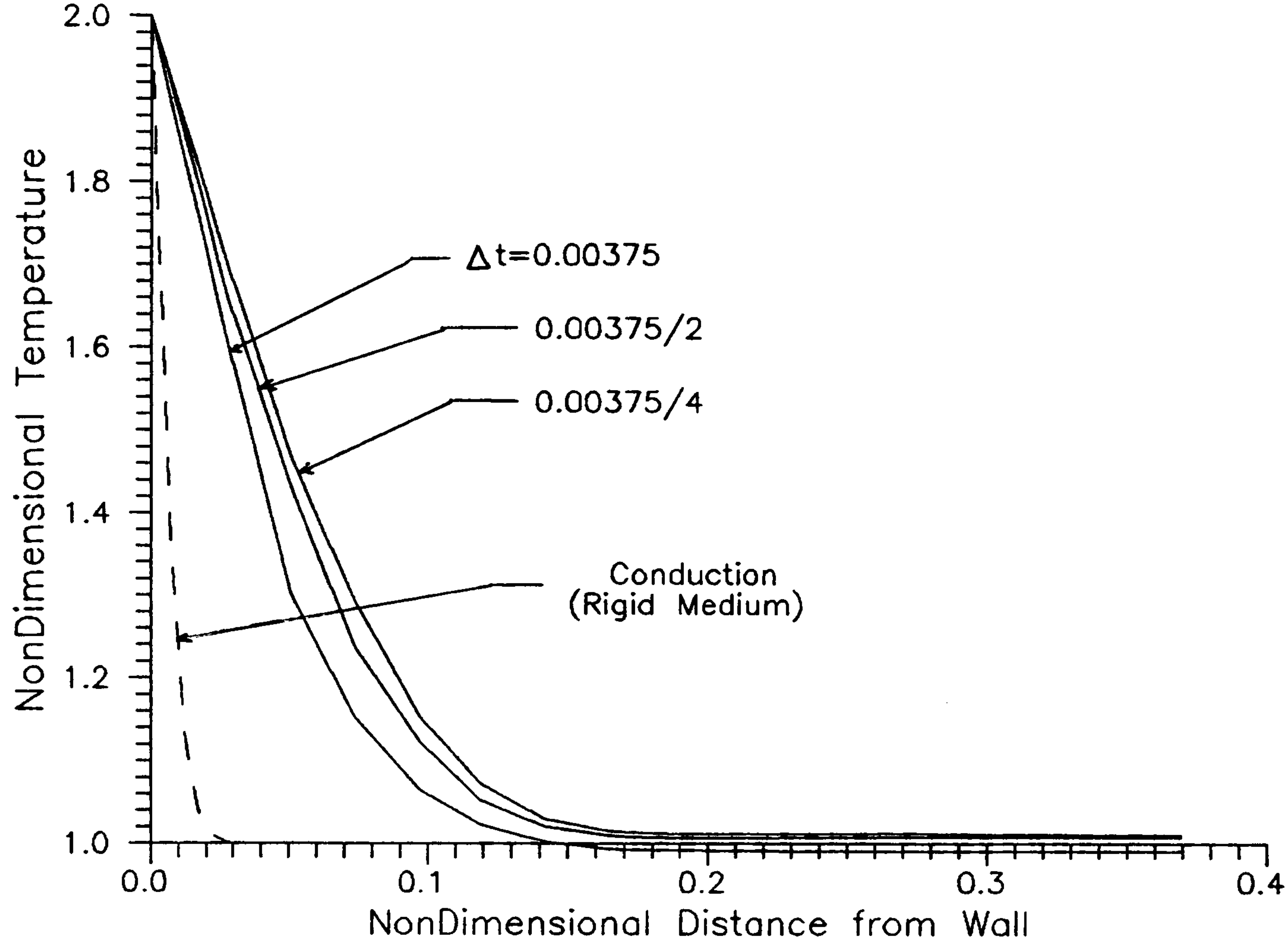


Figure 5-4. Temperature Distribution for Various Time Steps, t'=0.1 Seconds, $\Delta y=1/88$

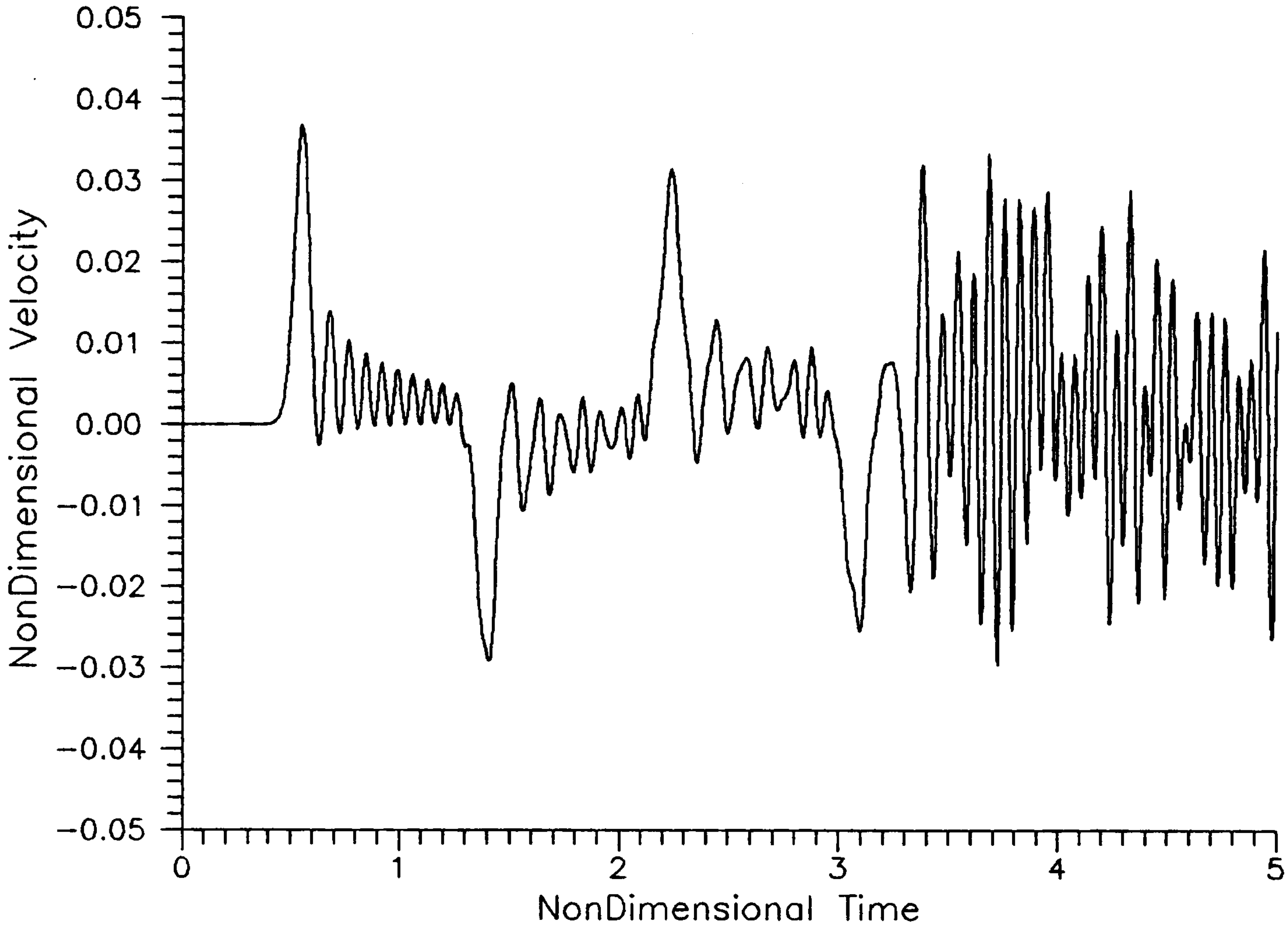


Figure 5-5. Unsteady Velocity at the Midpoint Between Plates, $\Delta y=1/88$, $\Delta t=0.0009375$

Upwind Difference Modification

An important feature of the numerical method of Spradley, et al. is the use of the upwind difference approach in the momentum and energy equations. The upwind difference scheme, which is applied to the convective terms in these equations, provides accurate results when the flow direction on opposite sides of a node are in the same direction. However, because of the oscillating nature of the acoustic velocity field, the velocity on opposing sides of any given node may often be in opposite directions. In this situation, the upwind scheme fails and may cause an accumulated error that is unacceptable.

This problem was corrected by modifying the method so that for the nodes at which the velocity changes direction for any given time step, the upwind difference scheme is replaced with the central-difference formulation. With this modification, the numerically calculated temperature field becomes independent of the size of the time increment. The temperature profile as a function of time is presented in Figure 5-6 for various Δt (same as in Figure 5-4) and a fixed Δy of 1/88. Comparison of these results with those of Figure 5-4 indicates that the error associated with the upwind scheme at the nodes where the velocity changes sign is significant. This modification appears to eliminate the dependence of the results on the size of the time increment Δt. The dependence of the

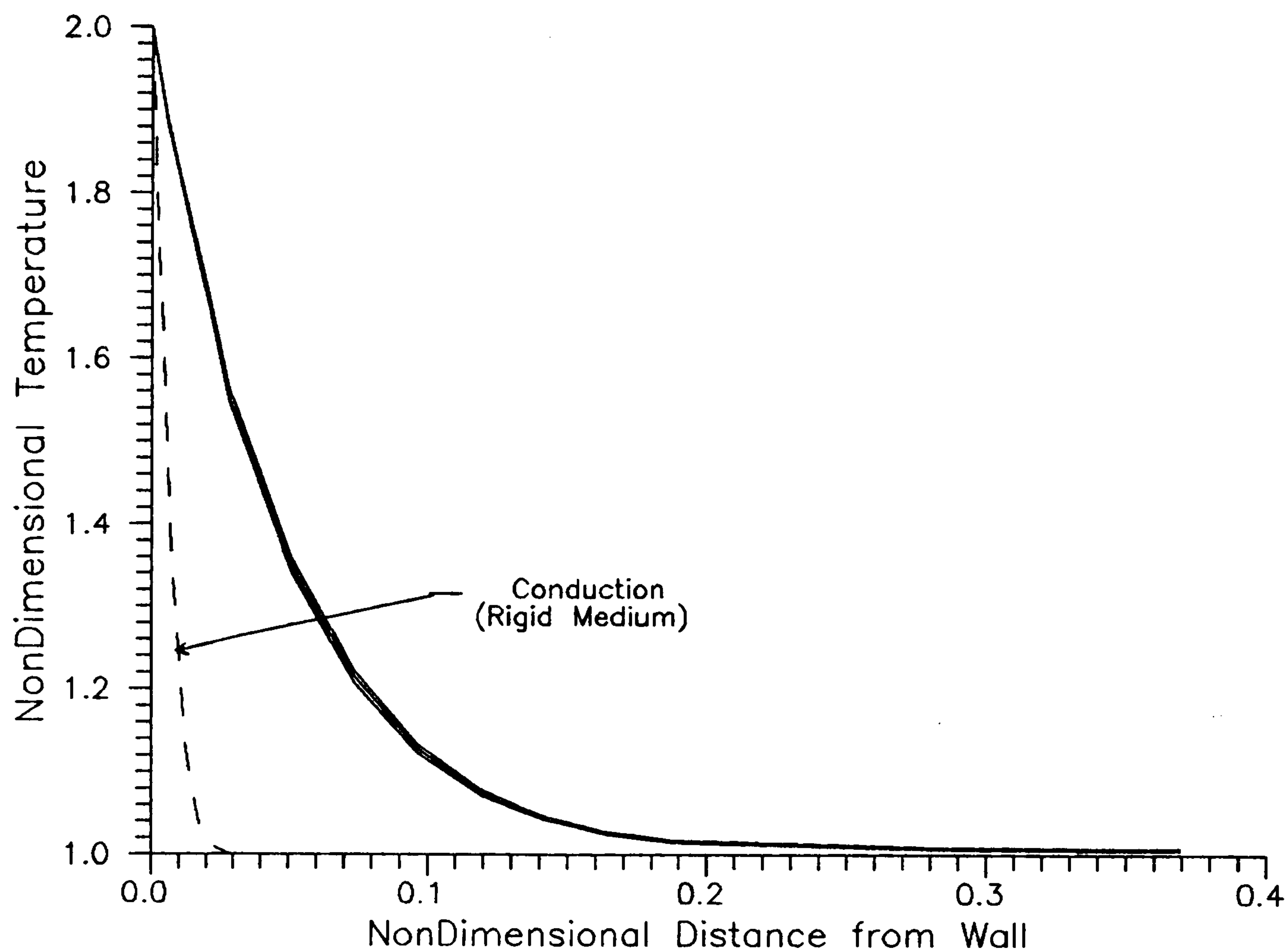


Figure 5-6. Temperature Distribution for the Same Time Steps as in Figure 5-4 with Upwind Difference Modification, t'=0.1 Seconds, $\Delta y=1/88$

results on the spatial grid size Δy, however, was not significantly affected by this modification.

The effect of this center-difference modification on the velocity profile is shown in Figure 5-7. The amplitude, frequency and character of the velocity profile are initially similar when compared with Figure 5-5, but do not become as apparently unstable as time progresses.

Since this modification effectively eliminates the size of Δt as a factor in the refinement of the temperature solution, it was retained as a permanent feature of the numerical model. That is, all modifications and the corresponding results presented in the remainder of this discussion include the implementation of this feature.

Second-Order Accuracy at the Wall

Accurate modeling of the physics of the heat and mass transfer adjacent to the wall appears to be critical to the overall success of the numerical model. Therefore, it is desirable to have second-order accuracy for all gradient terms in the equations. Spradley, et al. accomplish this in all cases except for the pressure gradient term adjacent to the wall in the momentum equation where a first-order forward-difference scheme is used so that

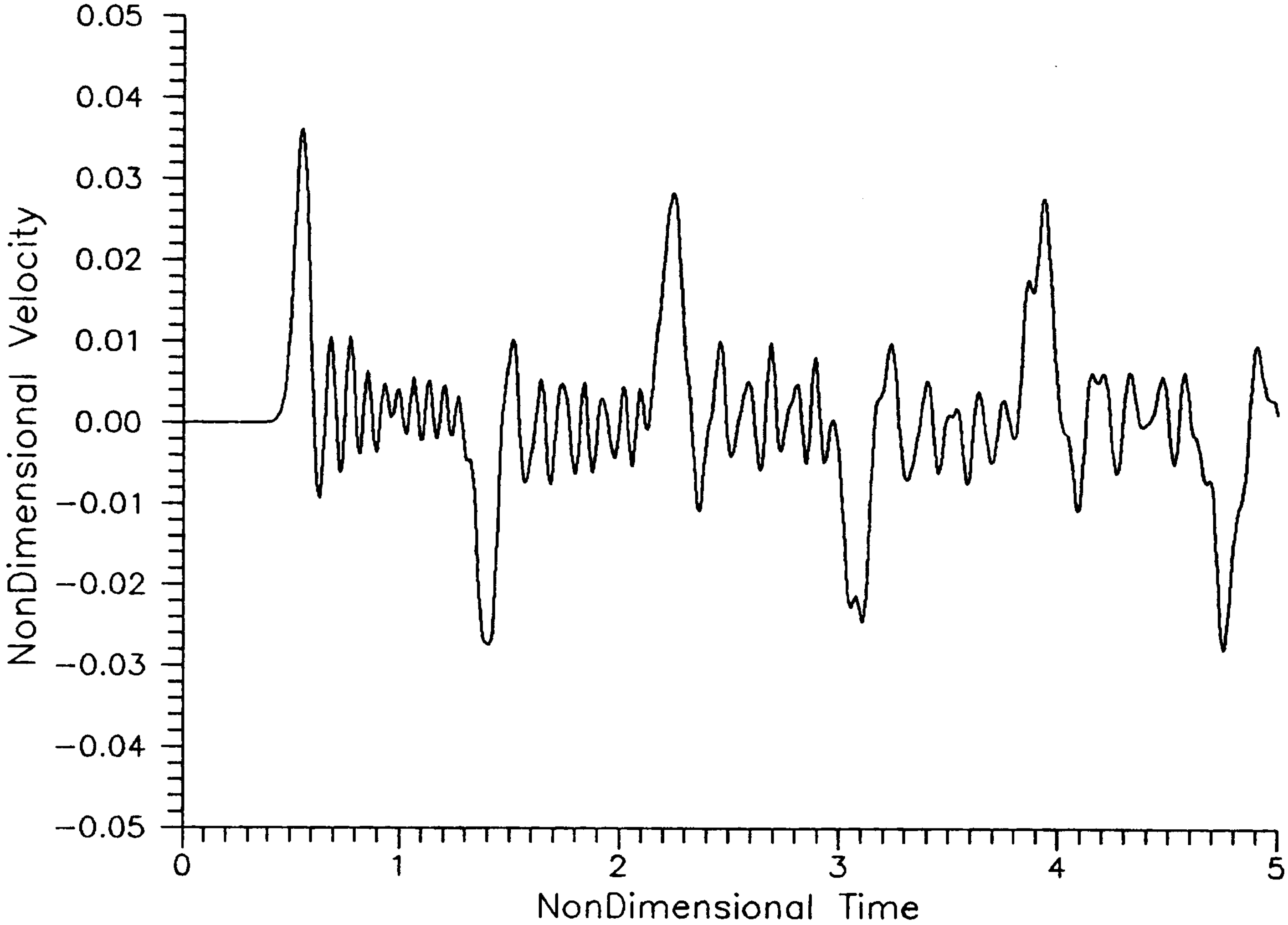


Figure 5-7. Unsteady Velocity at the Midpoint Between Plates with Upwind Difference Modification, $\Delta y=1/88$, $\Delta t=0.0009375$

$$\left.\frac{\partial P}{\partial y}\right)_1 = \frac{P_2 - P_1}{\Delta y} \qquad (5\text{-}7)$$

This has been modified to achieve second-order accuracy by fitting a quadratic curve to the first three nodes adjacent to the wall and determining the derivative at node 1 from that curve as

$$\left.\frac{\partial P}{\partial y}\right)_1 = \frac{1}{\Delta y}\left(-\frac{1}{2}P_3 + 2P_2 - \frac{3}{2}P_1\right) \qquad (5\text{-}8)$$

Selected results for temperature distribution at time t=0.1 seconds are shown in Figure 5-8. Comparison with Figure 5-3 clearly shows that the higher order accuracy at the wall has a significant effect on the results. The temperature profile is now considerably closer to the conduction-only curve.

Figure 5-9 shows the effect of the wall pressure gradient modification on the velocity profile. Here the velocity as a function of time is presented at both the first grid point near the heated wall and at the midpoint between the two walls. The amplitude of the velocity oscillations have decreased by one order of magnitude compared to the case shown in Figure 5-7. Also, a small net bulk flow away from the wall is observed to be present to satisfy mass conservation as a result of large density changes near the hot wall.

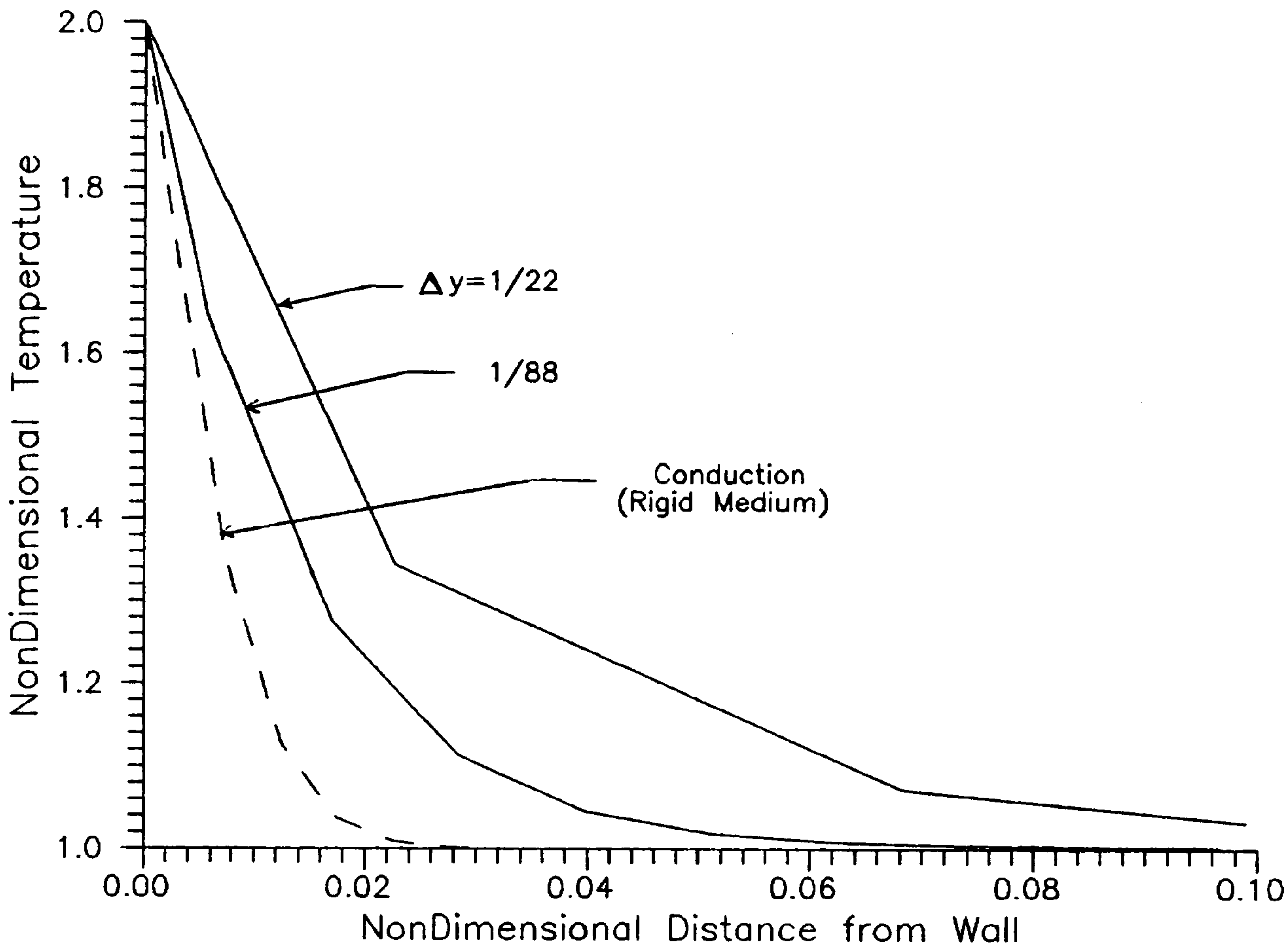


Figure 5-8. Temperature Distribution for Various Mesh Sizes with Second Order Accuracy at the Wall, $t'=0.1$ Seconds, $\Delta t/\Delta y=0.33$

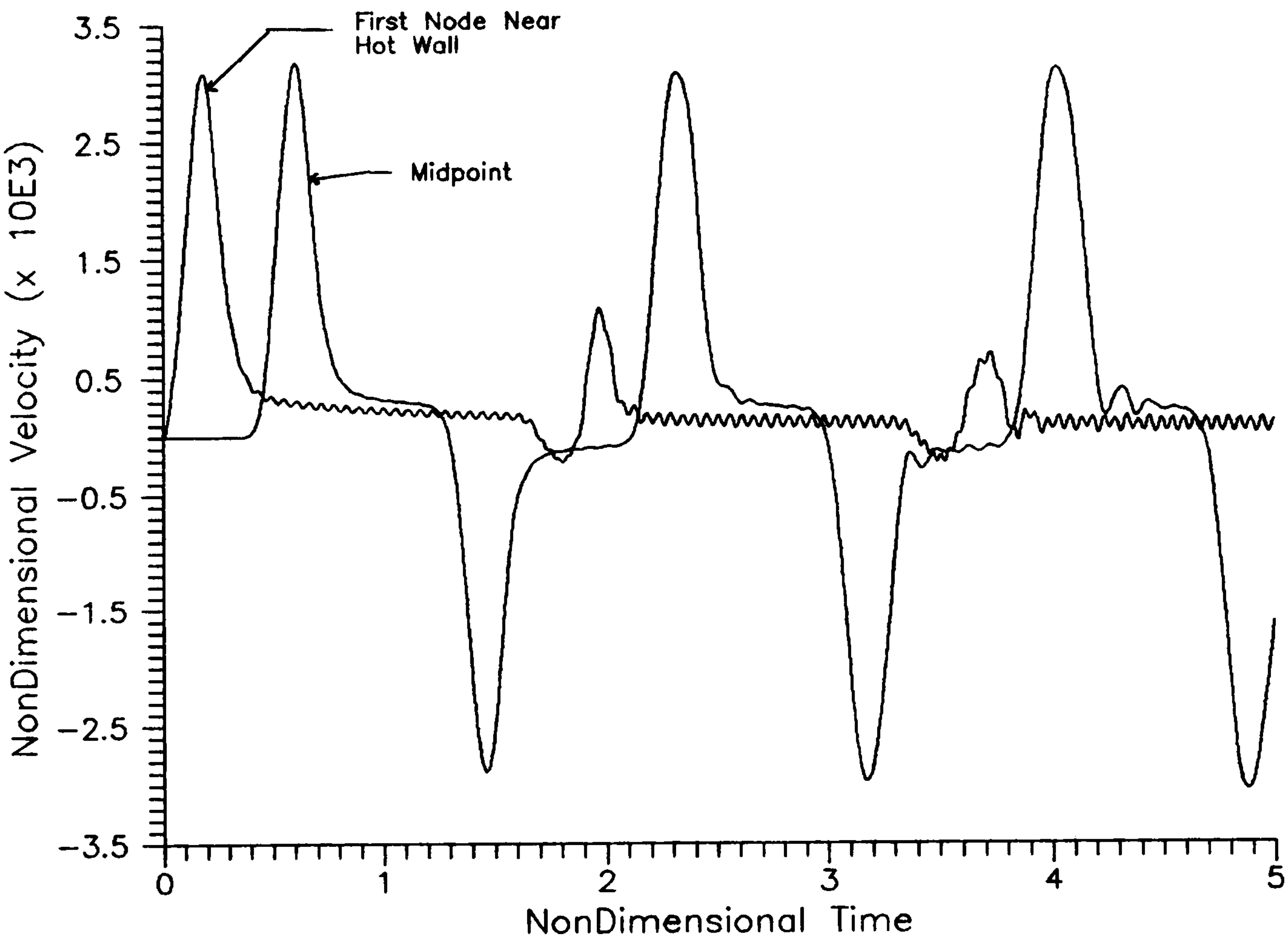


Figure 5-9. Unsteady Velocity Near the Wall and at the Midpoint with Second Order Accuracy at the Wall, Explicit, Δy=1/88, Δt=0.005

It is clear that improving the accuracy of the modeling of the pressure gradient at the wall has a substantial effect on the numerical results. The modification resulted in a large decrease in the magnitude of the fluid velocity away from the wall and therefore a large decrease in the convective heat transfer away from the wall. (Spradley and Churchill [29] incorporated a different second-order method for the wall pressure gradient, but the change, surprisingly, had no discernable effect on the results as compared to Reference 28.) Since this modification produces improved accuracy, it was permanently incorporated into the modified numerical model.

Implicit Scheme

The numerical method previously discussed employs an explicit scheme in time which restricts the choice of the time increment for a given size of space increment. When these limits are exceeded convergence is impossible. However, even before reaching these limits, the solution shows instabilities that, while they may remain bounded, affect the accuracy and reduce confidence in the results. For this reason, the explicit marching scheme in time was eliminated in favor of an implicit method in time.

The change to the implicit scheme removes the need for imposing a condition on the space and time increments as required by the explicit method. However, the physics of the flow requires the time step to be small enough so that the traveling of a pressure wave will be numerically resolved. Therefore, while no strict requirements for numerical stability exist in the implicit scheme, it was not possible to significantly increase the time step Δt. As a result, no savings in computational time were accomplished compared to the previous explicit method.

The results for temperature and velocity profiles are presented in Figures 5-10 and 5-11, respectively. Comparison of Figures 5-10 and 5-8 shows that the temperature profiles are generally closer to the conduction-only results in the implicit case. Comparison of Figures 5-11 and 5-9 shows that the amplitude of the velocity profile remains unchanged (for the given Δy and Δt) between the explicit and implicit cases. However, the instability of the velocity profile is observed to decrease significantly. This smooth character of the velocity profile is a feature of the employed implicit scheme since it does not change for any combination of space and time increments. A decline in the amplitude of the velocity profile is observed which is not the result of viscous decay. To verify this, the viscous term was removed from the equations and the same rate of velocity decline was observed. This damping of the velocity amplitude is presumably caused by the redistribution of the fluid mass in the enclosure and is required by the conservation of mass.

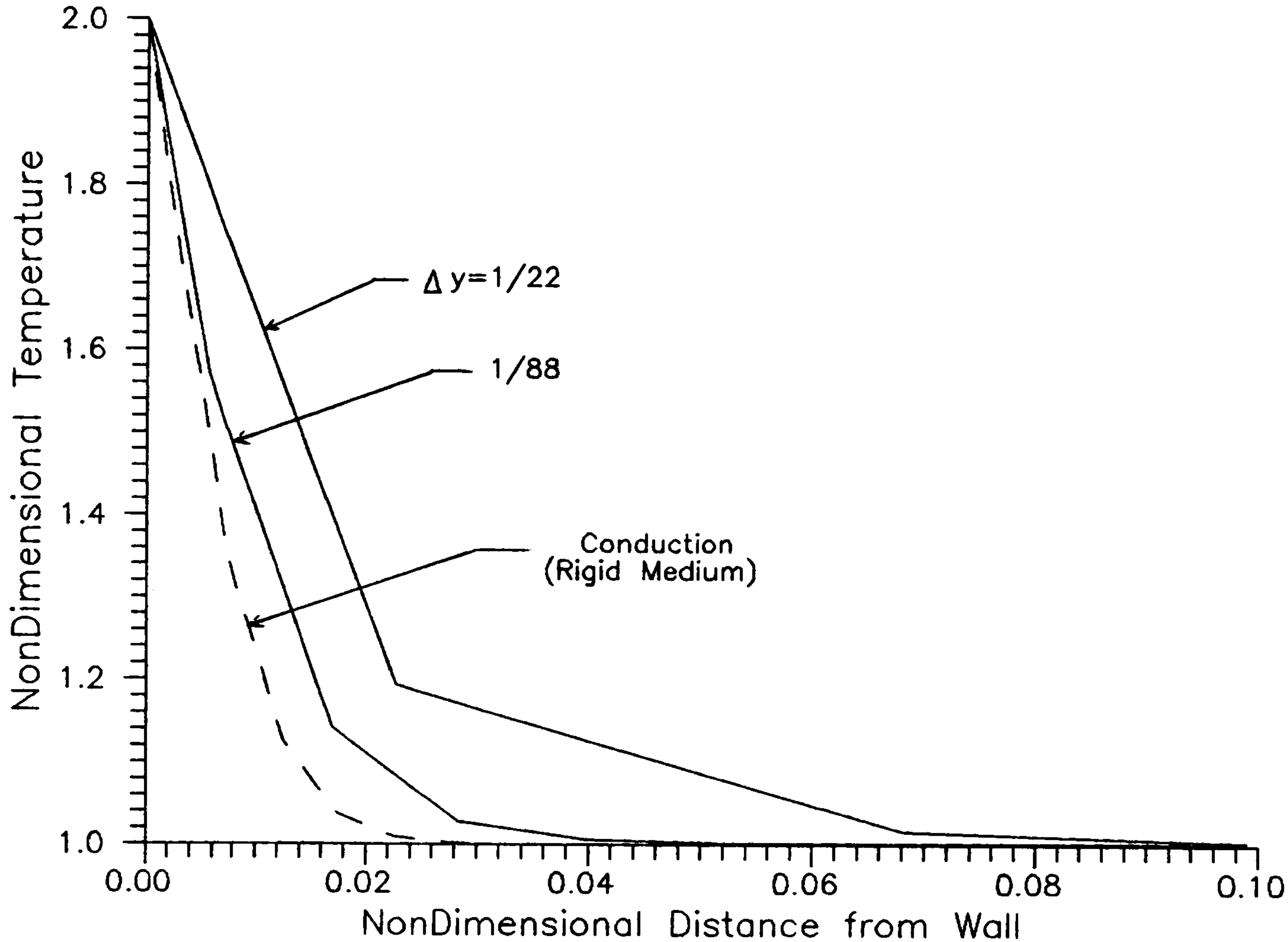


Figure 5-10. Temperature Distribution for Various Mesh Sizes, Implicit, t'=0.1 Seconds, Δt=0.005

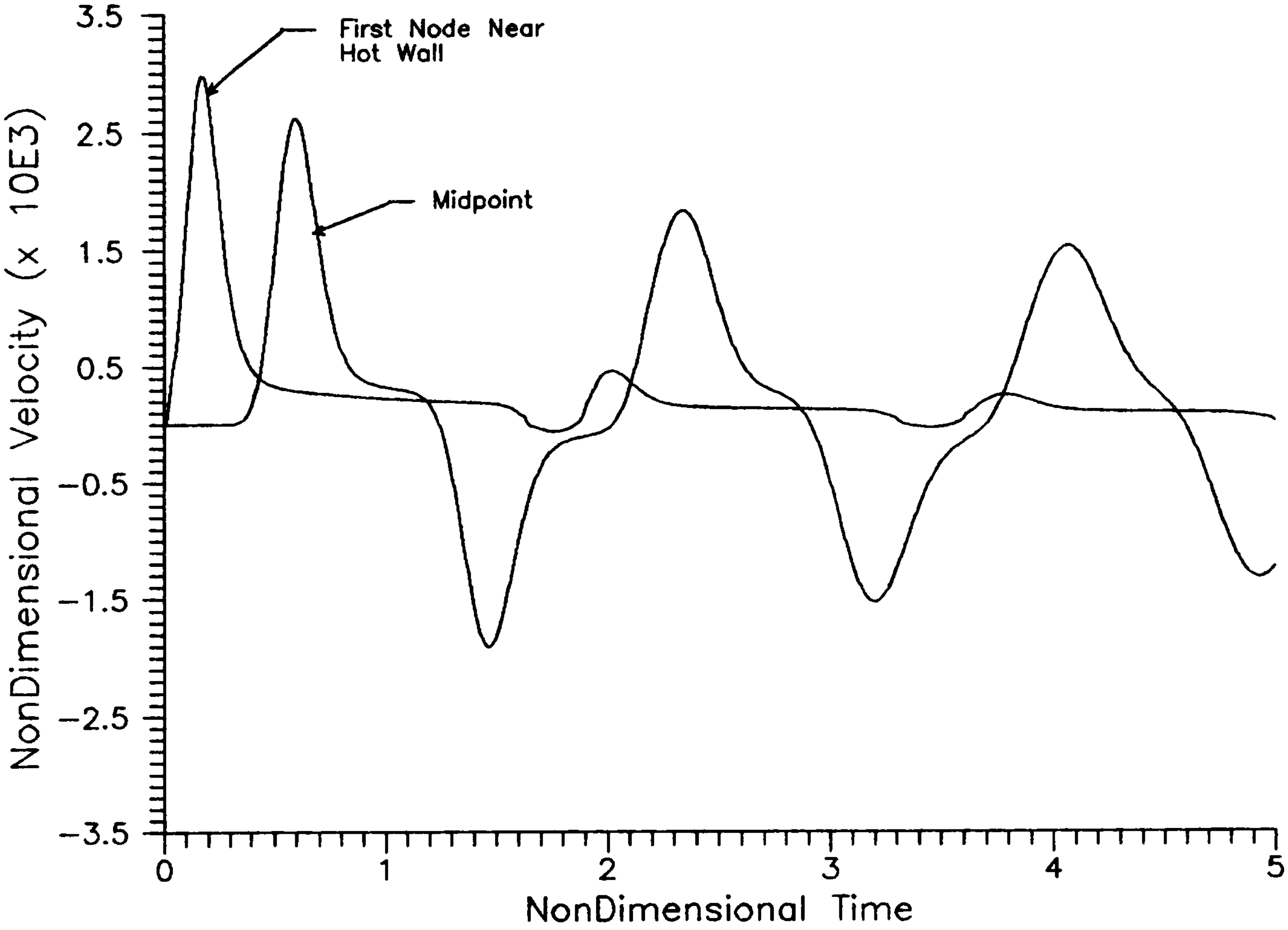


Figure 5-11. Unsteady Velocity Near the Wall and at the Midpoint, Implicit, Δy=1/88, Δt=0.005

Concluding Remarks on Previous Numerical Work

The numerical scheme of Spradley, et al. has been analyzed and modified in several ways in an attempt to improve the prediction of heat transfer to a rapidly heated gas. Significant improvement has been accomplished by modification of the upwind difference scheme, second order accurate modeling of the pressure gradient at the wall and the use of an implicit solution method. These improvements as well as the spatial grid size analysis demonstrate that the original numerical scheme of Spradley, et al., which did not contain a grid refinement analysis, fails to accurately model the phenomenon. It must also be pointed out that the dependence of the results on spatial grid size remains strong in spite of the revisions discussed above. The necessity of completing a grid refinement study in the application of this revised numerical scheme remains as valid as before.

5.4 Expanding Conduction Model

It is possible that the acoustic pressure waves (or velocity oscillations) have no direct impact on the transient development of the temperature field. The pressure waves serve the purpose of maintaining a spatially homogeneous pressure rise in the enclosure as they travel back and forth. However, the magnitude of the waves may be too small to have an impact on the heat transfer process. (The relative magnitude of the pressure waves is discussed in more detail in Appendix C.) If the acoustic waves are not important, thermally-induced convection heat transfer can be modeled numerically in a much simpler way than that discussed in the previous sections of this chapter.

Based on this possibility, a simplified numerical model was developed that eliminates the acoustic velocity field but considers the bulk motion of the fluid as it expands and contracts in response to the conduction-dominated temperature field and the spatially homogeneous pressure rise. This is accomplished by eliminating the momentum equation from consideration and adding the requirement that pressure be a function of time only. Eliminating the momentum equation from the set of governing equations greatly simplifies the numerical problem. Pressure waves traveling at the speed of sound no longer need to be resolved spatially or temporally. The velocity normal to the heated wall is calculated in a simple way by requiring that continuity be satisfied rather than from Newton's second law.

Consider an ideal gas enclosed by two infinitely long flat plates as described in Section 5.2. The initial temperature is T_o' and at time t=0 one plate is instantly raised to temperature T_w'. In this case, the fluid thermal properties k', c_v', and γ are a function of temperature. Viscosity is not considered since the momentum equation is not used. Also, the slower conduction time scale is used to normalize the equations since the acoustic time is not relevant in this model. Space, time, temperature, velocity, pressure and density are normalized with H', H'^2/α_o', T_o', α_o'/H', P_o' and ρ_o', respectively, where $\alpha_o' = k_o'/\rho_o' c_{v_o}'$ is the thermal diffusivity. With these assumptions, the normalized equations become

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial y} = 0 \tag{5-9}$$

$$\rho \frac{\partial (c_v T)}{\partial t} + \rho v \frac{\partial (c_v T)}{\partial y} = -(\gamma - 1) P \frac{\partial v}{\partial y} + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) \tag{5-10}$$

$$P = \rho T \tag{5-11}$$

The pressure at any time can be found from

$$P = \int_0^1 \rho T \, dy \tag{5-12}$$

These four equations are solved using an implicit finite volume numerical scheme. In this scheme the unknown temperature, pressure and density are calculated at the node points while the velocity is calculated at the node boundaries. The convective and conductive flux terms in the energy equation are evaluated at the node boundaries. This practice ensures that the model will be physically realistic. Details concerning the finite volume discretization technique are clearly explained by Patankar [35].

The coupled nonlinear equations require an iterative procedure for solution. The steps of the solution procedure are as follows:

(a) Solve the energy equation (5-10) for temperature using values for v, P and ρ from the previous time step. For the one-dimensional case, the temperature is solved for by using a tri-diagonal matrix solver.

(b) Solve Equation (5-12) for the pressure by taking the average value of the product ρT over all the nodes.

(c) Solve for density using the ideal gas state equation (5-11) and the newly calculated temperature and pressure.

(d) Return to step (b) and repeat until the solution for pressure and density converges.

(e) Solve the continuity equation (5-9) for the velocity at the node boundaries.

(f) Return to step (a) and repeat until the entire solution converges for the present time step.

(g) Advance to the next time step and repeat steps (a) through (f).

A significant advantage of this model is that its numerical solution is relatively simple compared to the previous model which includes the compressible momentum equation and its associated wave phenomena.

Temperature profiles calculated using this expanding conduction model are shown in Figure 5-12 at various nondimensional times. In this case, the nondimensional grid size and time step were $\Delta y=1/50$ and $\Delta t=0.01$, respectively; variations of the grid size and time step did not produce significant changes in the results. For air initially at standard conditions and $H'=4$ inches, a nondimensional time unit is equal to 342.97 seconds real time. Temperature profiles for an equivalent rigid material are also shown in Figure 5-12. It is observed that for this geometry, the temperature of the expanding gas is always higher than that of the corresponding rigid material. The energy transfer from the heated wall, however, is actually less for the expanding gas case since the density, and thus the heat capacity, is lower near the heated wall. Of course, once the final steady state temperature is reached, there will be no difference between the expanding gas and rigid material temperature profiles. The bulk velocity for the expanding gas model is shown in Figure 5-13. A nondimensional velocity unit

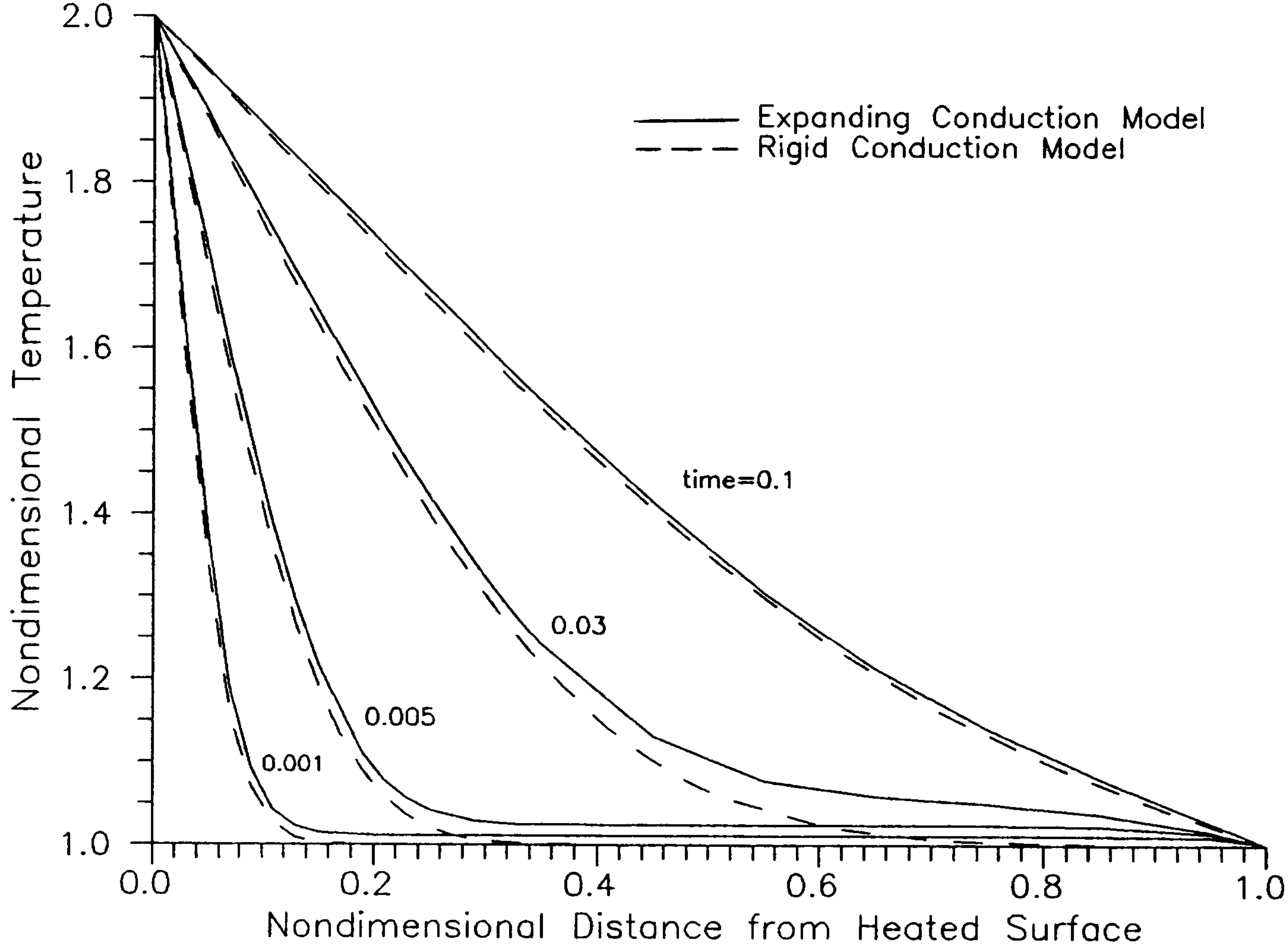


Figure 5-12. **Temperature Distribution at Various Nondimensional Times for the Expanding Conduction Model and Rigid Material Model**

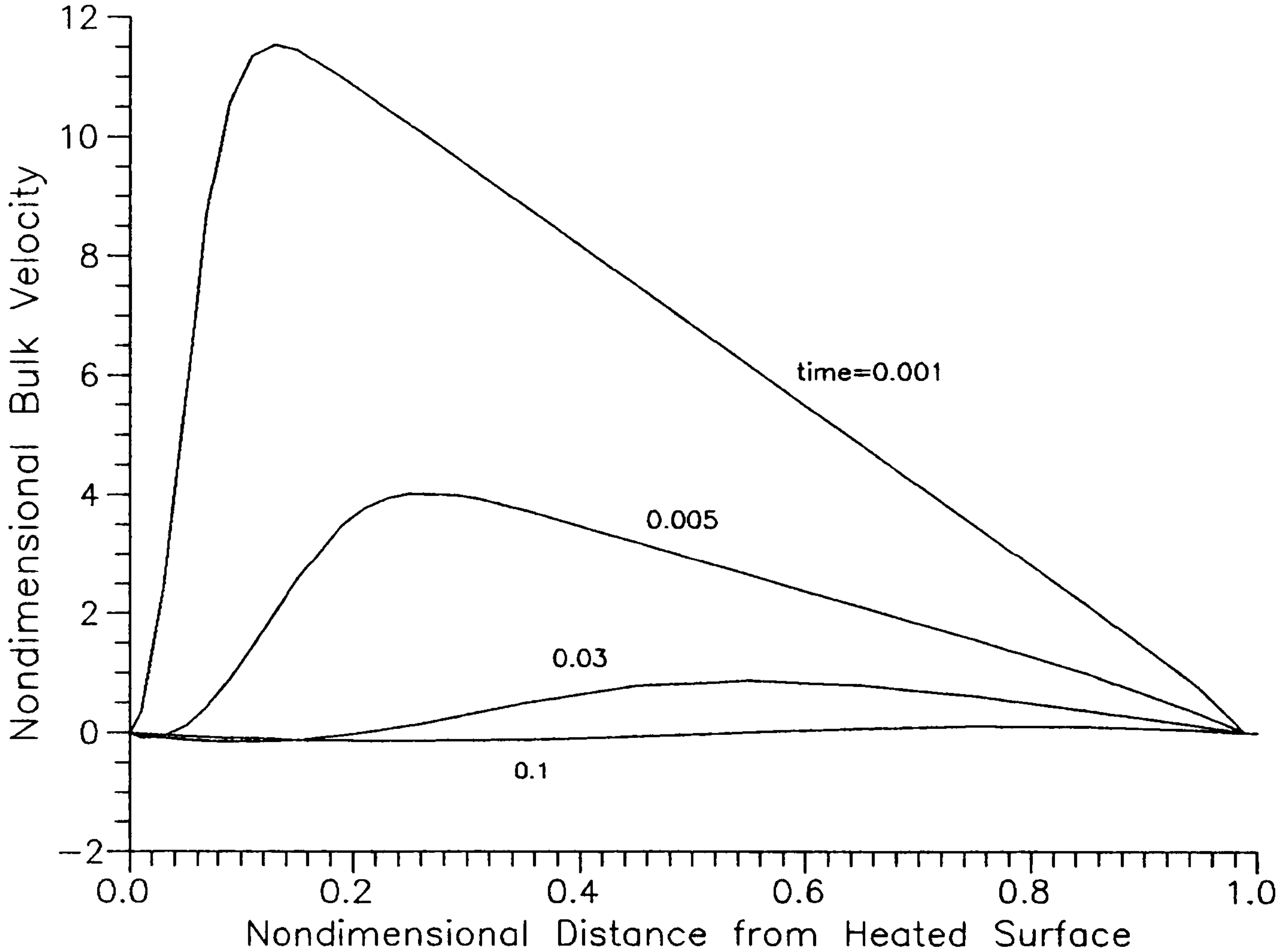


Figure 5-13. Bulk Velocity at Various Nondimensional Times for the Expanding Conduction Model

is equal to 0.6994 inches per minute for standard air and H'=4 inches. The velocity is highest near the heated wall at short times. As time progresses, the peak velocity occurs farther from the wall and begins to die out. At the nondimensional time of 0.005, a small flow back toward the heated wall to satisfy continuity is observed. The steady state condition is zero velocity everywhere.

The expanding conduction model was extended to three dimensions for the conduction terms in the energy equation. Bulk velocity, however, was assumed to be entirely in the y direction normal to the heated wall. (It is, in fact, not possible to add the other velocity components since the additional momentum equations can not be introduced into the model.) It is believed that this assumption provides a reasonable three-dimensional approximation for the purpose of including side wall temperature effects as long as temperature gradients parallel to the heated wall (x and z directions) are not too great. The results of the three-dimensional model are compared with experimental results in Chapter 6.

CHAPTER 6

COMPARISON OF EXPERIMENTAL AND NUMERICAL RESULTS

In this chapter, comparison is made between the results of the experimental investigation discussed in Chapters 2 and 3 and the results of the three-dimensional expanding conduction numerical model described in Chapter 5. Comparison is also made between the experimental work and the numerical model of Spradley, et al. [28]. Finally, the expanding conduction model is compared with the results of the perturbation analysis performed by Radhwan and Kassoy [22].

Experimental and numerical spatial temperature profiles are shown in Figures 6-1 through 6-4 for various times from 0.5 up to 10.0 seconds. The experimental temperature profiles are corrected such that the measured gage pressure equals the gage pressure calculated from the temperature profile. The experimental results are the same as those given previously in Figures 3-18 through 3-21. The expanding conduction numerical model is three-dimensional for the conduction terms in the energy equation (as discussed in Chapter 5). The time and spatially dependent wall temperatures (heated wall, side wall and bottom wall) were measured using thermocouples so that the actual experimental boundary conditions could be

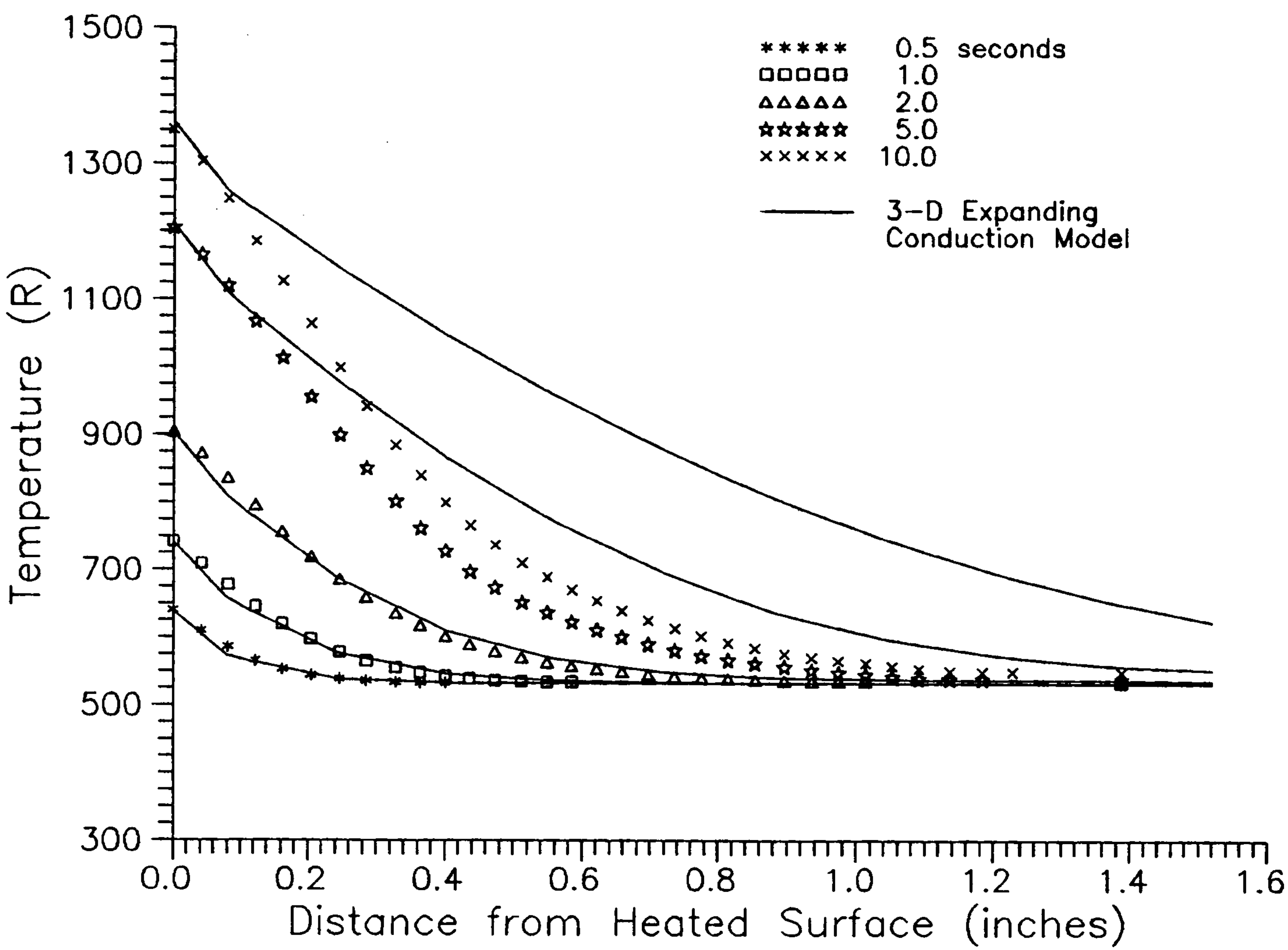


Figure 6-1. Comparison of Experimental and Numerical Temperature Results for Low Power

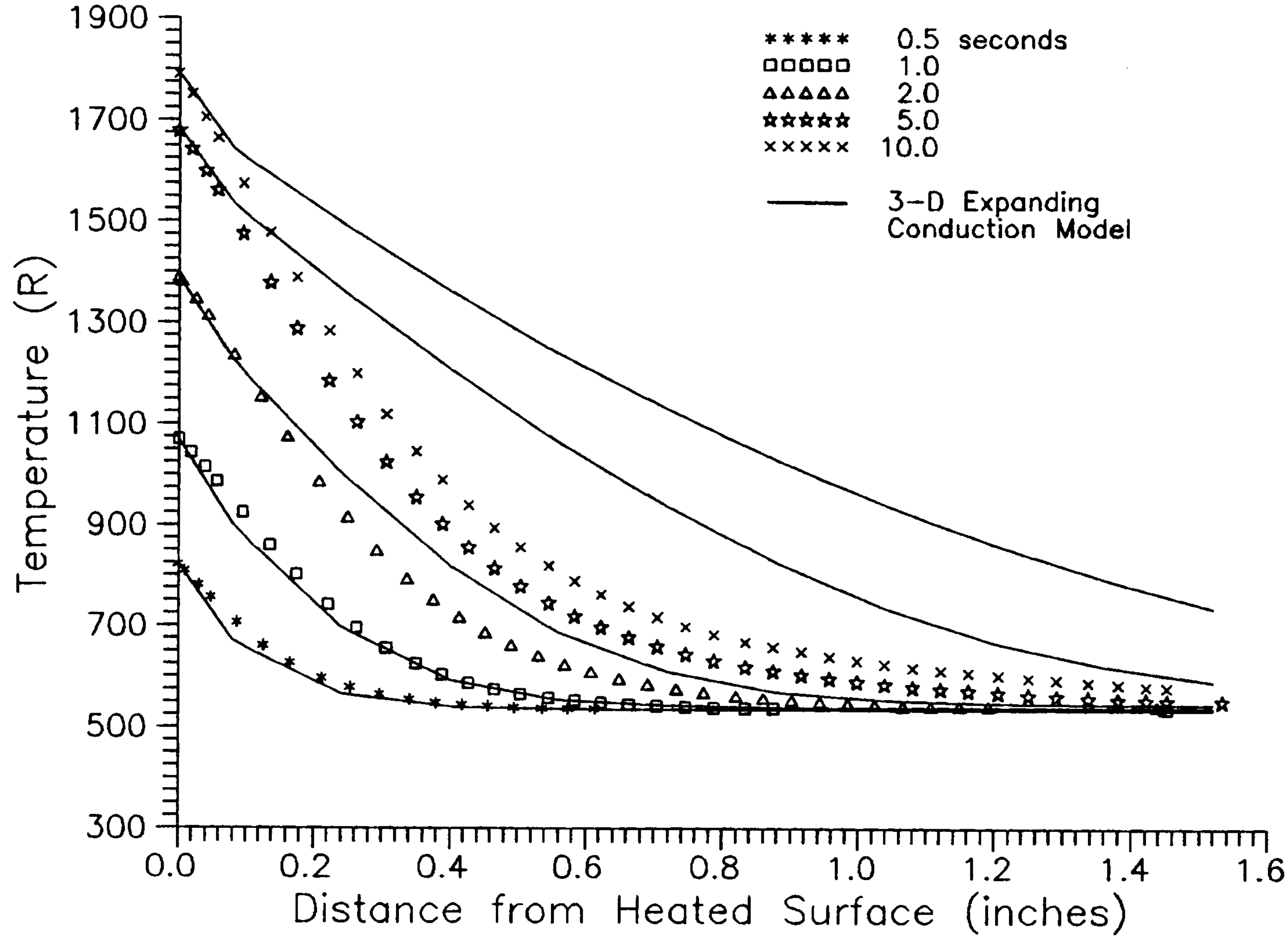


Figure 6-2. Comparison of Experimental and Numerical Temperature Results for Medium Low Power

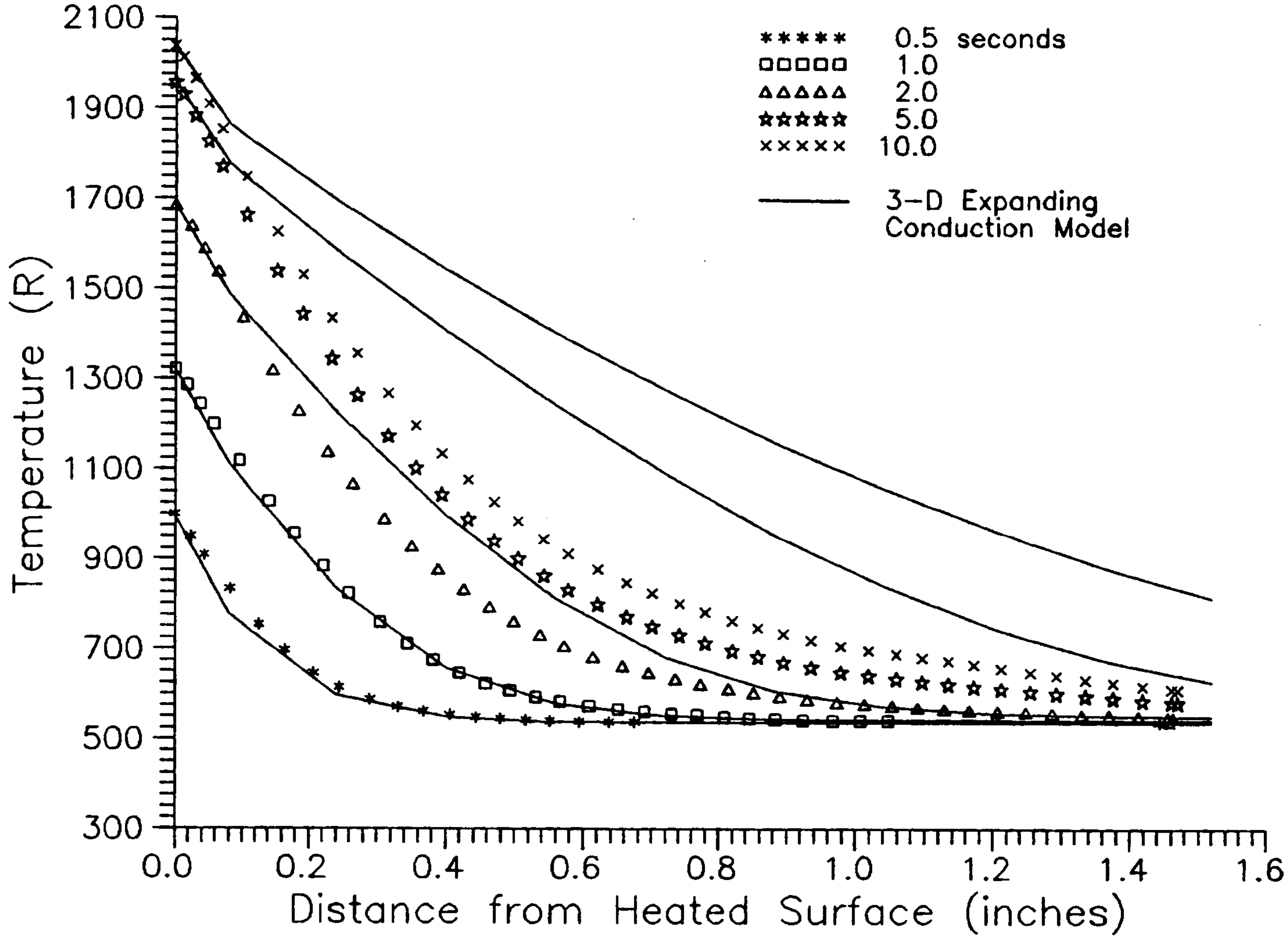


Figure 6-3. Comparison of Experimental and Numerical Temperature Results for Medium High Power

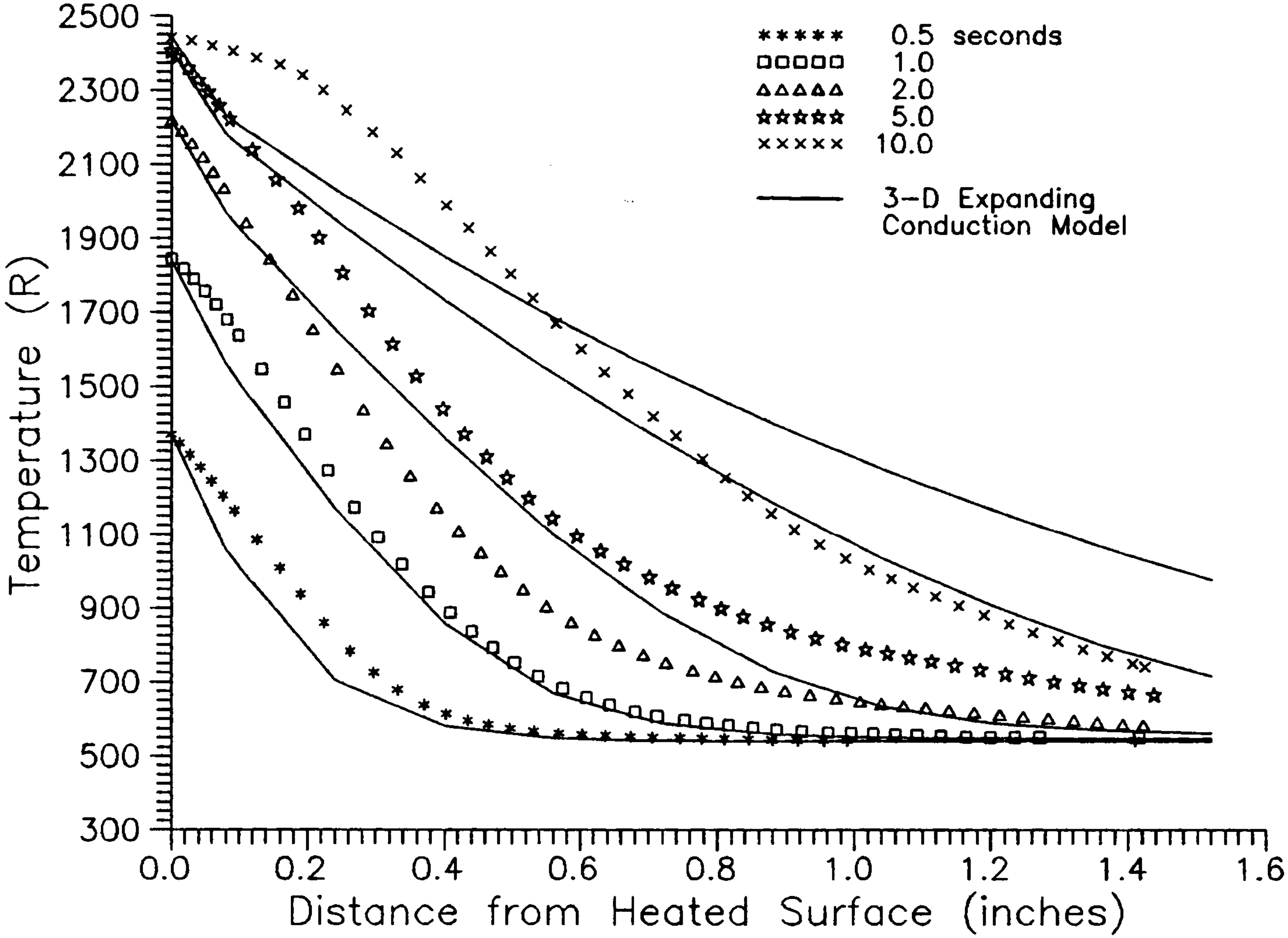


Figure 6-4. Comparison of Experimental and Numerical Temperature Results for High Power

included in the numerical model. The "extra" free volume enclosed in the test apparatus (discussed in Chapter 3) was represented by an equivalent "extra" test section height H_{ex} = 0.79H and was also considered in the numerical model. Numerical solutions were obtained both with and without the "extra" volume included and it was found that the effect on the temperature profiles was insignificant. Here, the nondimensional grid size was 1/25 in the direction normal to the heated wall and 1/9 in the other coordinate directions; the time step was 0.01. There was no significant change in the results for finer grid sizes or smaller time steps.

The experimental and numerical temperature profiles agree quite well for short times up through 1.0 seconds. After approximately one or two seconds, however, the experimental temperatures are considerably below the numerical temperature results. A possible explanation for the divergence of the results is that bulk axial flow away from the heated wall entrains cooler air from elsewhere and brings it into the heated boundary layer region near the wall. The source of the cooler air could be the porous side walls or small cavities in the spring mechanism above the heater element. Any effects caused by flow through the porous side walls were eliminated, however, since separate experiments were performed with nonporous side walls with no significant difference in the results. Another possibility is that acoustic-type waves, which are not considered in the numerical model, do indeed have a significant effect on the heat transfer and flow characteristics of the problem.

The experimentally measured pressure and the pressure calculated from the numerically determined temperature and density profiles are shown in Figure 6-5 for each power level. For times less than about two seconds, the experimental pressure is greater than the numerical pressure. At longer times the opposite is true. The experimental results are always higher than the numerical results for the high power case. The results of the high power experimental run may be erroneous, however, because of the charring of the insulation material during this run (see Section 3.4).

The experimentally measured temperature profiles at times 0.5 and 1.0 seconds and for the medium high power level are compared with the numerical model of Spradley, et al. in Figure 6-6. The numerical model uses constant fluid properties and is one dimensional, but even with these differences, it is clear that the numerical results are not verified except maybe for small times when conduction heat transfer is dominant and the heated boundary layer is restricted to a small area near the wall. Even here the better agreement at small times is probably the result of lower wall temperatures during the first few tenths of a second. This is discussed in further detail in Appendix G. After one second the

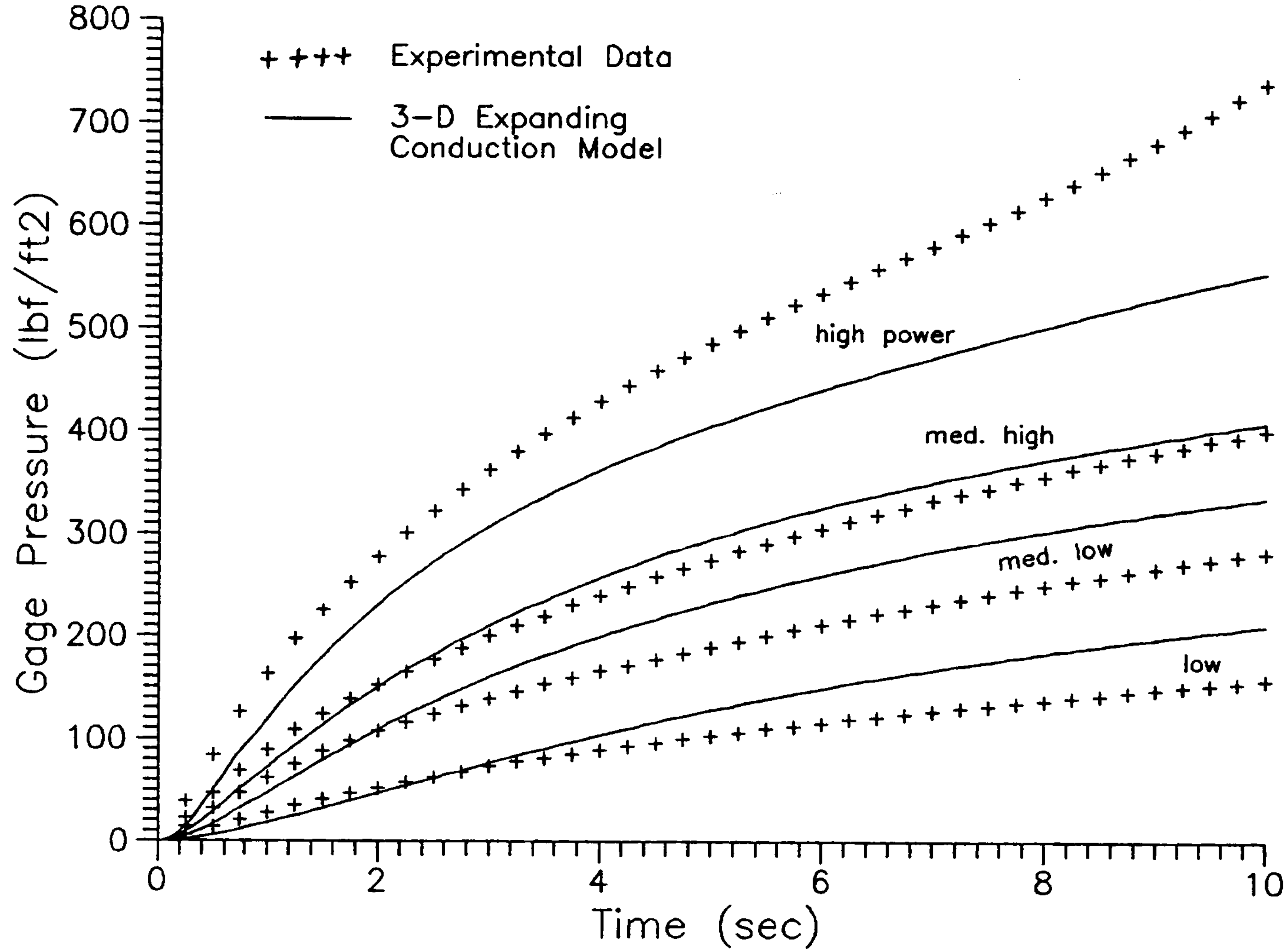


Figure 6-5. Comparison of Experimental and Numerical Pressure Results for Each Power Level

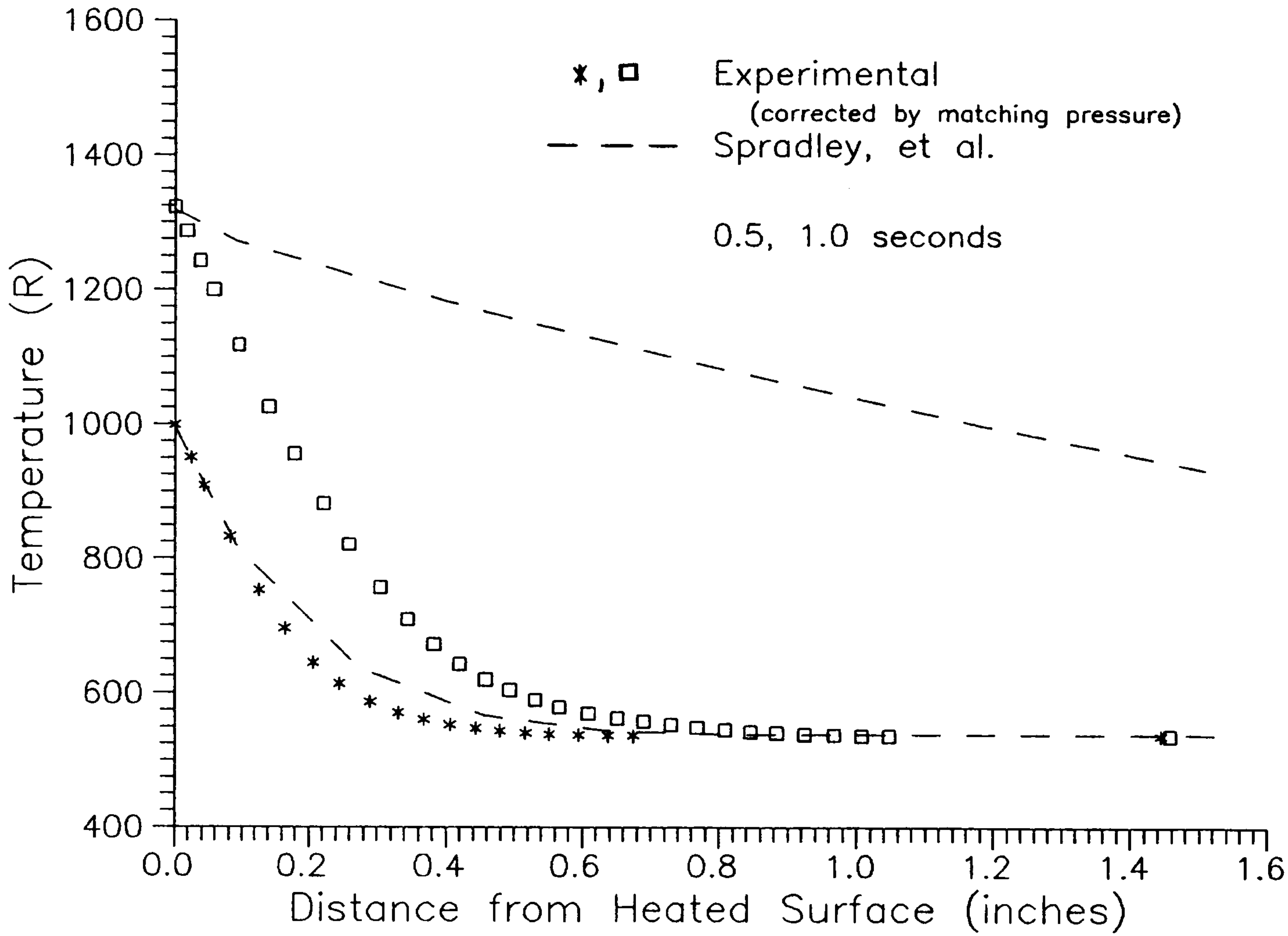


Figure 6-6. **Comparison of Experimental Temperature Profiles with Numerical Results from Reference 28 for Medium High Power**

numerical results indicate a much faster rate of temperature rise than the experiments. Furthermore, the numerical model was unstable for heated wall temperatures above about 1500°R (nondimensional temperature of about 2.8). The inadequacy of this numerical model was also demonstrated from an analytical point of view in Chapter 5.

Temperature profiles obtained by Radhwan and Kassoy are compared with temperature profiles for the expanding conduction numerical model in Figure 6-7. Radhwan and Kassoy used the singular perturbation method with a small parameter defined by the ratio of a short acoustic time scale and a long conduction time scale. It was assumed that $k=T$, $c_v=1$, and $\gamma=1.4$ for the dimensionless thermal conductivity, constant volume heat capacity, and heat capacity ratio, respectively. The boundary and initial conditions are the same as those previously described for the numerical model of Spradley, et al. except that both of the infinite plates are instantly elevated to a nondimensional temperature of 2.0 at time $t=0$. For the purposes of comparison, the same fluid properties and boundary and initial conditions are applied to the one-dimensional expanding conduction model. Here the nondimensional time unit τ is equal to 480.17 seconds real time for air initially at standard conditions and $H'=4$ inches. This is a factor $\gamma=1.4$ greater than the nondimensional time unit previously used in the expanding

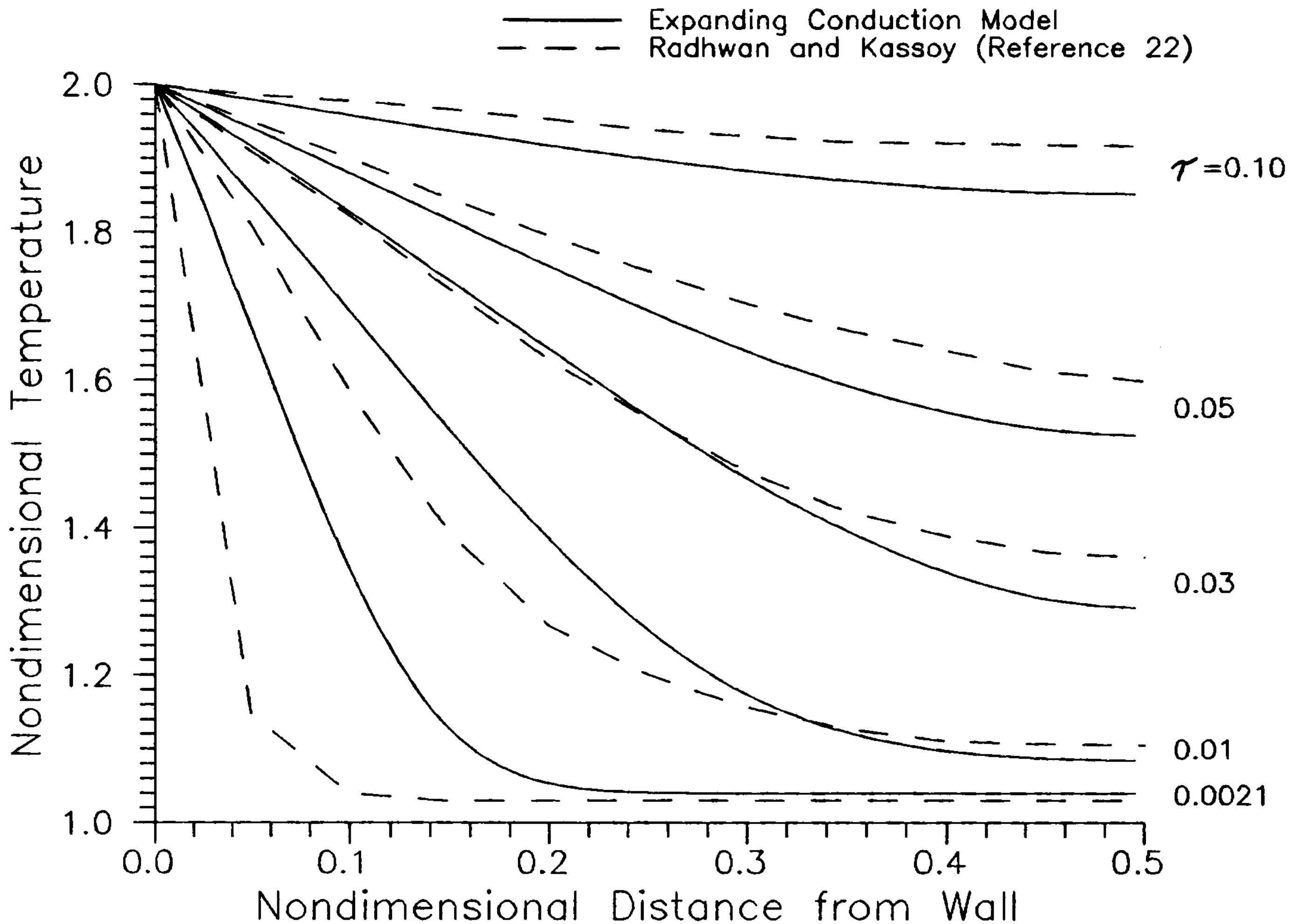


Figure 6-7. **Comparison of Numerical Temperature Profiles with Analytical Results from Reference 22**

conduction model. The temperatures obtained by Radhwan and Kassoy are lower than those of the expanding conduction model at short times, but become higher at longer times until the constant temperature steady state condition is reached.

For the purpose of qualitative comparison with the results of Radhwan and Kassoy, the same temperature profiles for the expanding conduction model with fluid boundary and initial conditions as described above are compared with temperatures for an equivalent rigid material in Figure 6-8. It is observed that the temperature of the expanding gas is initially higher than that of the rigid material, but at longer times the rigid material temperature is higher than the expanding gas temperature. This is because of the redistribution of mass back toward the walls to satisfy continuity as time progresses. These results can be compared to those given in Figure 5-12 for only one heated wall where the steady state density distribution is not homogeneous.

The velocity distribution at various times for the perturbation analysis of Radhwan and Kassoy is compared with that of the expanding conduction model in Figure 6-9. For the perturbation analysis, the velocity is observed to be everywhere toward the midpoint at short times before changing directions so that the flow is everywhere back toward the walls at longer times to satisfy continuity. The expanding conduction model, however, produces flow

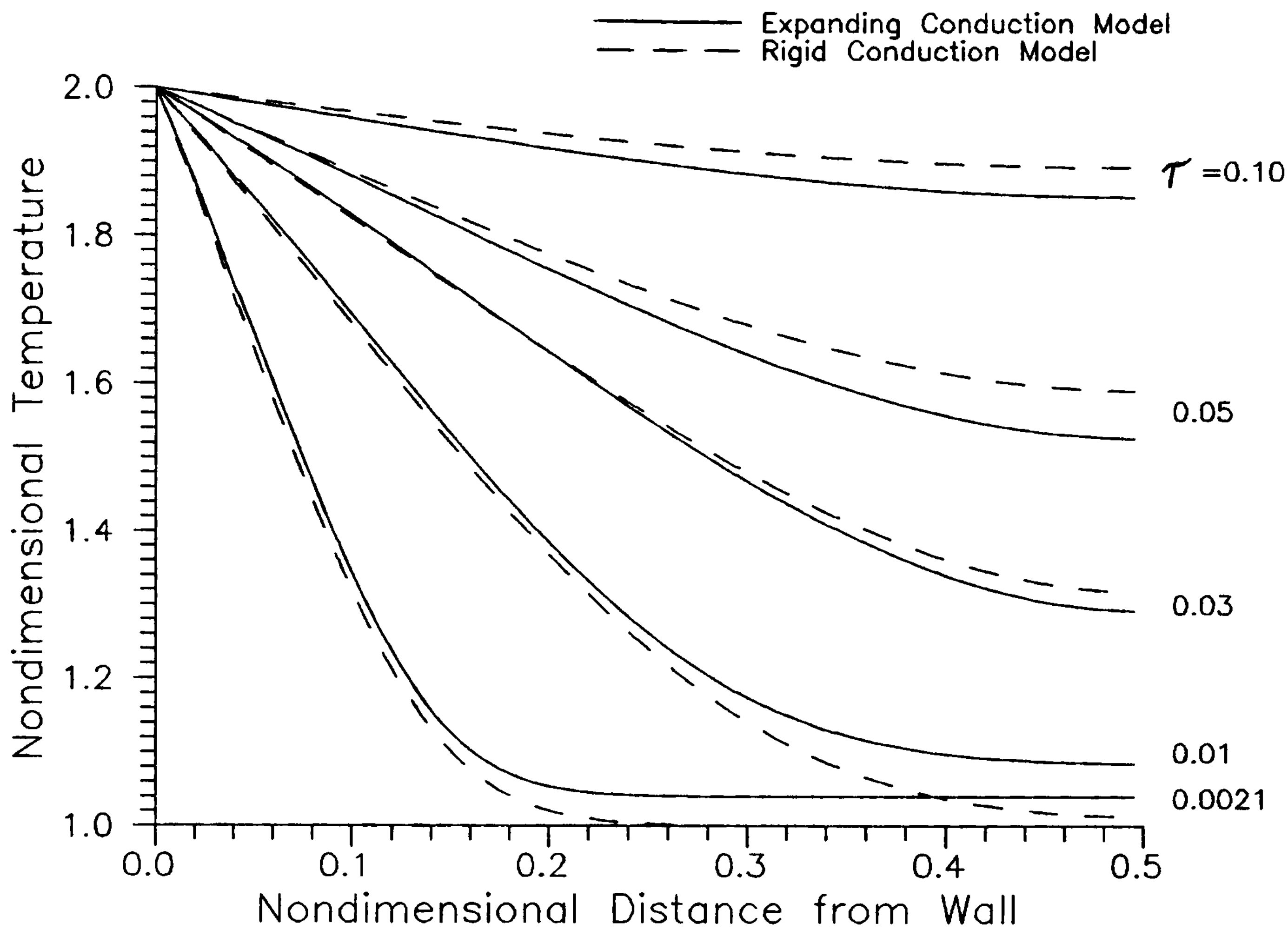


Figure 6-8. Comparison of Temperature Profiles of the Expanding Conduction Model and Rigid Material Model

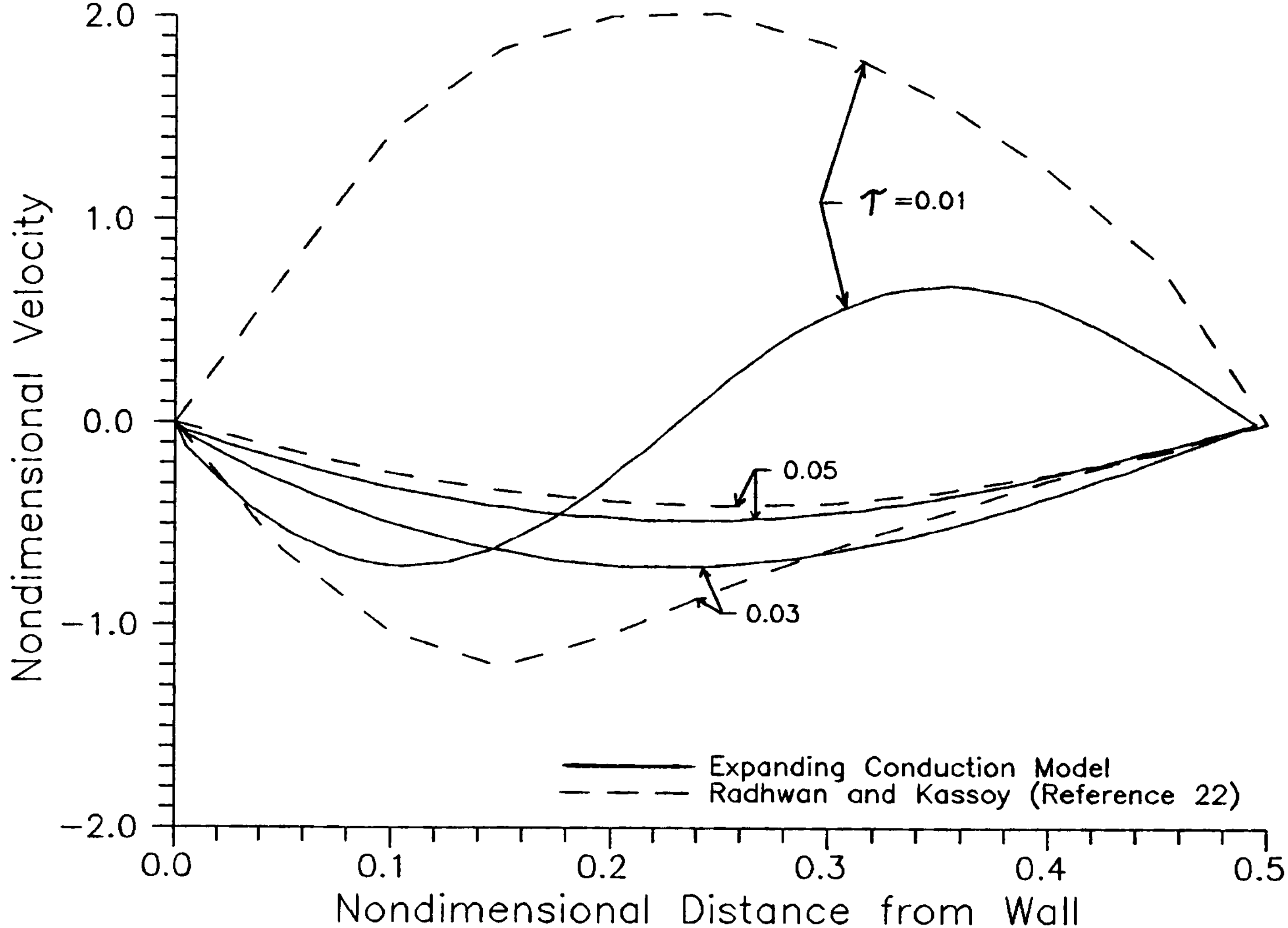


Figure 6-9. Comparison of Numerical Velocity Distribution with Velocity Distribution from Reference 22

back toward the wall at a location very near the wall at short times. This back-flow progresses toward the midpoint until the flow is everywhere toward the wall. In both cases, the steady state velocity is zero.

CHAPTER 7

CONCLUSIONS

The effects of thermally-induced convection heat transfer have been studied by measuring the unsteady temperature profile of air enclosed in a cavity and subjected to one-dimensional rapid heating at a boundary. An optical method, configured in a new and unique design, was employed for the purpose of measuring the temporal and spatial variation of the fluid index of refraction. This was accomplished by measuring the deflection of a laser beam as a function of distance from the heated boundary for a given time. The heated wall temperature was measured directly by thermocouples and the cavity pressure was measured with a pressure transducer. This data was used to determine transient air temperature profiles for various boundary heat fluxes.

A preliminary design for an in-space experiment to investigate thermally-induced convection heat transfer effects in microgravity has been presented. The design incorporates the optical temperature measurement technique and, to compensate for limited space for the experimental apparatus, includes a prism system designed to amplify the laser beam deflections. The ground experiments that have been performed in this study verify the feasibility of the in-space experiment and provide gravity experimental data for comparison with results obtained in microgravity experiments.

A finite volume numerical model which predicts thermally-induced convection heat transfer was developed. The numerical scheme accounts for the bulk velocity generated as the fluid responds to the changing temperature and pressure fields, but does not consider acoustic-type velocity perturbations. The three-dimensional effects imposed by the side walls were also accounted for in the energy equation. The experimental and numerical results were found to be in agreement for short times of about one or two seconds. For times of about two through ten seconds, the numerical model predicted temperatures higher than the experimental results. It is possible that the acoustic velocity and pressure perturbations, which are suppressed in the numerical model, do have a significant effect on the thermally-induced heat transfer phenomenon. Another possibility is that relatively cool air is mixed with the heated boundary region because of entrainment or another unexplained reason.

The numerical model described in Reference 28 was also compared with the experimental results of the present study. It was found that, except for very short times (or relatively low heated wall temperatures), the numerical model predicts much faster progress toward steady state than found by experiment. It was demonstrated that the numerical model of Reference 28 is very sensitive to the selected grid size. The model was improved by reducing the grid size resulting in a decrease of the calculated rate of temperature rise. The grid size dependence, however, was not eliminated. The model was also improved by including second order accurate methods at the wall, revising the upwind difference scheme for the convection terms, and using an implicit solution method.

Comparison was also made between the numerical model developed in this study and the perturbation analysis of Reference 22. The temperature profiles predicted in Reference 22 are lower than those of the present model at short times and higher at longer times, however, the results are found to be in qualitative agreement.

REFERENCES

REFERENCES

1. Feldman, K.T., Review of the Literature on Sondhauss Thermoacoustic Phenomena, *J. Sound Vib.*, **7**, 71-82, 1968.

2. Feldman, K.T., Review of the Literature on Rijke Thermoacoustic Phenomena, *J. Sound Vib.*, **7**, 83-89, 1968.

3. Lord Rayleigh, The Explanation of Certain Acoustical Phenomena, *Nature*, **18**, 319, 1878.

4. Clement, J.R., Gaffney, J., Thermal Oscillations in Low-Temperature Apparatus, *Adv. in Cryogenic Eng.*, **1**, 302-306, 1960.

5. Daney, D.E., Ludtke, P.R. and Jones, M.C., An Experimental Study of Thermally-Induced Flow Oscillations in Supercritical Helium, *J. of Heat Transfer*, **101**, 9-14, 1979.

6. Carome, E,F., Clark, N.A. and Moeller, C.E., Generation of Acoustic Signals in Liquids by Ruby Laser-Induced Thermal Stress Transients, *Applied Physics Letters*, **4**, 95-97, 1964.

7. Stewart, E., Stewart, P. and Watson, A., Thermo-Acoustic Oscillations in Forced Convection Heat Transfer to Supercritical Pressure Water, *Int. J. of Heat and Mass Transfer*, **16**, 257-270, 1973.

8. Welsh, M.C., Hourigan, K., Welch, L.W., Downie, R.J., Thompson, M.C. and Stokes, A.N., Acoustics and Experimental Methods: The Influence of Sound on Flow and Heat Transfer, Experimental Heat Transfer, Fluid Mechanics and Thermodynamics, Elservier Science Publishing Co,, Inc., New York, NY, 1988.

9. Grodzka, P.G. and Bannister, T.C., Heat Flow and Convection Demonstration Experiments Aboard Apollo 14, *Science*, **176**, 506-508, 1972.

10. Grodzka, P.G. and Bannister, T.C., Heat Flow and Convection Experiments Aboard Apollo 17, *Science*, **187**, 165-167, 1975.

11. Parang, M. and Salah-Eddine, A., Thermoacoustic Convection Heat Transfer Phenomenon, *AIAA Journal*, **22**, 1984.

12. Parang, M. and Salah-Eddine, A., Experiments on Thermoacoustic Convection Heat Transfer in Gravity and Low Gravity Environment, AIAA 22nd Thermophysics Conference, Honolulu, Hawaii, AIAA-87-1651, June, 1987.

13. Parang, M. and Catt, J.A., Convective Air Motion Generated by Rapid Heating in an Enclosure, AIAA 28th Aerospace Sciences Meeting, Reno, Nevada, AIAA-90-0412, Jan., 1990.

14. Giarratano, P.J., Kumakawa, A., Arp, V.D., Owen, R.B., Transient Heat Transfer Studies in Low Gravity Using Optical Measurement Techniques, *J. Thermophysics and Heat Transfer*, **4**, 53-58, Jan., 1990.

15. Trilling, L., On Thermally Induced Sound Fields, *J. Acoustical Soc. of America*, **27**, 425-431, 1955.

16. Chu, B.T., Pressure Waves Generated by Addition of Heat in a Gaseous Medium, NACA TN-3411, 1955.

17. Chu, B.T. and Ying, S.J., Thermally Driven Nonlinear Oscillations in a Pipe with Traveling Shock Waves, *The Physics of Fluids*, **6**, 1625-1637, 1963.

18. Chu, B.T., Analysis of a Self-Sustained Thermally Driven Nonlinear Vibration, *The Physics of Fluids*, **6**, 1638-1644, 1963.

19. Rott, N., Thermoacoustics, *Adv. in Applied Mechanics*, **20**, 135-175, 1980.

20. Rott, N., Thermoacoustic Heating at the Closed End of an Oscillating Gas Column, *J. of Fluid Mechanics*, **145**, 1-9, 1984.

21. Kassoy, D.R., The Response of a Confined Gas to a Thermal Disturbance, I: Slow Transients, *SIAM J. Applied Mathematics*, **36**, 624-634, 1979.

22. Radhwan, A.M. and Kassoy, D.R., The Response of a Confined Gas to a Thermal Disturbance: Rapid Boundary Heating, *J. Engineering Mathematics*, **18**, 133-156, 1984.

23. Liburdy, J.A., Thermal Acoustic Oscillation in Annular Flow of Low Temperature Helium, ASME 82-HT-12, 1982.

24. Arp, V. and Persichetti, J.M., Thermally Induced Flow in a Confined Fluid 1. Theoretical Concepts, *Int. Comm. Heat Mass Transfer*, **14**, 567-576, 1987.

25. Luikov, A.V. and Berkovsky, B.M., Thermoconvective Waves, *Int. J. Heat Mass Transfer*, **13**, 741-747, 1970.

26. Keck, H. and Schneider, W., Thermoconvective Waves in a Compressible Fluid, *Int. J. Heat Mass Transfer*, **22**, 1501-1512, 1979.

27. Larkin, B.K., Heat Flow to a Confined Fluid in Zero Gravity, *Progress in Astronautics and Aeronautics, Planetary Bodies*, **20**, 819-833, 1967.

28. Spradley, L.W., Bourgeois, S.V., Fan, C. and Grodzka, P.G., A Numerical Solution for Thermoacoustic Convection of Fluids in Low Gravity, NASA CR-2269, May, 1973.

29. Spradley, L.W. and Churchill, S.W., Pressure- and Buoyancy-Driven Thermal Convection in a Rectangular Enclosure, *J. of Fluid Mechanics*, **70**, 705-720, 1975.

30. Krane, R.J. and Parang, M., Scaling Analysis of Thermoacoustic Convection in a Zero-Gravity Environment, *J. of Spacecraft and Rockets*, **20**, 316-317, 1983.

31. Eckert, E.R.G. and Goldstein, R.J., Measurements in Heat Transfer, Second edition, Chapter 5, McGraw-Hill, 1976.

32. Robben, R., Schefer, R., Agrawal, V. and Namer, I., Catalyzed Combustion in a Flat Plate Boundary Layer I. Experimental Measurements and Comparison with Numerical Calculations, Western States Section, Combustion Institute, Paper No. 77-37, Sept. 1977.

33. Talbot, L., Cheng, R., Schefer, R. and Willis, D., Thermophoresis of Particles in a Heated Boundary Layer, *J. of Fluid Mechanics*, **101**, 737-758, 1980.

34. Parang, M. and Crocker, D.S., In-Space Experiment on Thermoacoustic Convection Heat Transfer Phenomenon - Experiment Definition, NASA CR-187070, November, 1990.

35. Patankar, S.V., Numerical Heat Transfer and Fluid Flow, McGraw-Hill-Hemisphere, 1980.

APPENDICES

APPENDIX A

APPLICATIONS FOR OPTICAL TEMPERATURE MEASUREMENT TECHNIQUE

APPENDIX A

APPLICATIONS FOR OPTICAL TEMPERATURE MEASUREMENT TECHNIQUE

An optical temperature measurement technique was developed and utilized in this investigation for measuring the temporal and spatial temperature variation in air. The effectiveness and accuracy of the technique as described in Chapters 2, 3 and 4 has been demonstrated for the specific application studied in this report. A range of applications that may benefit from the use of the technique and comparison with other gas temperature measurement methods are briefly discussed in this appendix.

The present optical technique is compared with holographic interferometry methods and thermocouples for temperature measurement in Table A-1. The present technique has several advantages over thermocouples including lack of intrusiveness, no radiation-induced error, and better time response. The primary advantages over holographic interferometry are significantly less complexity and cost. In general, the present technique seems to have advantages for applications where the temperature and/or the temperature gradient are high. The technique may also be especially useful for transient problems.

Table A-1. Comparison of Gas Temperature Measurement Methods

	The Present Optical Technique	Holographic Interferometry	Thermocouples
Intrusive	No	No	Yes
Radiation Error	No	No	Yes
Time Response	Small	Small	Large
Complexity	Moderate	Complex	Simple
Cost	Moderate	High	Low
Accuracy	Good for high temperature gradients; also for transient problems	Excellent for low or moderate temperature gradient; also for transient problems	Excellent for low temperature and steady state or slow transient

The present study successfully measured temperature variations in one dimension. It may also be possible to measure transient temperature in two dimensions by using several laser sheets arranged in a grid pattern. Two-dimensional measurements could be made for steady state problems with an ordinary circular laser beam. Since the technique measures the gas temperature gradient averaged along one direction, three-dimensional temperature measurements cannot be made.

The key to producing accurate temperature measurements with the present technique is to obtain accurate laser deflection measurements. This requires deflections that are large enough that they can be accurately determined. The laser deflection is given by Equation (2-12) which is repeated here

$$y_s = -\frac{(n_o - 1)}{\rho_o R}\frac{P}{T^2} L z_s \frac{\partial T}{\partial y} \qquad \text{(A-1)}$$

where the variables are defined in Chapter 2. The temperature gradient required to produce a laser deflection of 1/2 inch and 1 inch as a function of the experiment geometry Lz_s is shown in Figure A-1 for typical air fluid properties. It is clearly easier to produce adequate laser deflection for high temperature gradient applications, but it may be possible to obtain sufficient deflection even for low gradient cases.

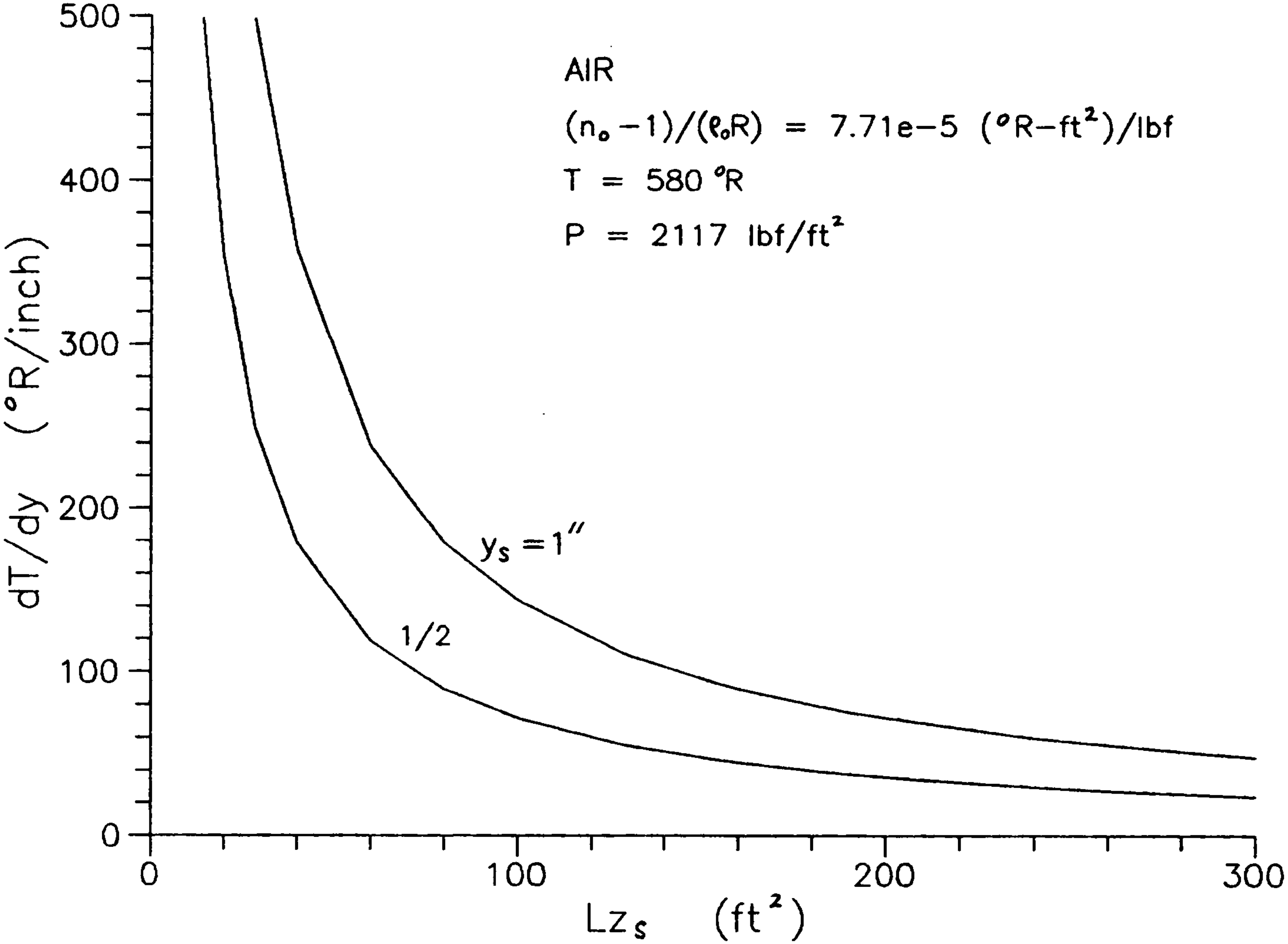


Figure A-1. Temperature Gradient Required for 1/2 and 1 Inch Laser Deflection as a Function of Experiment Geometry

APPENDIX B

DETAILED DRAWINGS OF HEATED CAVITY

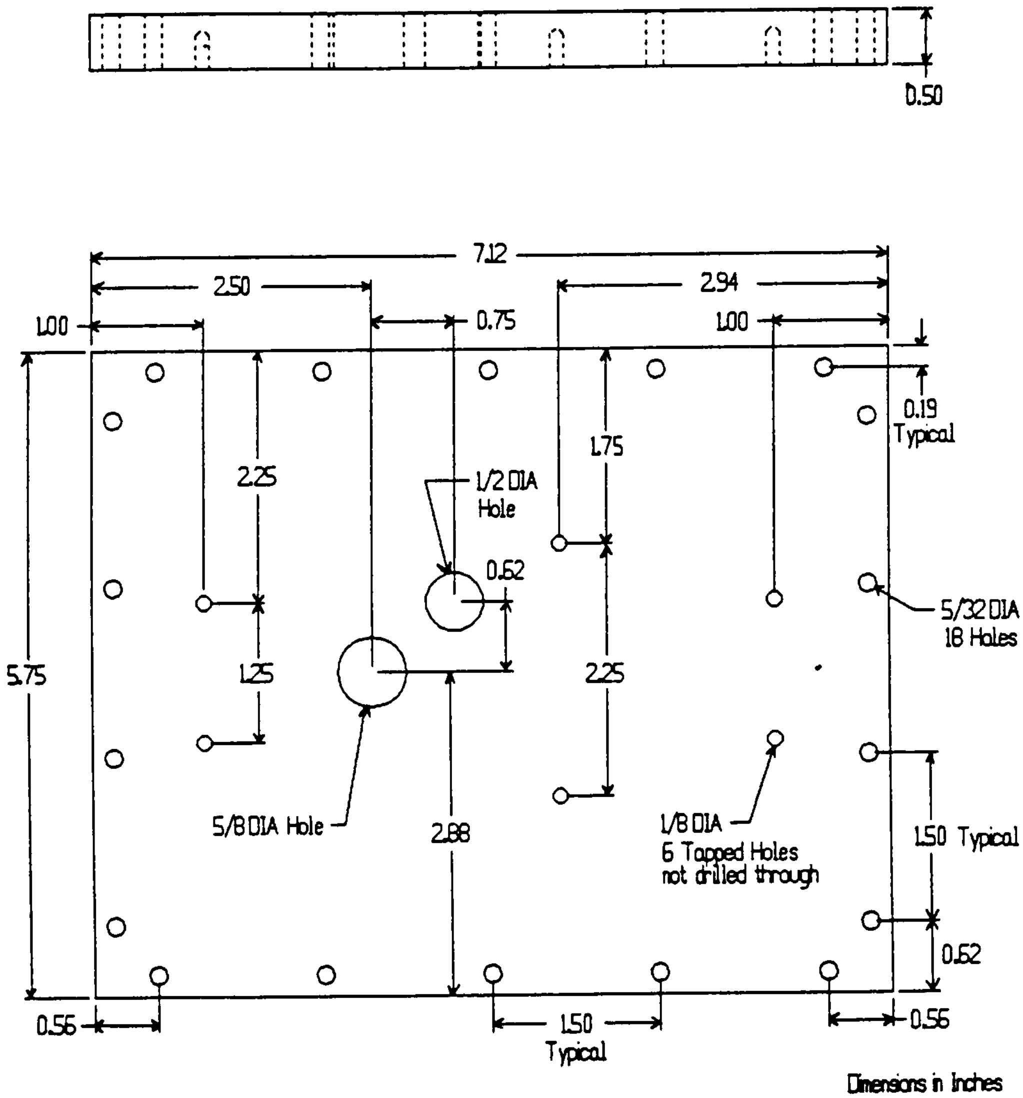


Figure B-1. Heated Cavity Apparatus, Top Wall

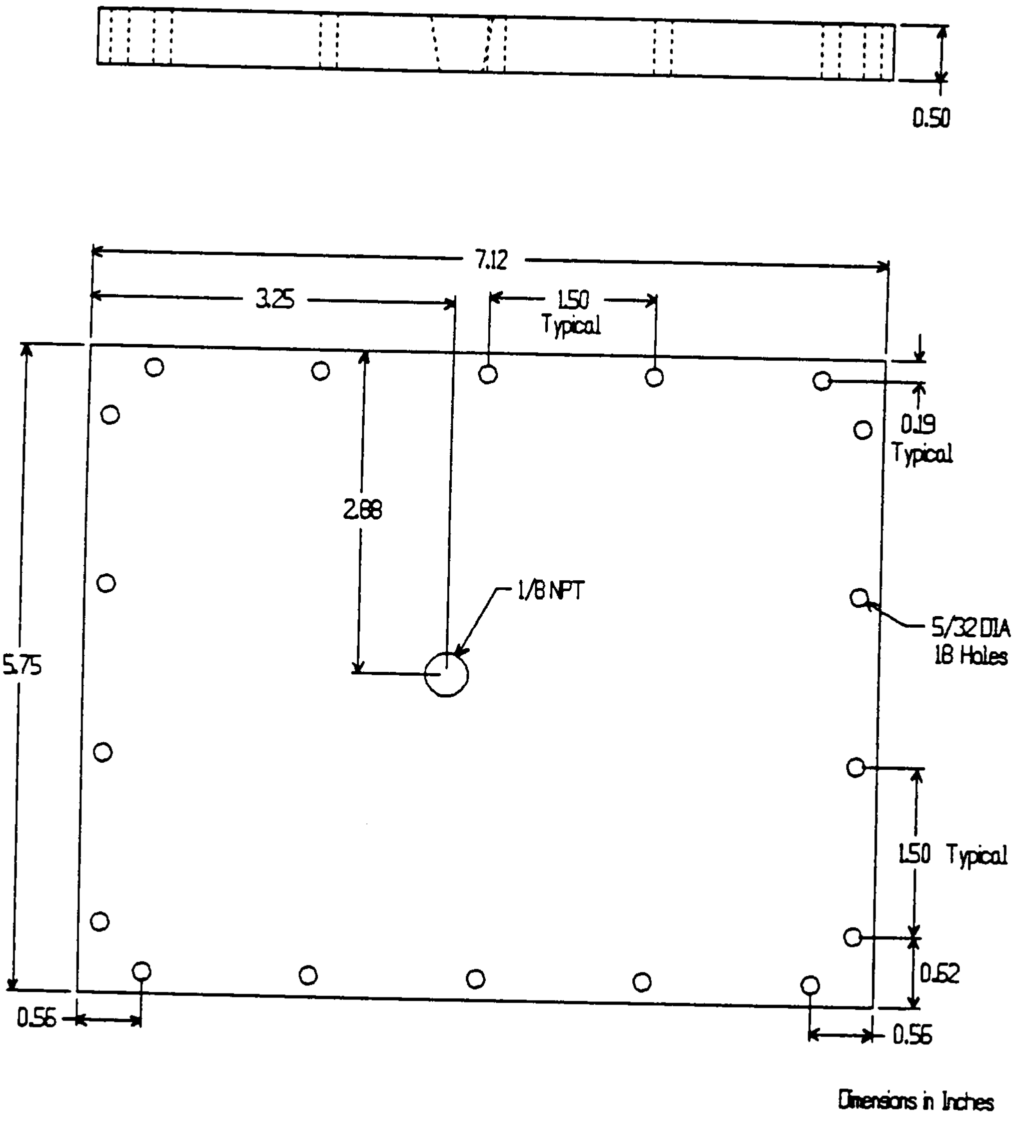


Figure B-2. Heated Cavity Apparatus, Bottom Wall

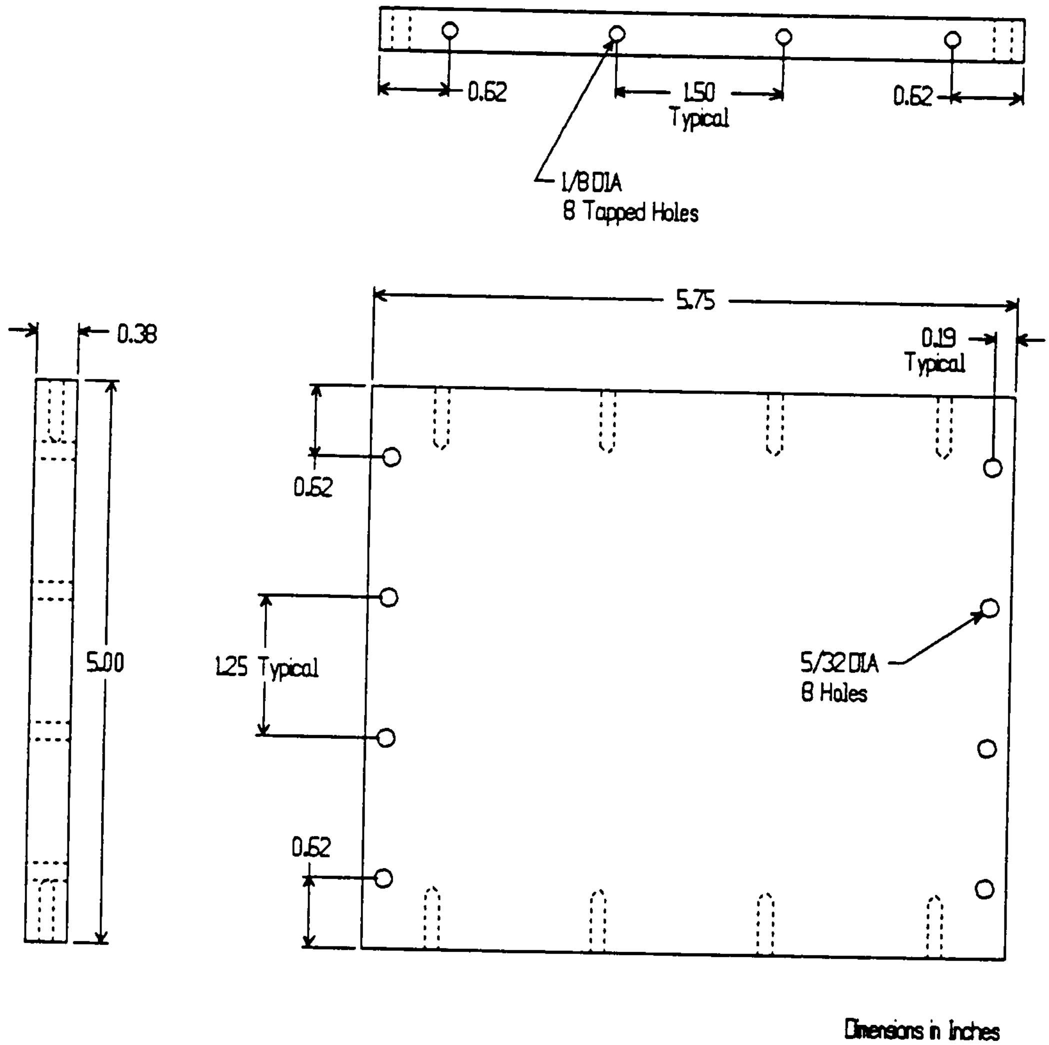


Figure B-3. Heated Cavity Apparatus, Side Wall

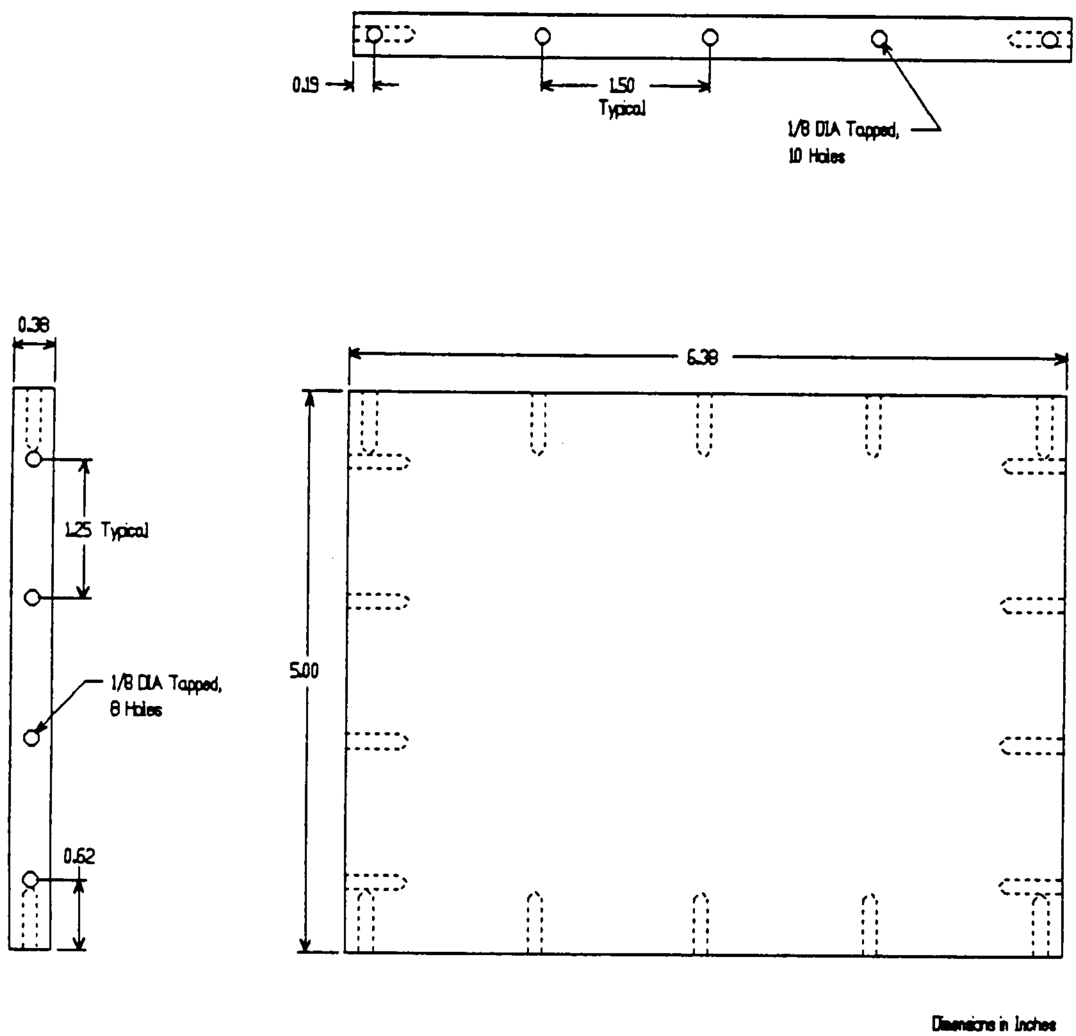


Figure B-4. Heated Cavity Apparatus, Plexiglass Wall

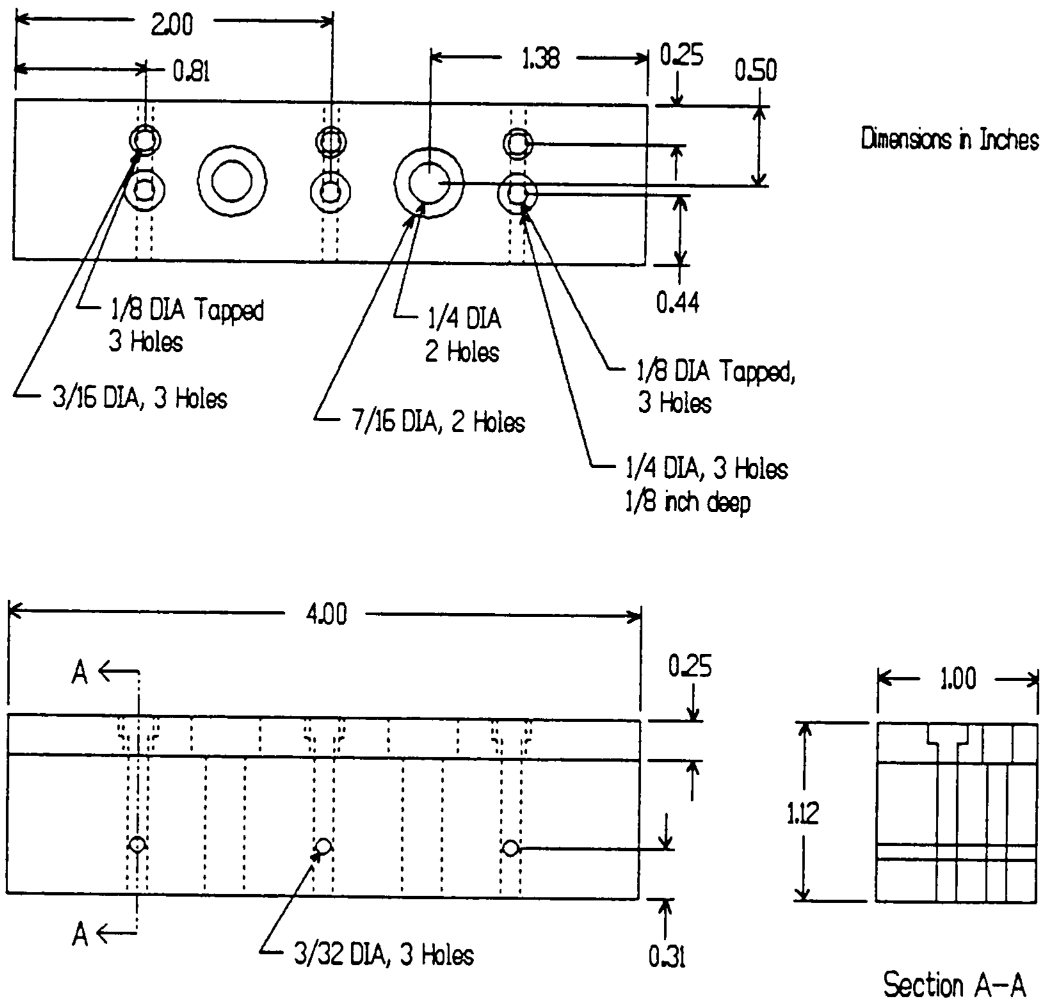


Figure B-5. Heated Cavity Apparatus, Left Busbar

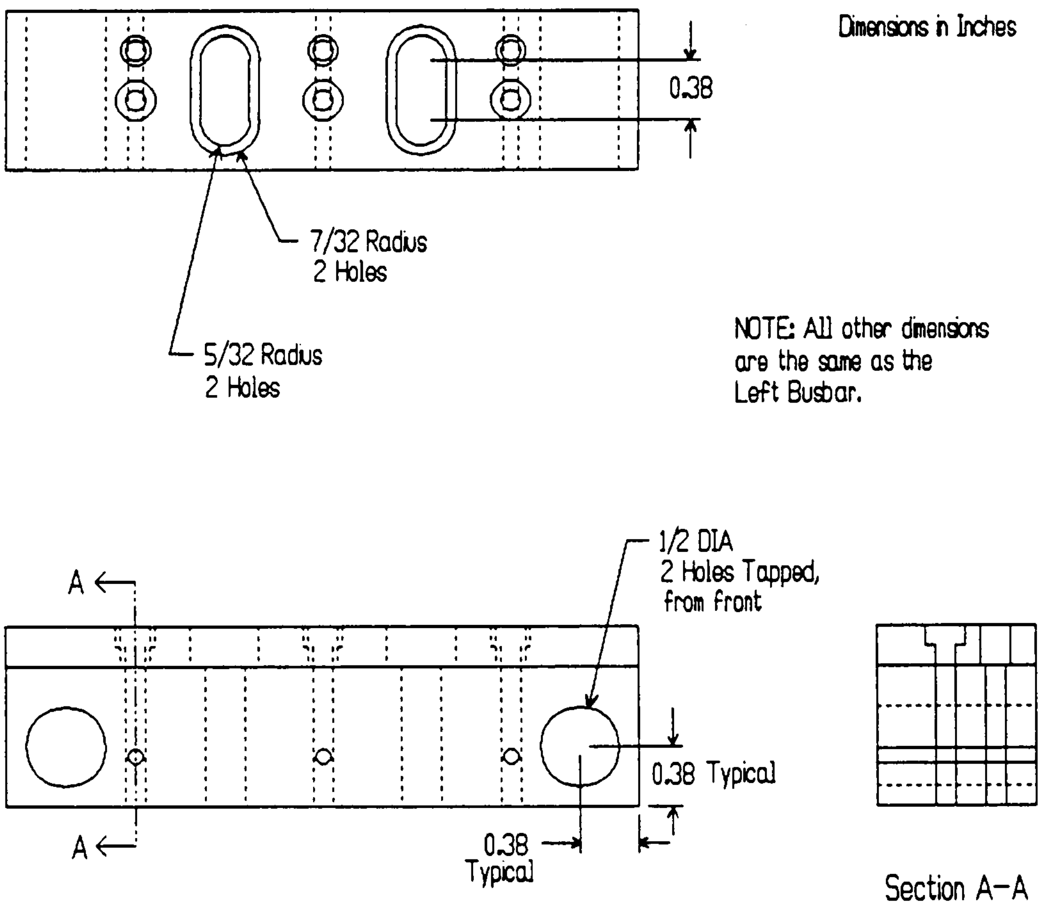


Figure B-6. Heated Cavity Apparatus, Right Busbar

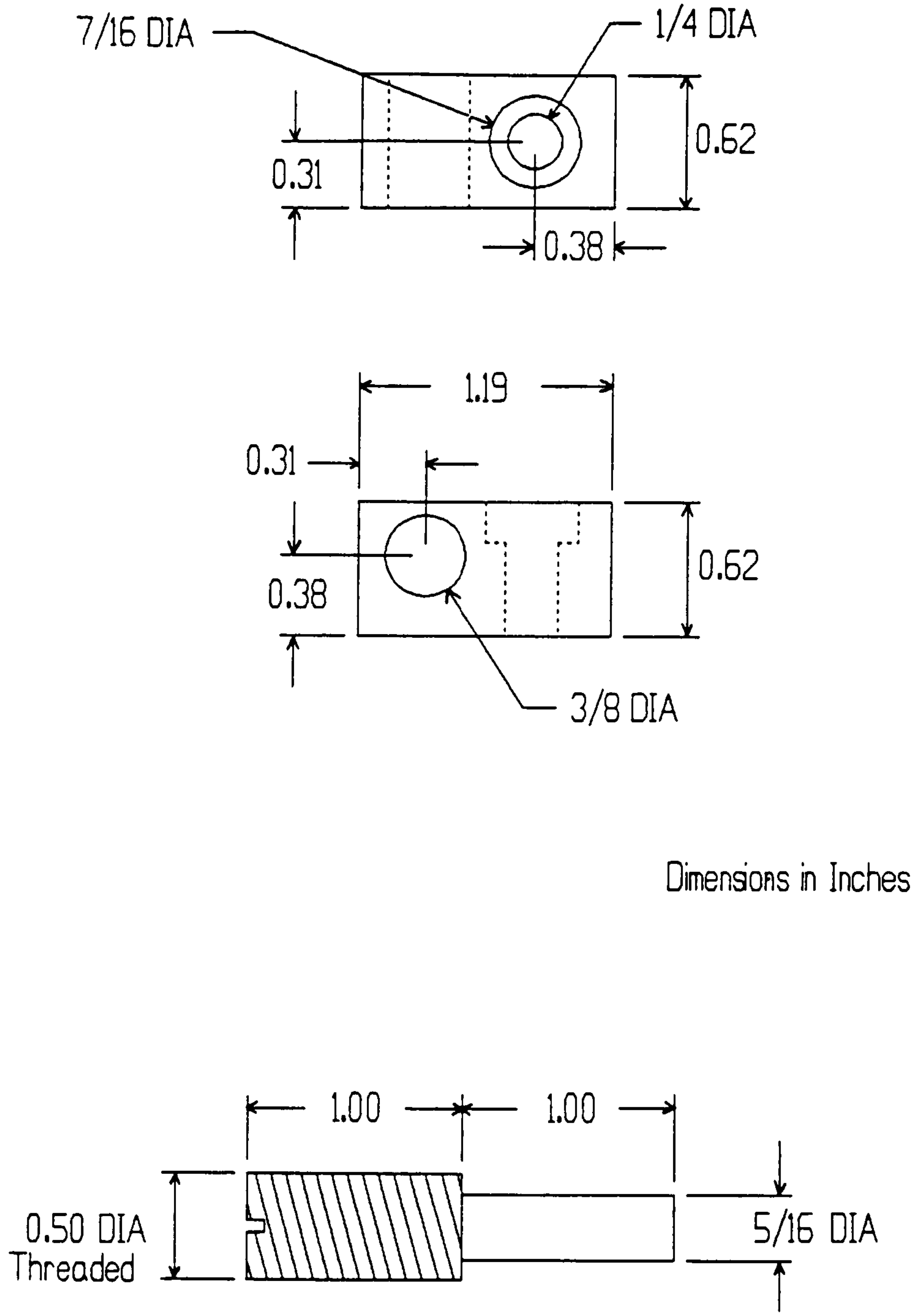


Figure B-7. Heated Cavity Apparatus, Spring Mechanism

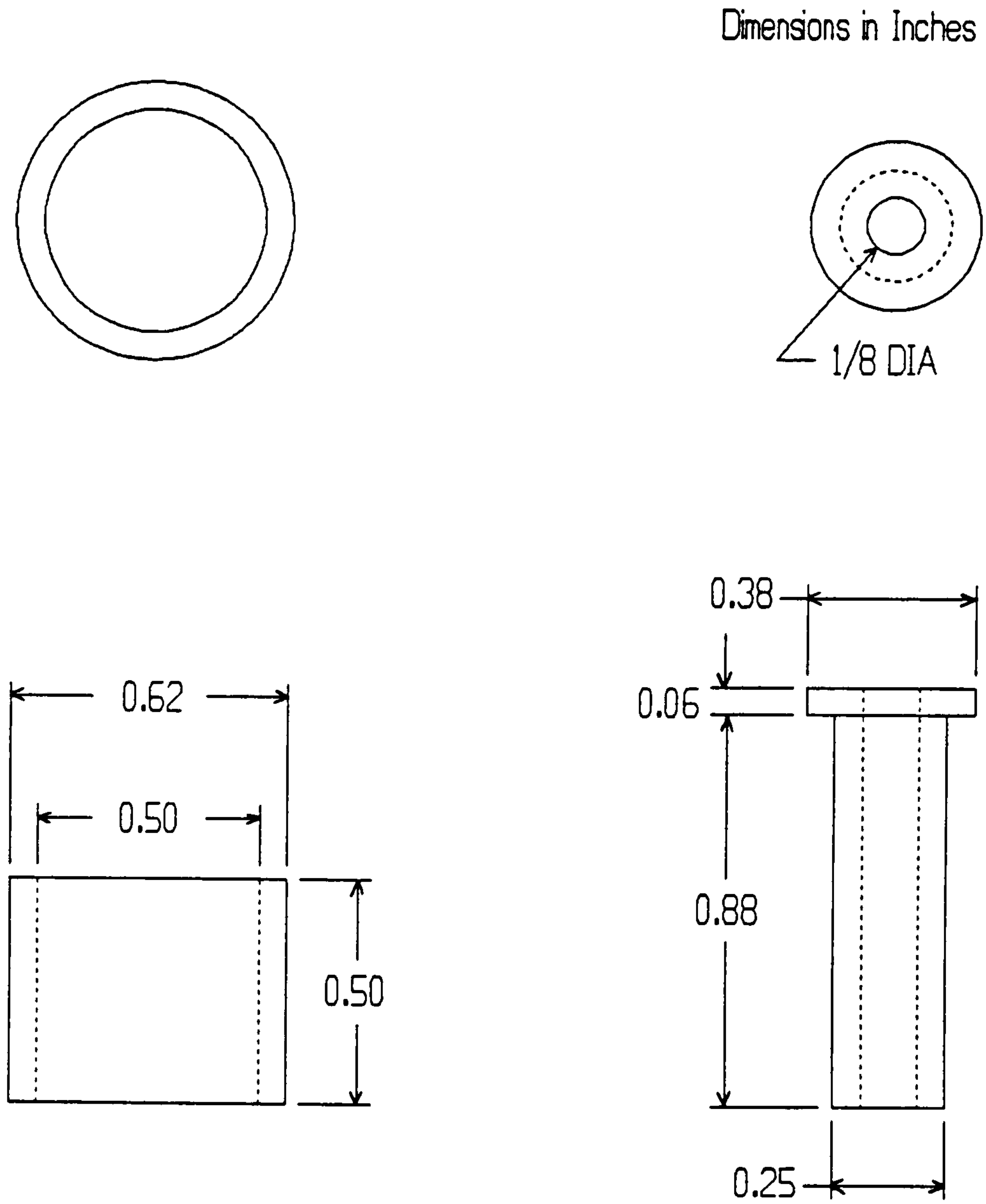


Figure B-8. Heated Cavity Apparatus, Isolators

APPENDIX C

RELATIVE MAGNITUDE OF PRESSURE WAVE PERTURBATIONS

APPENDIX C

RELATIVE MAGNITUDE OF PRESSURE WAVE PERTURBATIONS

The temperature gradient in the air is related to the laser deflection by Equation (2-13). Successfully determining the air temperature gradient (and thus the air temperature profile by integration from the known wall temperature) requires knowledge of the local pressure. The average pressure can be measured, but local acoustic pressure disturbances are unknown. The following argument, however, shows that the acoustic pressure disturbances cannot be large enough to cause significant error in the temperature gradient calculation.

An upper bound for a local pressure disturbance at the heated wall can be determined by considering a one-dimensional volume of heated air near the wall represented by Δx. The energy transferred to this volume in a given time Δt is given by

$$E = \dot{q}\,\Delta t = \rho_o c_v (\Delta x)(\Delta T) \qquad \text{(C-1)}$$

where E is energy, $\dot{q}$ is energy transfer per time per area, ρ_0 is the air density before the disturbance, c_v is the constant volume heat capacity, and ΔT is air temperature change. The pressure rise ΔP will be a maximum if the density is assumed constant. Since the pressure disturbance moves away from the wall at the local speed of sound a, the time Δt is limited to

$$\Delta t = \frac{\Delta x}{a} \tag{C-2}$$

where,

$$a = \sqrt{\gamma R T} \tag{C-3}$$

also,

$$\Delta T = \frac{\Delta P}{\rho_o R} \tag{C-4}$$

Combining the above equations gives

$$\Delta P = \frac{\dot{q}}{c_v}\sqrt{\frac{R}{\gamma T}} \tag{C-4}$$

Note that ΔP is independent of Δx, so the pressure disturbance is described by Equation (C-5) even very near the heated wall.

A maximum heat transfer rate $\dot{q}$ to the air at the moment power is applied to the heated surface can be conservatively estimated to be 200 (ft-lbf)/ft^2-sec). This is determined from the rate of increase of the measured pressure at time t=0. With this heat transfer rate, Equation (C-5) results in a maximum pressure perturbation of ΔP = 0.071 lbf/ft^2 which is just 0.003% of the ambient pressure. This demonstrates that the use of the average measured pressure in Equation (2-13) does not introduce significant error.

APPENDIX D

ERROR ANALYSIS

APPENDIX D

ERROR ANALYSIS

An analysis was performed to determine the maximum error in the measured temperature profile. Several sources of error have been identified and are discussed below:

a) Laser Sheet Deflection -- The least count measurement of the video image is one pixel, however, imperfect sharpness of the laser sheet at the screen can increase the uncertainty.

b) Heated Wall Flatness Correction -- If the heated wall is not perfectly flat and parallel to the laser beam direction, the end of the laser sheet nearest the heated wall will not be exactly adjacent to the heated wall. The end of the laser sheet will actually pass an average distance h away from the wall. This distance may change with time if the heated wall deforms as it is heated. This changing distance Δf can be estimated by the change in the location of the end of the laser sheet on the screen. The temperature profile was corrected by assuming that $\partial T/\partial y$ at the wall is equal to $\partial T/\partial y$ at $y=h$.

c) Wall Temperature -- Although thermocouple measurements should be accurate within ±2°R, experimental measurements indicate larger uncertainty even near the center of the heated wall where the end effects are minimal. Heated wall temperature variations were probably caused by a varying level of contact pressure with insulation behind the heated wall.

d) Test Section Length, L -- A significant source of error in L (see Figure 2-8) was the gaps between the heated wall and the glass insulation on the two transparent side walls. These gaps were necessary to avoid heat damage to the glass. The error in L will not be as large as the gaps, however, since the gap space is not directly heated.

e) Other Dimensions -- There was uncertainty in the measurement of the following quantities: the laser path length z_s to the screen, the angle θ and the width w of the laser sheet in the test section (see Figure 2-6).

f) Three-Dimensional Effects -- The temperature of both the heated wall and the air will be affected by the side walls. This effect was measured and accounted for. Effects on the air temperature were estimated by numerical approximation since their experimental measurement was difficult.

The uncertainty for the above sources of error has been estimated and the effects on the temperature profile determined. The error in the temperature was found by adding the estimated uncertainty to the appropriate variable and recording the change in the resulting temperature. The results for the temperature profile for medium high power at 5.0 seconds are given in Table D-1 and Figure D-1. Table D-1 gives the sign and magnitude of the maximum temperature error as a percent of the heated wall temperature rise for each source of error. Note that the wall flatness correction is always positive, but it results in increased temperature near the wall and decreased temperature at the edge of the temperature profile. The worst-case high and low percent error is also given. Figure D-1 shows a plot of the temperature profile with a worst case high and low band. The error band is biased toward higher temperatures because of the effects of the wall flatness correction and the uncertainty in the test section length.

Table D-1. Temperature Error Analysis for Medium High Power, Time = 5.0 seconds

Uncertain Quantity	Amount of Uncertainty	Maximum Temperature Error as a percent of Wall Temperature Increase
Laser Sheet Deflection	+1 pixel -1	-2.8 +3.0
Heated Wall Flatness	0.05 + Δf inches	+4.8 -0.5
Wall Temperature	+40°R -40	+2.8 -2.8
Test Section Length, L	+0.167 feet -0.0	+3.6 0.0
Laser Path Length to Screen, z_s	+0.1667 feet -0.1667	+0.2 -0.2
Laser Sheet Angle, θ	+1.0° -1.0°	+1.0 -1.0
Width of Laser Sheet, w	+0.025 inches -0.025	-1.0 +1.0
Worst Case High	--	+8.5
Worst Case Low	--	-2.8

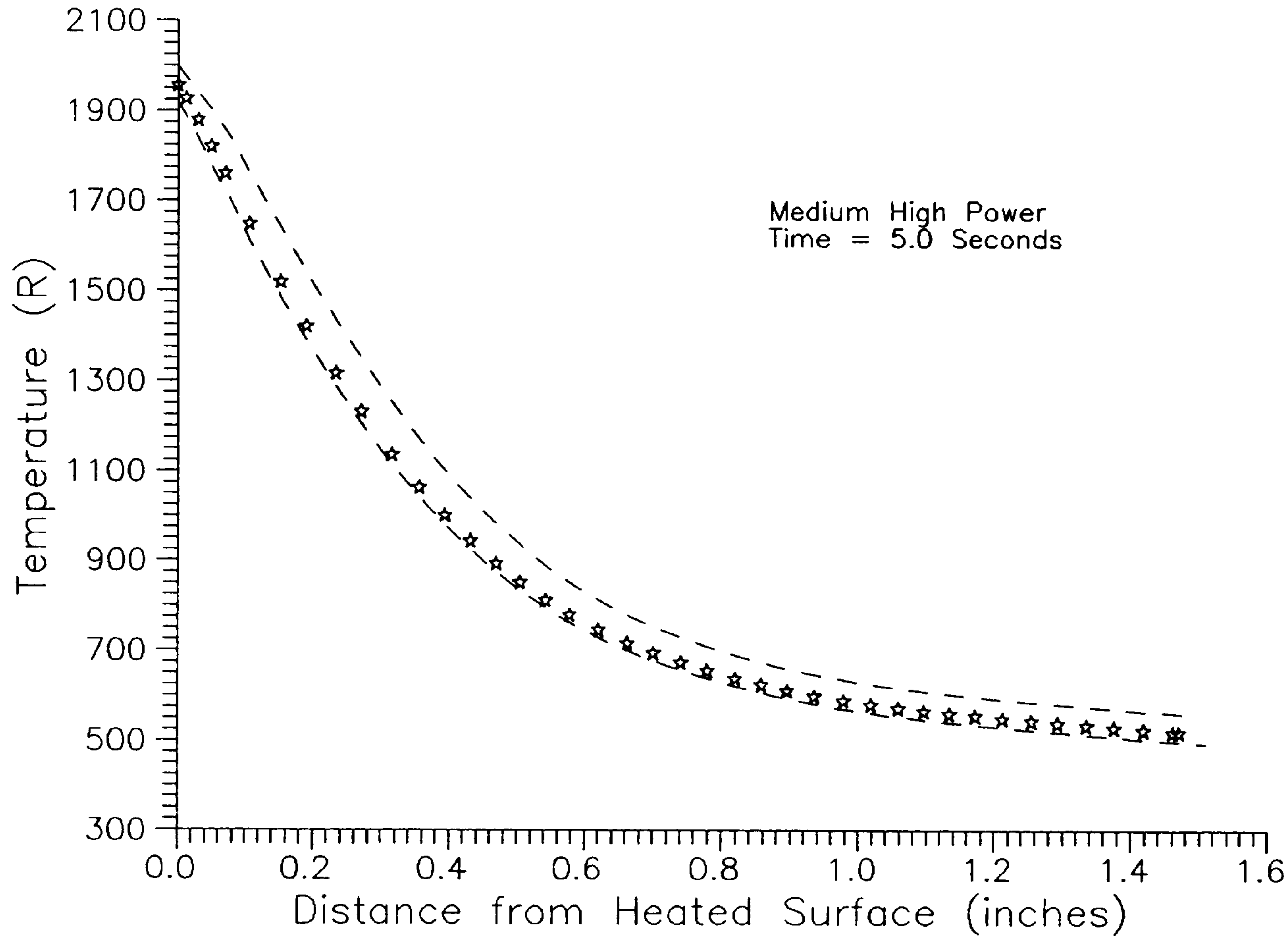


Figure D-1. Temperature as a Function of Distance from the Heated Wall with Worst-Case Error Band (Typical Case)

APPENDIX E

ENERGY BALANCE METHOD FOR ADJUSTING TEMPERATURE PROFILE

APPENDIX E

ENERGY BALANCE METHOD FOR ADJUSTING TEMPERATURE PROFILE

The temperature profile can be adjusted by an alternative method. Here T_c is determined from an energy balance between the thermal boundary layer and the core region where it is assumed that air at the edge of the thermal boundary layer flows into the core region at the average core temperature. This yields

$$T_c = \frac{\left[\rho_o + \frac{\gamma}{2}(\rho_c - \rho_o)\right]}{\left[\rho_c - \frac{\gamma}{2}(\rho_c - \rho_o)\right]} T_o \tag{E-1}$$

where ρ_c is the density of the core at the time of interest. Conservation of mass yields ρ_c as

$$\rho_c = \rho_o + (\rho_o - \rho_b)\frac{y_b}{(H + H_{ex} - y_b)} \tag{E-2}$$

The average density of the thermal boundary is

$$\rho_b = \frac{1}{R}\int_0^{y_b} \frac{P}{T_a}\,dy \qquad \text{(E-3)}$$

Equation (E-3) is solved numerically using the measured pressure and the adjusted temperature profile and the gas constant R. Equations (3-8), (E-1), (E-2) and (E-3) can be solved by iteration to give the adjusted temperature profiles and the core temperature. The adjusted temperature profiles, using this energy balance method, are shown in Figures E-1 through E-4.

The gage pressure calculated from these adjusted temperature profiles is compared with the measured gage pressure and is shown in Figures E-5 through E-8. The discrepancy between the measured and calculated pressure at times above two seconds may be due to the underestimation of the core temperature. Note that the temperature profiles in Figures E-1 through E-4 are only slightly different from those in Figures 3-18 through 3-21. This indicates that small changes in the core temperature result in large changes in pressure. The largest difference in the core temperature occurs at 10.0 seconds after the initiation of heating. The core temperature as a percent of the total temperature rise is 1.2, 2.2, 3.3, and 4.8 percent for low, medium low, medium high, and high power levels, respectively.

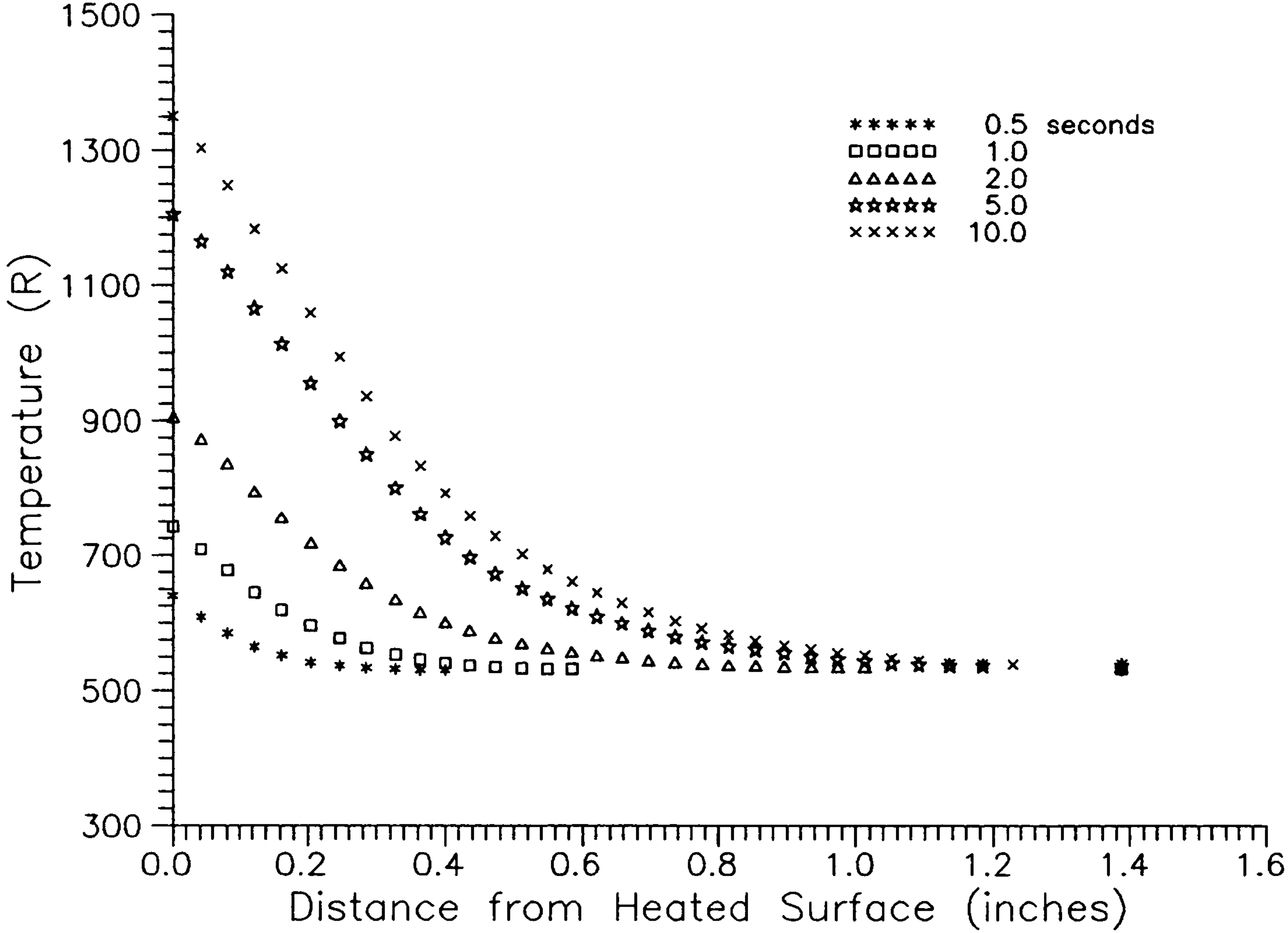


Figure E-1. **Corrected Temperature as a Function of Distance from the Heated Wall for Low Power, Energy Balance Method**

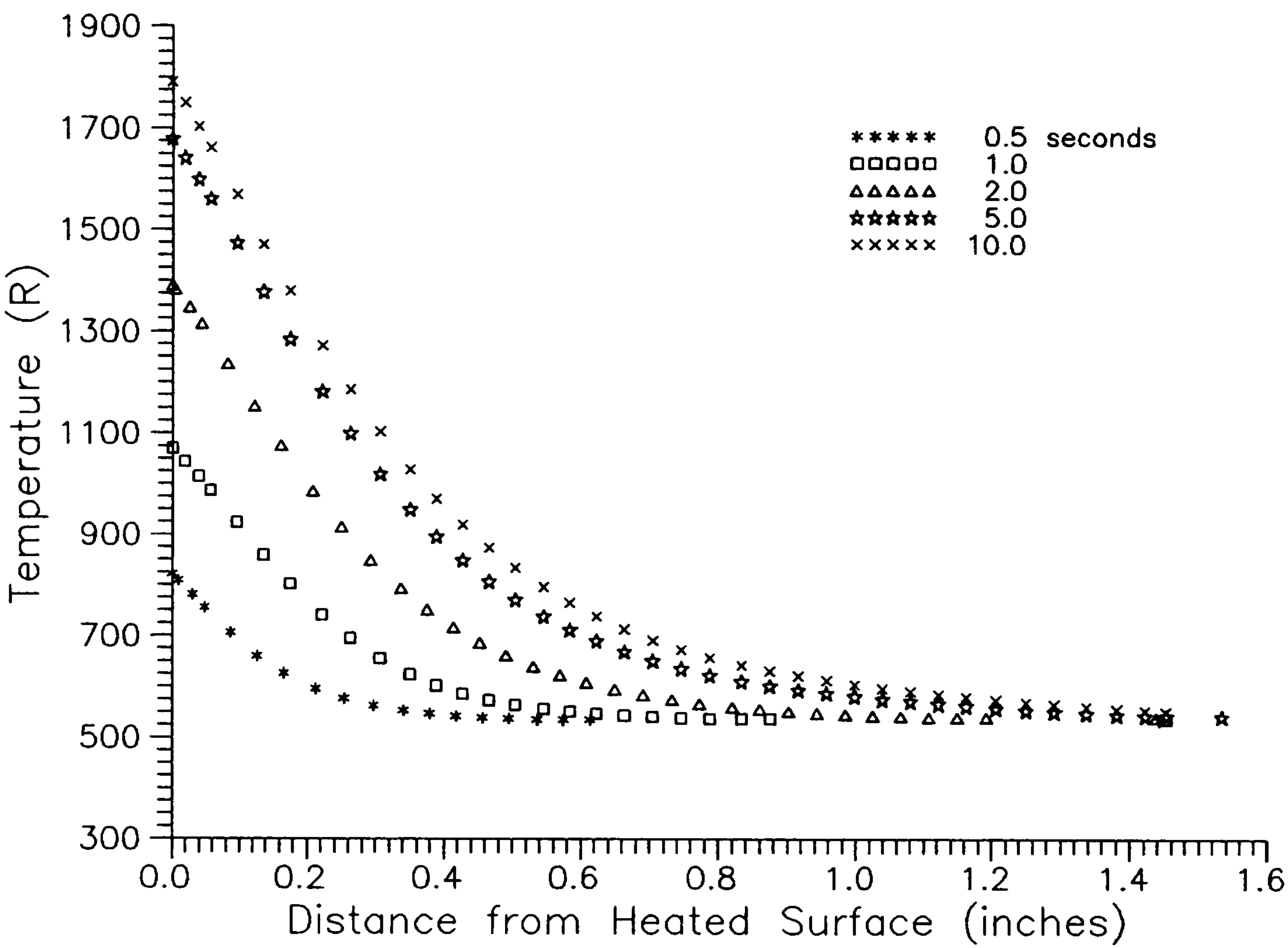


Figure E-2. Corrected Temperature as a Function of Distance from the Heated Wall for Medium Low Power, Energy Balance Method

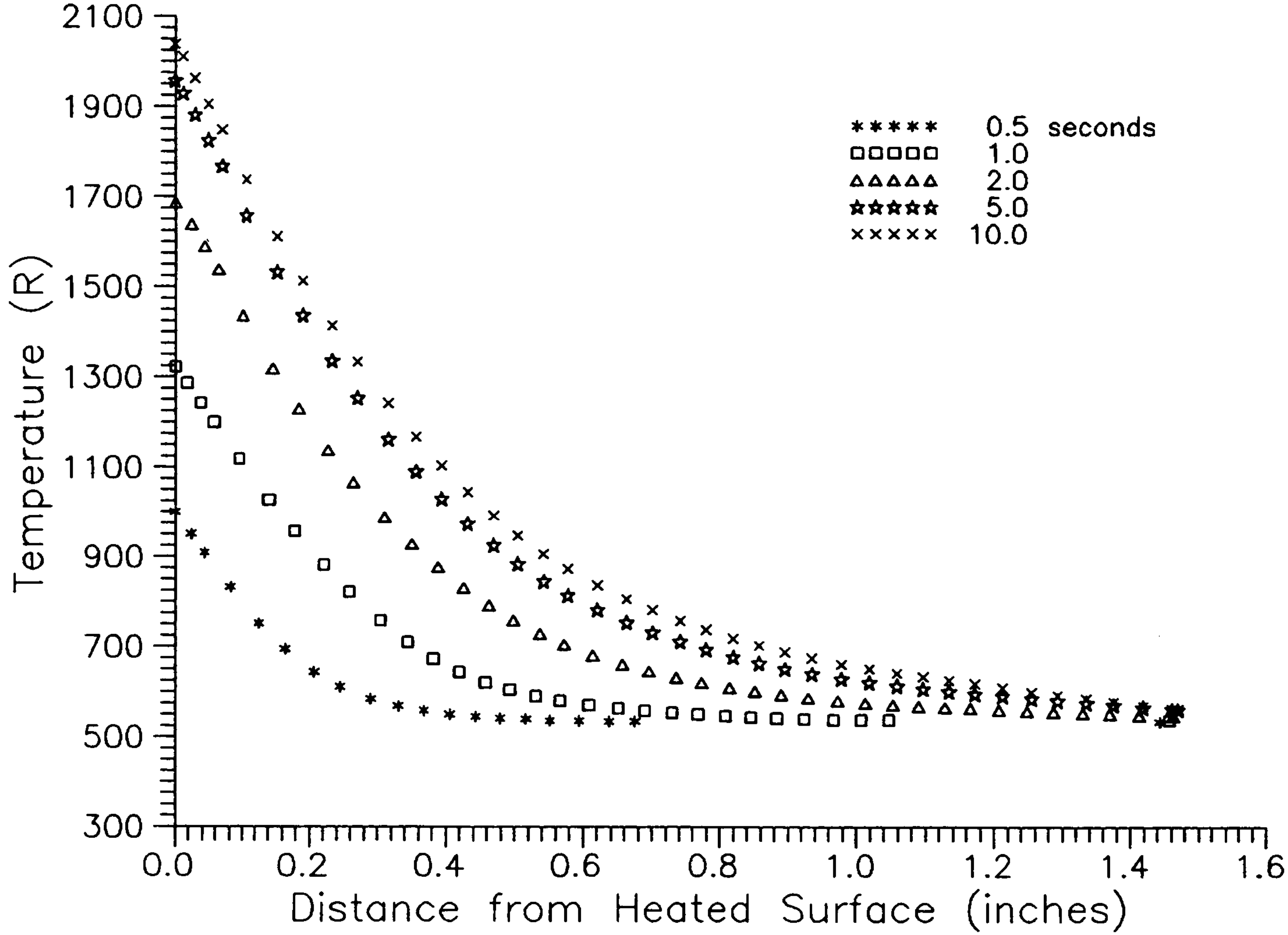


Figure E-3. **Corrected Temperature as a Function of Distance from the Heated Wall for Medium High Power, Energy Balance Method**

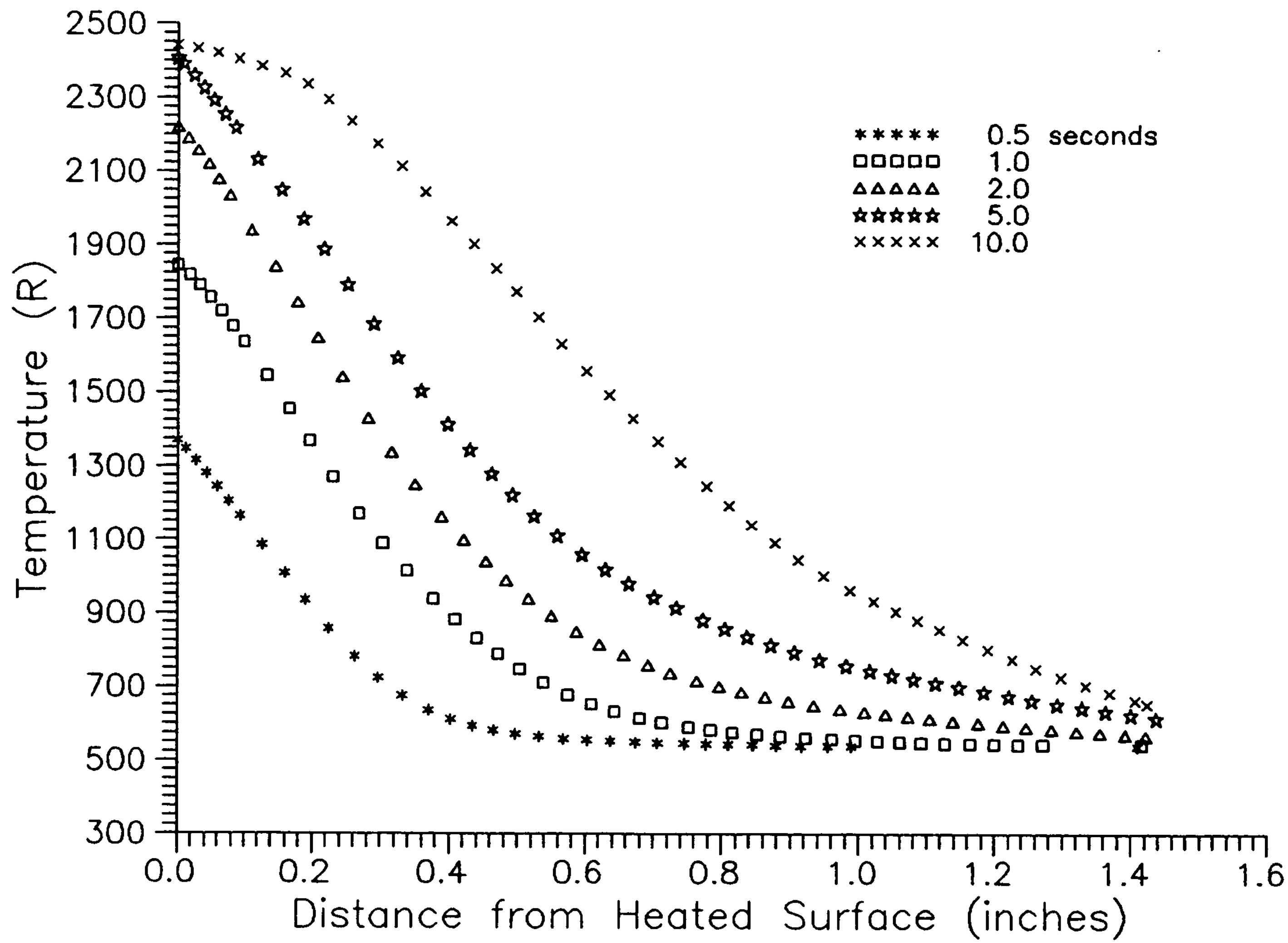


Figure E-4. **Corrected Temperature as a Function of Distance from the Heated Wall for High Power, Energy Balance Method**

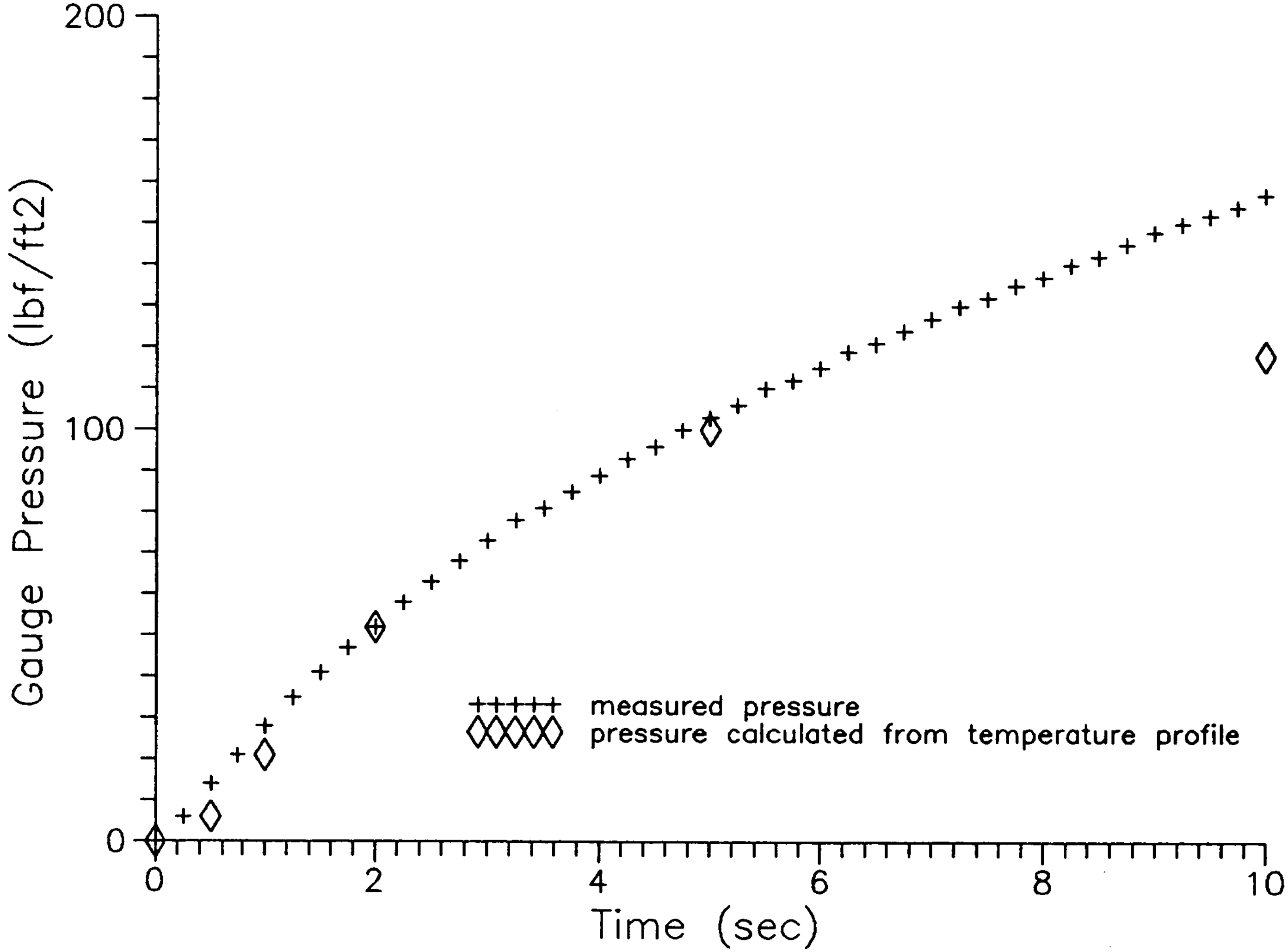


Figure E-5. Pressure as a Function of Time for Low Power

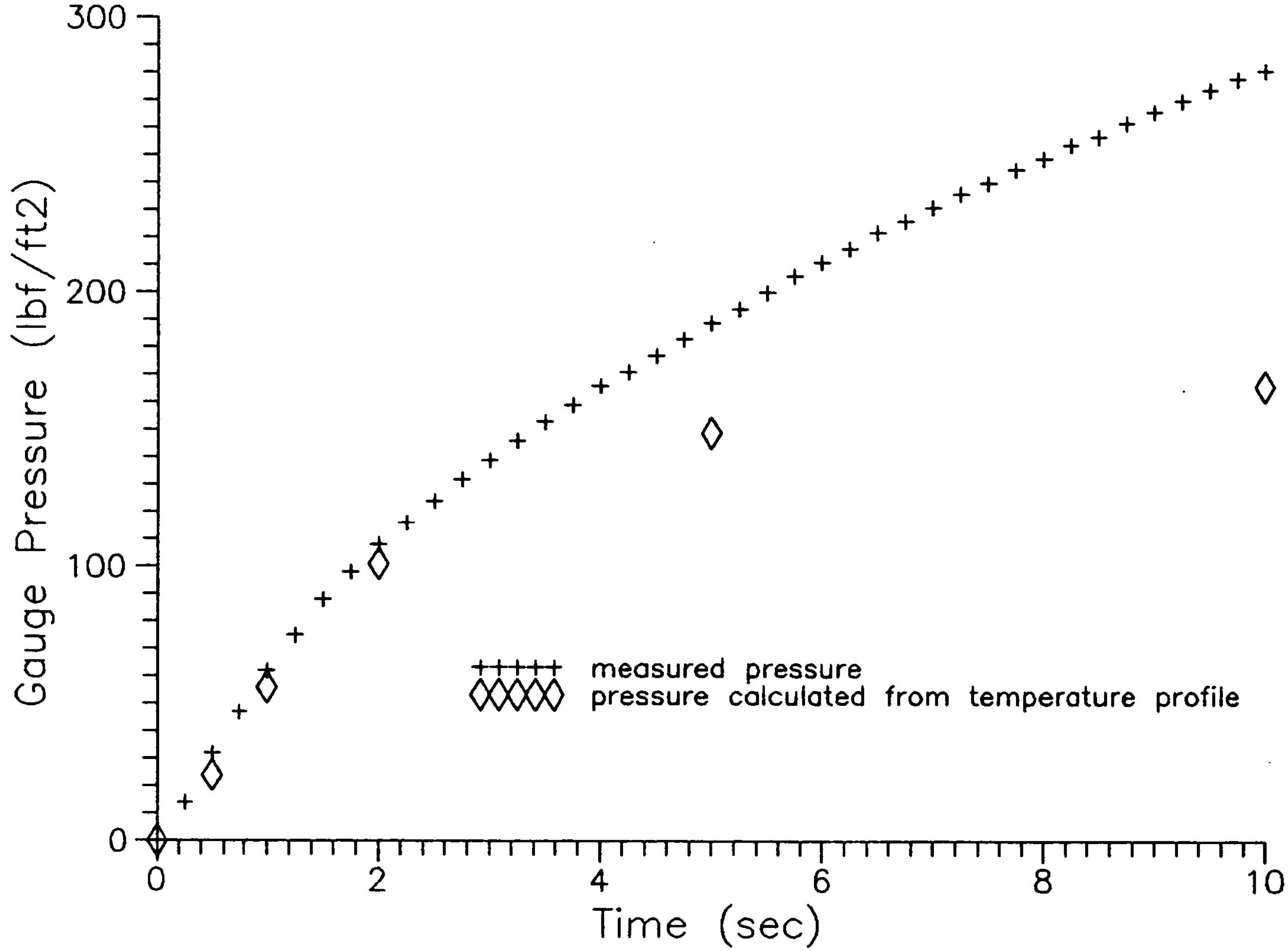


Figure E-6. Pressure as a Function of Time for Medium Low Power

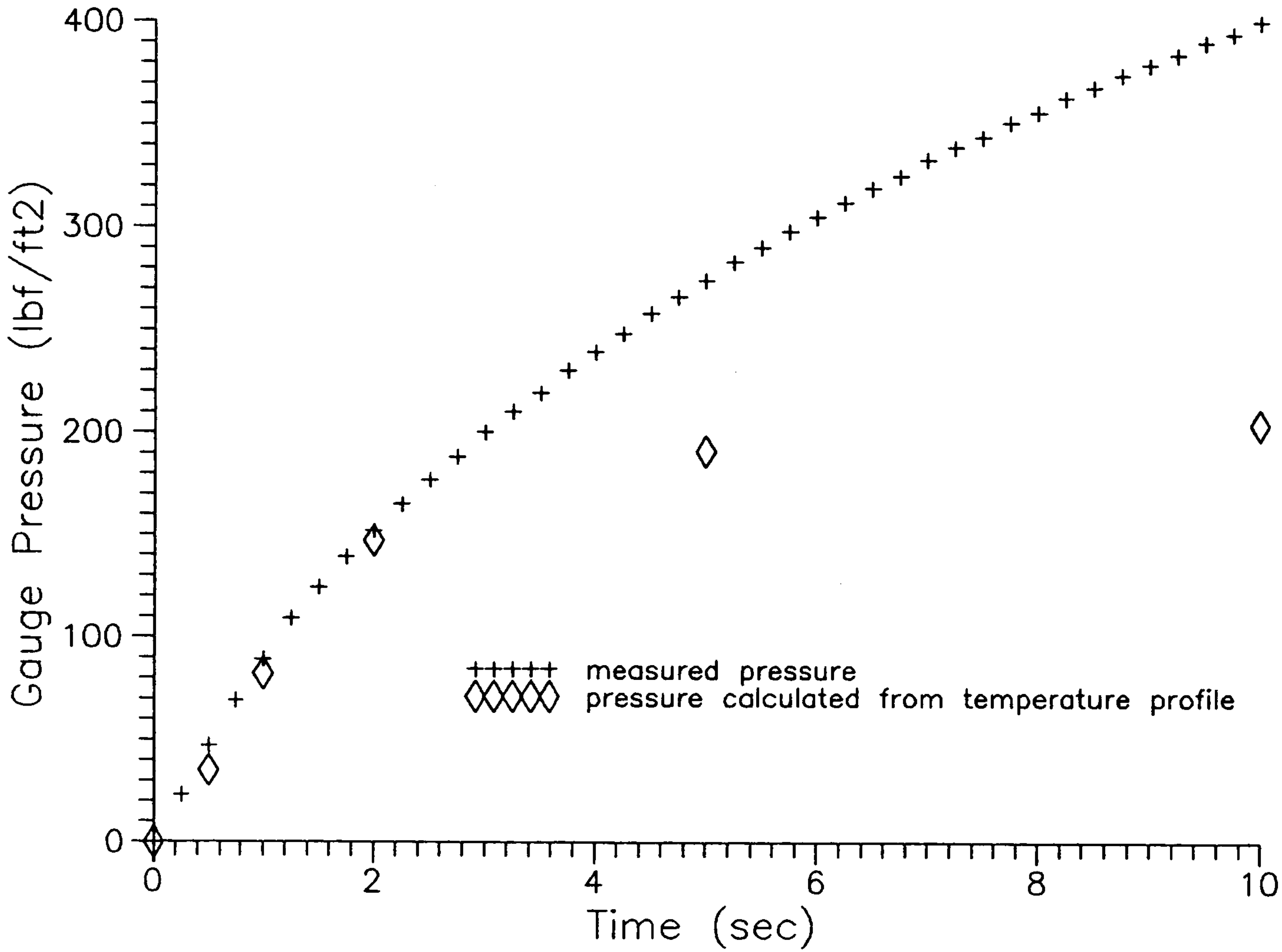


Figure E-7. Pressure as a Function of Time for Medium High Power

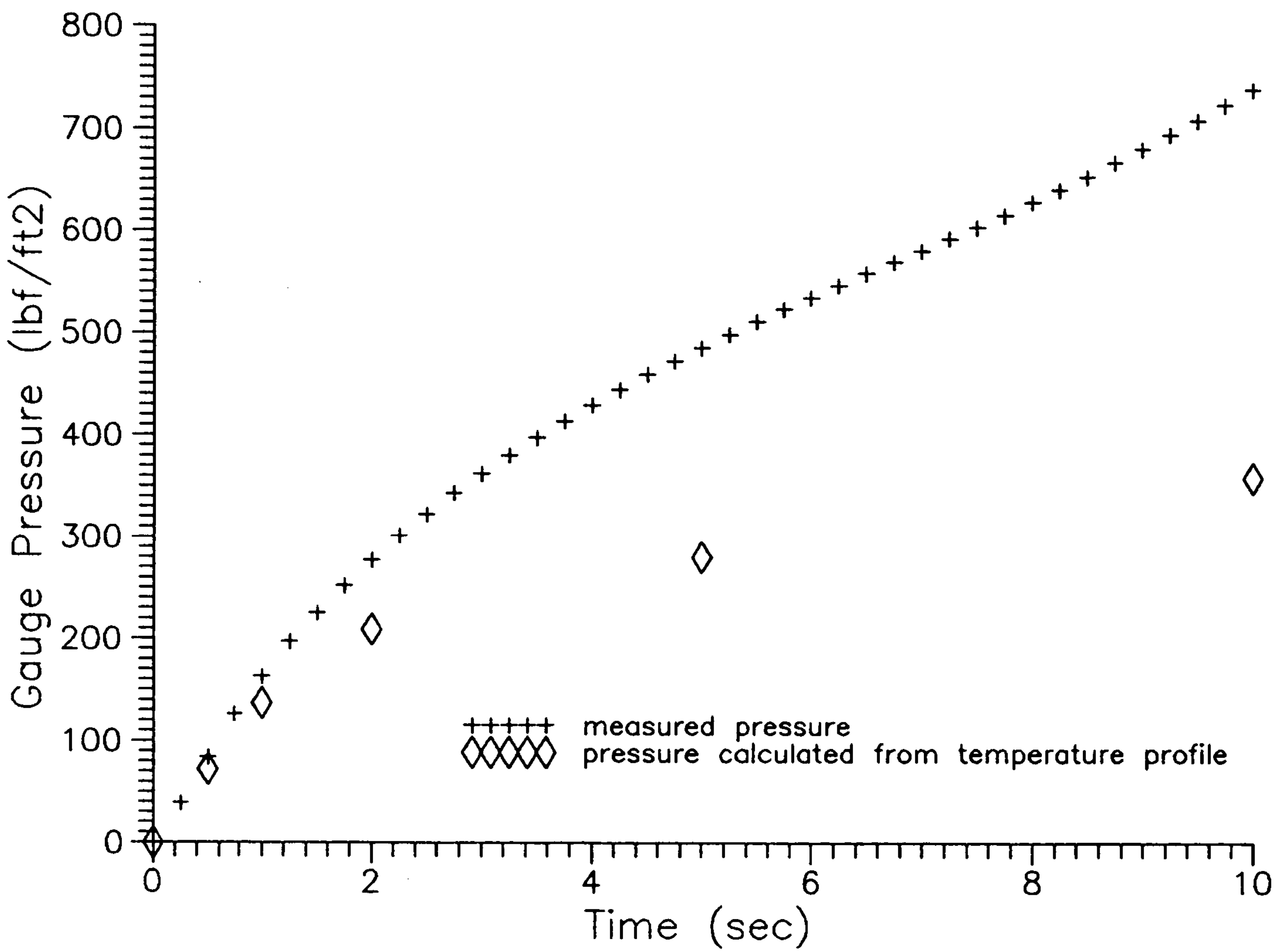


Figure E-8. Pressure as a Function of Time for High Power

APPENDIX F

DETAILED DRAWINGS OF THE SPACE EXPERIMENT DESIGN AND SPECIFICATIONS FOR OFF-THE-SHELF HARDWARE

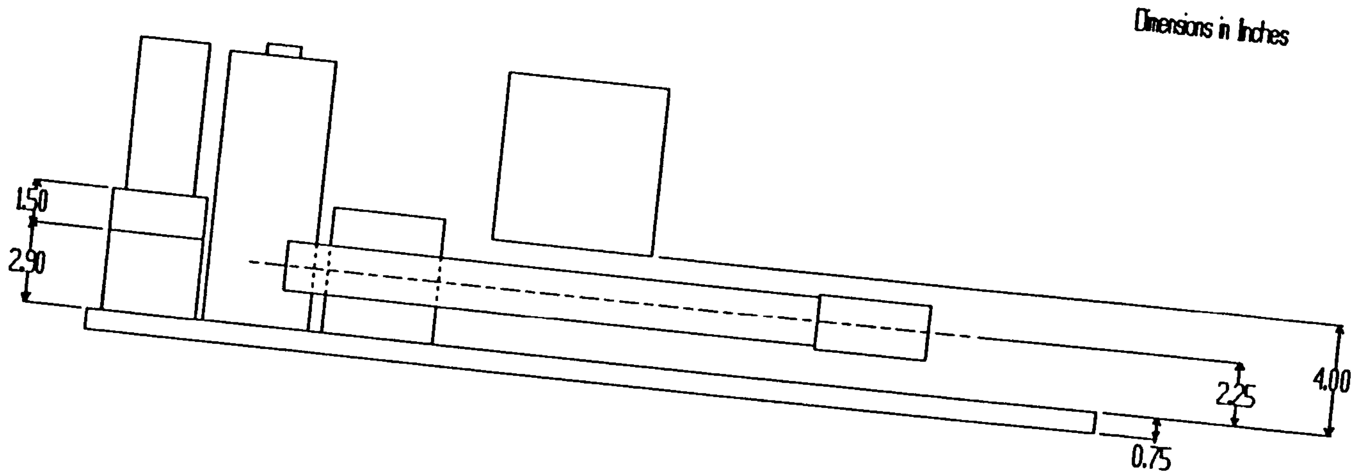


Figure F-1. Space Experiment Hardware Layout, Front View

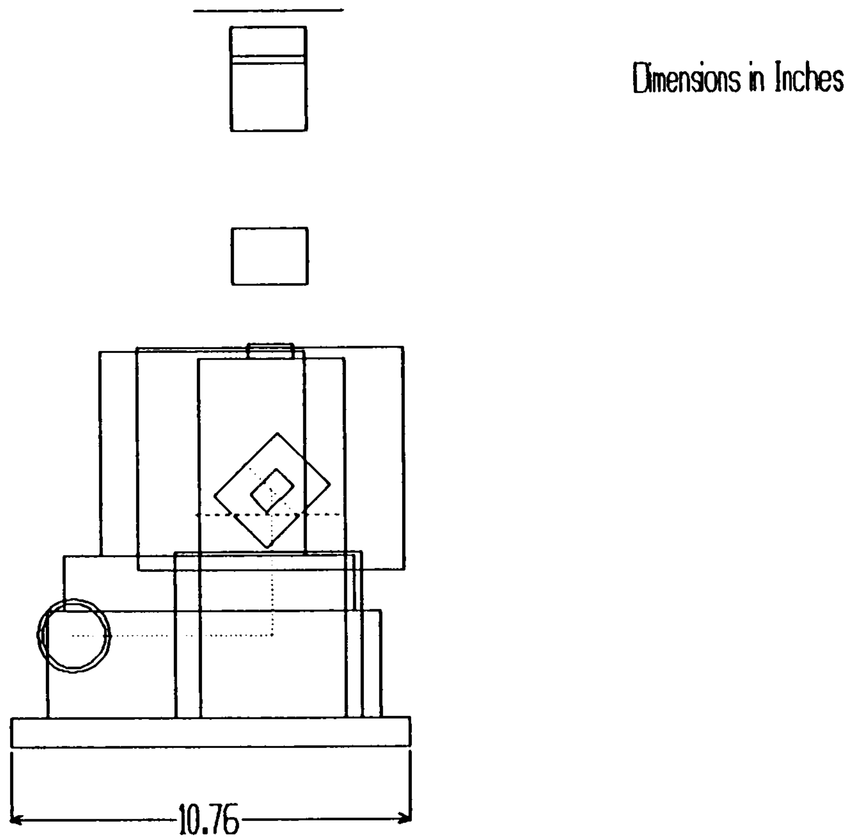


Figure F-2. Space Experiment Hardware Layout, Side View

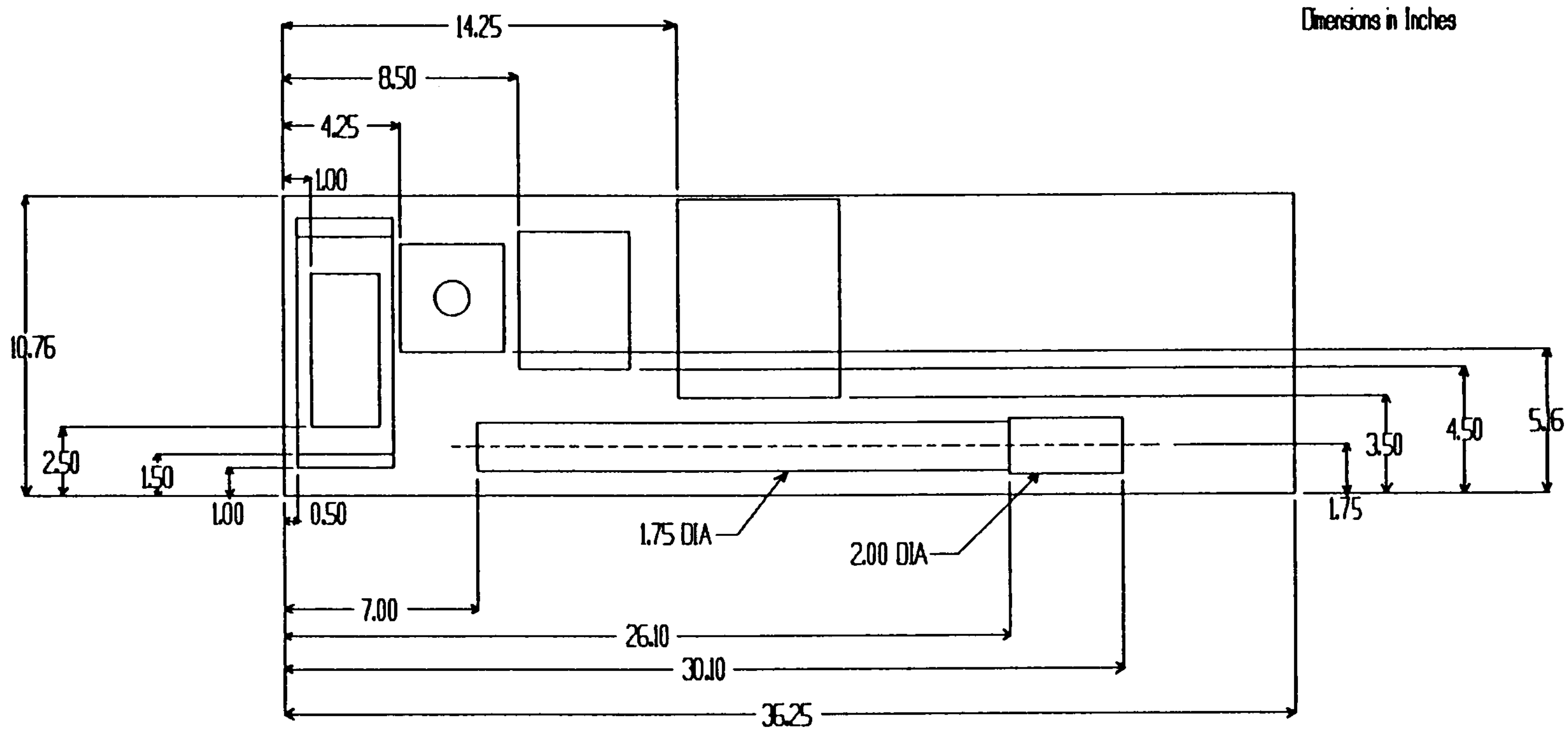


Figure F-3. Space Experiment Hardware Layout, Top View

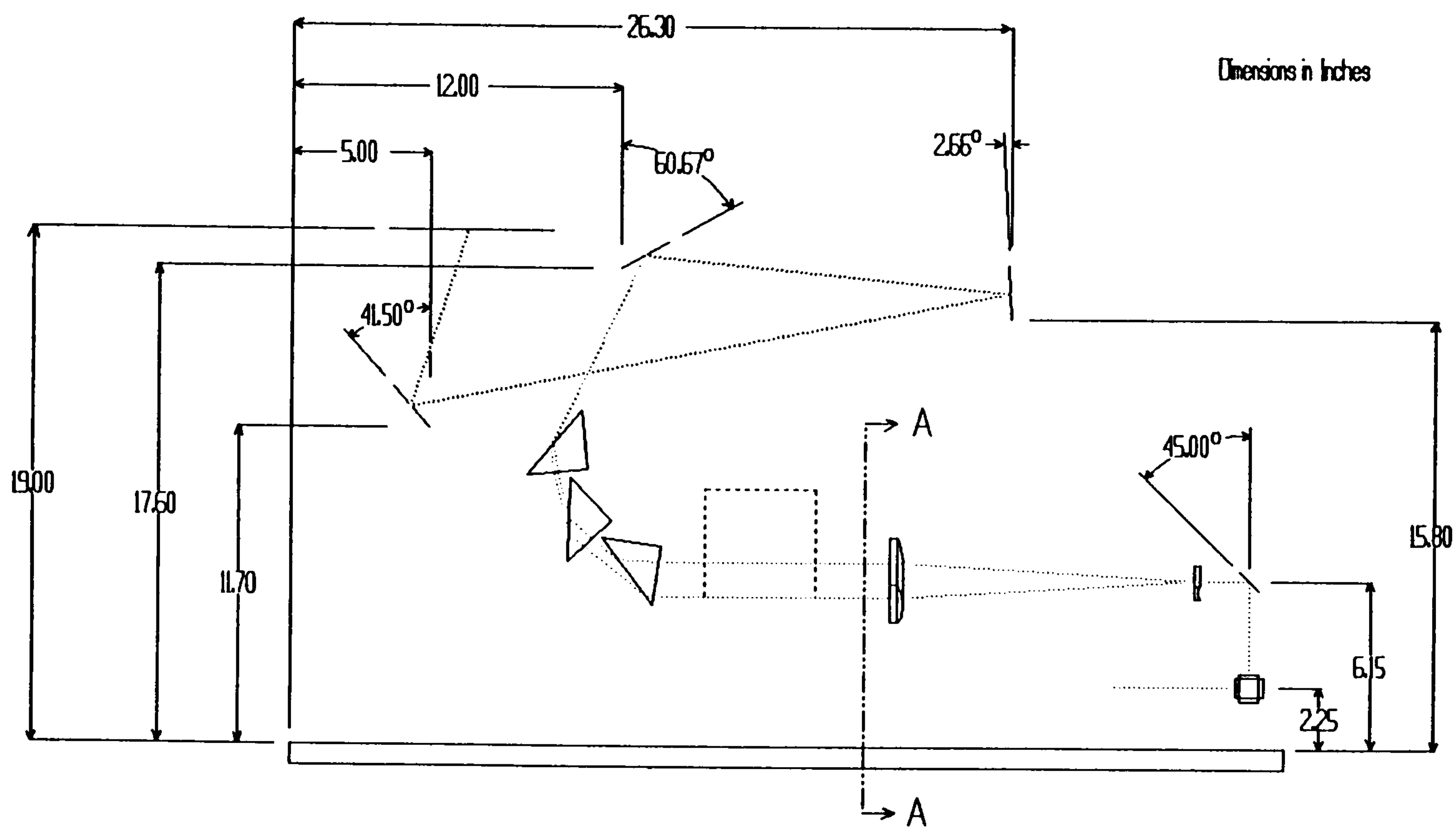


Figure F-4. Optical System Design, Front View

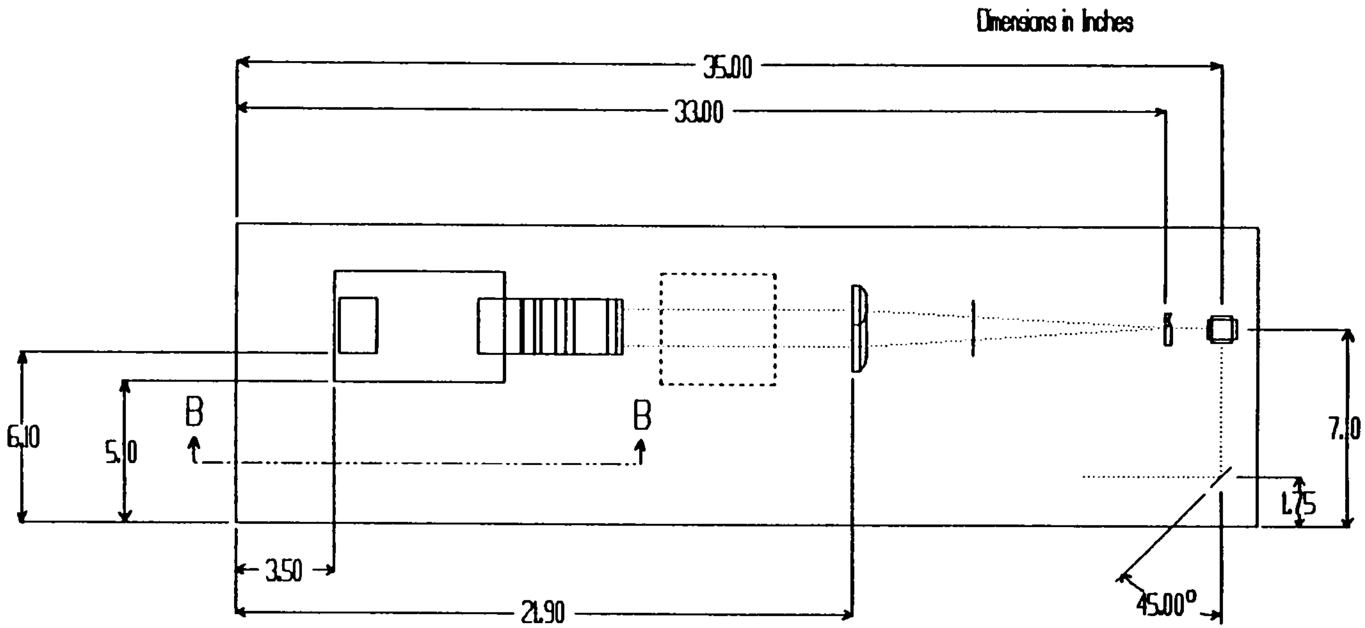


Figure F-5.   Optical System Design, Top View

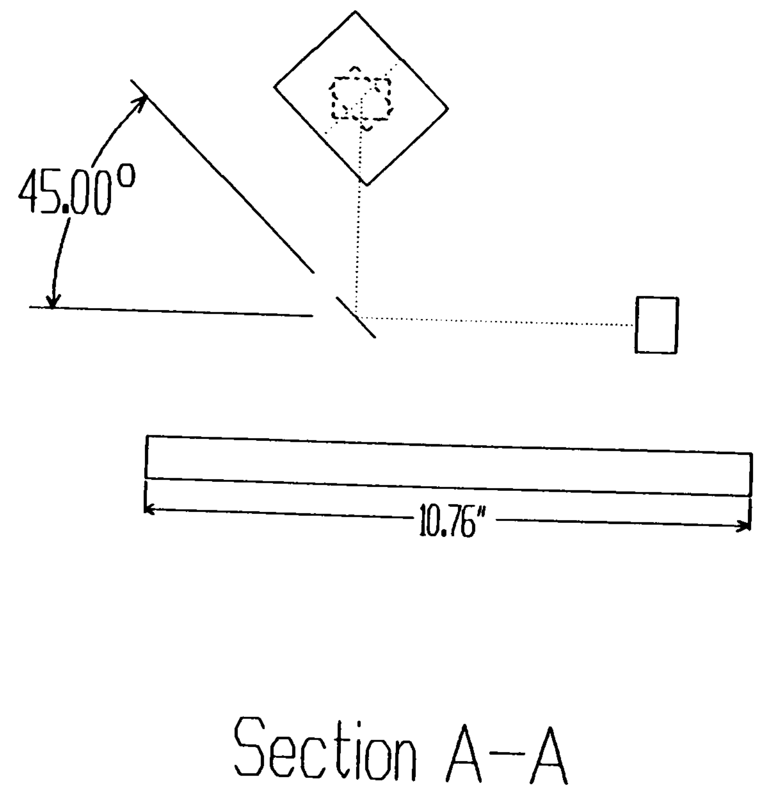


Figure F-6. Optical System Design, Section A-A

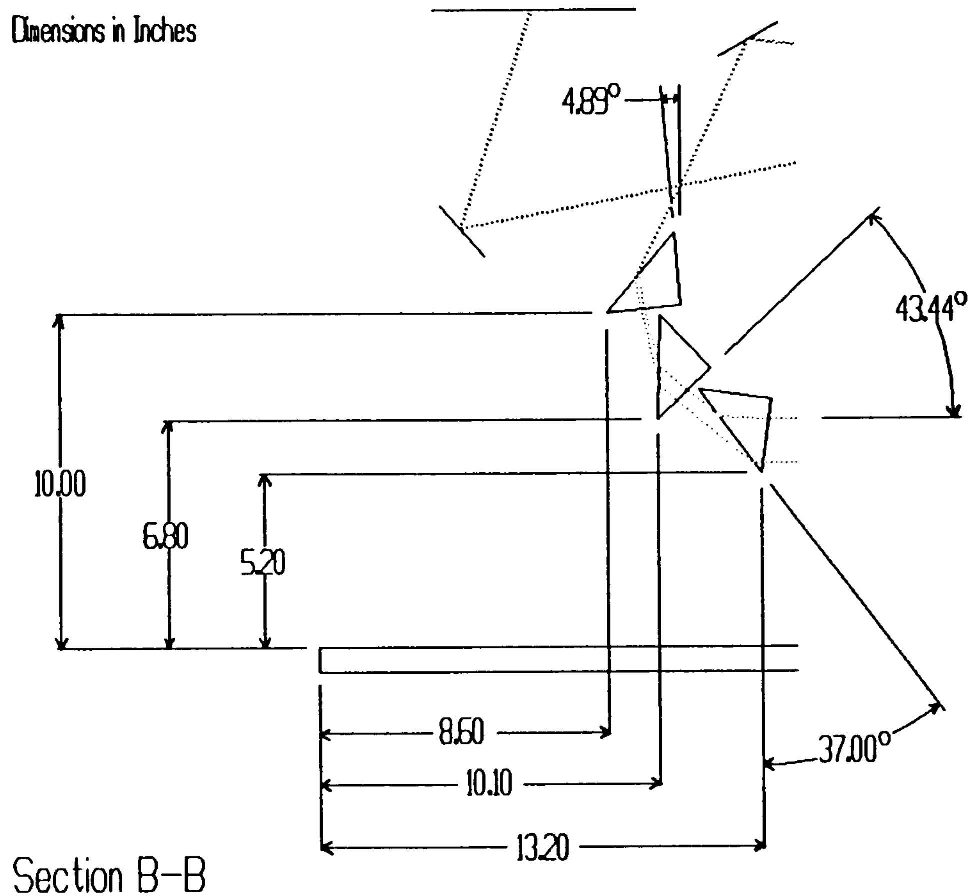


Figure F-7. Optical System Design, Section B-B

The specifications for selected off-the-shelf hardware that could be used in the space flight experiment are given below. These specifications were used in the baseline design.

1) Laser
    Supplier: Oriel
    Type: 10 mW He-Ne Laser; Model #79274; Includes power supply
    Beam Diameter: 0.68 mm @ $1/e^2$
    Beam Divergence: 1.2 mrad
    Laser Dimensions: length=19.1 in., diameter=1.75 in.
    Power Supply Dimensions: 5.5x5.5x2.5 inches
    Cost: $1180

2) Laser Holder
    Supplier: Oriel
    Type: Precision Vee-Block Laser Base; holds a 1.75 inch diameter laser head; beam will be 2.25 inches above laser base.
    Dimensions: 5.5x3.0 inch base
    Cost: $236

3) Beam Expander System
    Supplier: Oriel
    Components:
    a) Input lens model #15940, focal length=16 mm
    b) Beam expander model #15900, focal length=80 mm
    c) Laser Mounting Adapter #16140
    Description: Provides 5x magnification
    Dimension: Length=4.0 inches, diameter=2.0 inches
    Cost: $475

4) Optical Components
    Supplier: Melles Griot
    Components:
    a) Cylindrical lens #01LCP137, focal length=300 mm
    b) Cylindrical lens #01LCN125, focal length=-19 mm
    c) Prisms (qty 3) #01PRS037/066
    Description of Prisms: Right angle prisms with 3 sides of length 55 mm and hypotenuse of length 77.8 mm. Includes MgF_2 coating for hypotenuse and sides opposite hypotenuse.
    Cost: a) $363, b) $210, c) $329 per prism

5) Data Logger
Supplier: Campbell Scientific
Type: Data Logger model CR10
Description: An example of capability -- can read 12 thermocouples and reference and one pressure transducer and store the values about 5 times per second.
Dimensions: 9.0x3.5x2.9 inches
Weight: 2 pounds
Cost: $1530

6) Data Storage Module
Supplier: Campbell Scientific
Type: Data Storage Module Model SM716
Description: Can store 358,000 data values
Dimensions: 7.8x3.5x1.5 inches
Weight: 2 pounds
Cost: $990

7) Video Camera
Supplier: Canon
Type: Canon E06 8 mm Video Camera
Dimensions: 3.75x9.7x3.9 inches
Weight: 2.5 pounds
Cost: $1300

APPENDIX G

COMPUTER PROGRAMS

```
program TacProcess4;
{ ****************************************************************** }
{ TACPROC4.PAS  (Compiled with Borland Turbo Pascal 6.0)             }
{               (version 5.5 can be used with little modification)   }
{                                                                    }
{                                                                    }
{   -- Experimental Data Processing Program --                       }
{                                                                    }
{                                                                    }
{    This program combines several data processing steps for laser   }
{ deflection data and thermocouple data with conduction program      }
{ COND5C.PAS to provide a wide range of 'plotable' data in one       }
{ output file called TAC???.DAT.  All the input data is in           }
{ TAC???.IN.                                                         }
{    Additionally, TacProcess2 allows numerous parameters to be      }
{ varied for error band determination.                               }
{    Tacproc3 removes core temperature data from numerical work and  }
{ uses thermodynamic relationships.                                  }
{    Tacproc4 gets pressure from the unadjusted temperature profile  }
{ as well as the adjusted temperature profile.                       }
{ ****************************************************************** }

uses crt,utility;

const gam : double = 1.4;

type
  profile = array[0..60] of double;  dataout = array[1..201] of double;

var
  indat,outdat : text;
  time1out,time2out,timeavg,tmpavg,tavgcl : dataout;
  timepress,tc1out,tc2out,pressout,pressoutadj : dataout;
  nout1,nslice : integer;
  timeslice,pressure1,pressurea,walldtdy,wavydis : array[1..15] of double;
  del,T,dtdy,Ta1,Ta2 : profile;
  walldis : array[0..15,0..60] of single;
  ndef : array[0..15] of integer;
  ydefav,ydefcl : array[1..15,0..60] of double;
  discorr,defcorr : array[1..15,1..30] of single;
  L,Zs,ydepth,exvol,tcorea1,tcorea2 : double;
  Lerror,Zserror,pixelerror,walltemperror,pressureerror : double;
  testlengtherror,angleerror : double;
  addtowavydis : byte;

procedure TCPdata; forward;
procedure Deflect; forward;
procedure TempProf(n : byte); forward;
procedure Timesliceoutput(n : byte); forward;
procedure Presscalc(n:byte; temp:profile; var pressure:double); forward;
procedure Output2; forward;

procedure Go;
var
  runnumber : string[3]; inname,outname : string[12]; n : byte;

begin
  clrscr; gotoxy(10,3);
  write( 'Enter three digit run number (ex. 007 or 124) --> ');
  runnumber:=GetStr(3,-1,-1,''); inname := 'TAC'+runnumber+'_4.IN';
  outname := 'TAC'+runnumber+'_4.DAT';
  Openfile(indat,inname); if error <> 0 then stop;
  Createfile(outdat,outname);
```

```
  writeln(outdat,'          Output Data for Run ',runnumber);
  lff(outdat,1);

  Readcomments(indat);  readln(indat,L,Lerror,Zs,Zserror);
  L:=L+Lerror; Zs:=Zs+Zserror;
  Readcomments(indat);  readln(indat,ydepth);

  lfs(1);
  writeln(' Working  (TCPdata)');
  TCPdata;

  writeln(' Working  (Deflect)');
  Deflect;

  for n := 1 to nslice do
  begin
    writeln(' Working  (TempProf, time slice ',timeslice[n]:6:2,')');
    TempProf(n);
    Timesliceoutput(n);
    writeln(' Working  (PressCalc)');
    Presscalc(n,T,pressure1[n]);
    Presscalc(n,Ta1,pressurea[n]);
  end;

  Output2;

  close(outdat);
end;
{-------------------------------------------------------------------}
procedure TCPdata;

{   TCPdata reads in thermocouple and pressure data that is output  }
{  from the HP.  The input is time (msec), thermocouple 1 (R),      }
{  thermocouple 2 (R), (both thermocouples on the hot wall) and     }
{  pressure transducer (lbf/ft2).  The first set of output is each  }
{  thermocouple value versus time.  The second set of data is       }
{  average temperature vs. time and pressure versus time.  The times }
{  in each case are interpolated appropriately between readings.    }
{  Finally, the pressure is adjusted for the experimentally         }
{  determined pressure drop rate.                                   }

var
  n,i,j,k,istart,ic,ic1,ic2,ic3,ic4 : integer;
  a,b,c,d,tsum,zd,f1,tm,ct,at,tcp1,tcp2,pr,pradj : single;
  timestart,timedif,midtime,adjtime : double;
  timein,tc1,tc2,press : dataout;
  leaksum,leakrate : double;
  timeadj, droprate : array[1..15] of single;
  wttime,wta,wtb,wtc,wtd : array[1..25] of single;

begin
  Readcomments(indat); readln(indat,timestart);
  Readcomments(indat);
  n := 0;
  repeat
    inc(n); readln(indat,timein[n],tc1[n],tc2[n],press[n]);
  until timein[n] < 0.0;
  n:=n-1;

  time1out[1]:=0.0; time2out[1]:=0.0; tc1out[1]:=tc1[1]; tc2out[1]:=tc2[1];
  timeavg[1]:=0.0; tmpavg[1]:=(tc1[1]+tc2[1])/2.0;
  timepress[1]:=0.0; pressout[1]:=0.0;
  istart:=0;
  repeat  inc(istart); until timein[istart+1]>timestart;
```

```
j:=1;
for i := istart to n-1 do
begin
  inc(j);
  timedif := timein[i+1]-timein[i];
  adjtime := timein[i]-timestart;
  if timedif < 100.0 then
    begin writeln('timein data error'); stop; end;
  if timedif < 180.0 then
  begin
    time1out[j] := adjtime + 0.31*timedif;
    time2out[j] := adjtime + 0.62*timedif;
    timepress[j] := adjtime + 0.93*timedif;
  end
  else
  begin
    time1out[j] := adjtime + 0.56*timedif;
    time2out[j] := adjtime + 0.76*timedif;
    timepress[j] := adjtime + 0.96*timedif;
  end;
  tc1out[j] := tc1[i];
  tc2out[j] := tc2[i];
  pressout[j] := press[i];
  timeavg[j] := (time1out[j]+time2out[j] )/2.0;
  tmpavg[j] := (tc1out[j]+tc2out[j] )/2.0;
end;
inc(j); nout1:=j; adjtime := timein[n]-timestart;
time1out[j] := adjtime + 0.31*140.0;
time2out[j] := adjtime + 0.62*140.0;
timepress[j] := adjtime + 0.93*140.0;
tc1out[j] := tc1[n];
tc2out[j] := tc2[n];
pressout[j] := press[n];
timeavg[j] := (time1out[j]+time2out[j] )/2.0;
tmpavg[j] := (tc1out[j]+tc2out[j] )/2.0;
for i := 1 to nout1 do
begin
  time1out[i]:=time1out[i]/1000.0;
  time2out[i]:=time2out[i]/1000.0;
  timeavg[i]:=timeavg[i]/1000.0;
  timepress[i]:=timepress[i]/1000.0;
end;
if time1out[2] < 0.0 then time1out[2]:=0.0;
if time2out[2] < 0.0 then time2out[2]:=0.0;
if timeavg[2]  < 0.0 then timeavg[2] :=0.0;
if timepress[2] < 0.0 then timepress[2]:=0.0;

  {Adjust pressure for pressure drop rate}

Readcomments(indat);
i:=0;
repeat
  inc(i); readln(indat,timeadj[i],droprate[i]);
until timeadj[i] > 100.0;

pressoutadj[1] := pressout[1];
leaksum := 0.0;  j:=2;
for i := 2 to nout1 do
begin
  timedif := timepress[i]-timepress[i-1];
  midtime := (timepress[i]+timepress[i-1] )/2.0;
  while midtime > timeadj[j] do j:=j+1;
  leakrate:=Interpolate(timeadj[j-1],timeadj[j],
                        droprate[j-1],droprate[j],midtime);
```

```
    leaksum := leaksum + leakrate*timedif;
    pressoutadj[i] := pressout[i] + leaksum;
  end;

   { Find average temp. on centerline between glass walls.}

  Readcomments(indat);  i:=0;
  repeat
    inc(i);
    readln(indat,wttime[i],wta[i],wtb[i],wtc[i],wtd[i]);
  until wttime[i] < 0;
  wttime[i]:=9.9e9; wta[i]:=wta[i-1]; wtb[i]:=wtb[i-1];
  wtc[i]:=wtc[i-1]; wtd[i]:=wtd[i-1];

  ic:=0;
  for i := 1 to nout1 do
  begin
    while timeavg[i] >= wttime[ic+1] do inc(ic);
    c:=Interpolate(wttime[ic],wttime[ic+1],wtc[ic],wtc[ic+1],timeavg[i]);
    d:=Interpolate(wttime[ic],wttime[ic+1],wtd[ic],wtd[ic+1],timeavg[i]);

    tsum:=0.0;
    for k := 0 to 20 do
    begin
      zd:=k/20;
      f1:=1.0 + c*zd*zd + d*zd;
      if f1 > 1.0 then f1:=1.0;
      tsum:=tsum+f1*tmpavg[i];
    end;
    tavgcl[i]:=tsum/21.0;
  end;

  ic1:=0; ic2:=0; ic3:=0; ic4:=0;
  writeln(outdat,'--- Heated Wall Temperature and Measured Pressure');
  lff(outdat,1);
  writeln(outdat,
  '  time   Center   Average   TC1     TC2     Pressure   Adj Press');
  writeln(outdat,
  '         Temp.    Temp.                      (gage)     (gage)');
  writeln(outdat,
  '  (sec)   (R)      (R)      (R)     (R)    (lbf/ft2)  (lbf/ft2)');
  for i := 0 to 40 do
  begin
    tm:=0.25*i;
    while tm >= timeavg[ic1+1] do inc(ic1);
    while tm >= time1out[ic2+1] do inc(ic2);
    while tm >= time2out[ic3+1] do inc(ic3);
    while tm >= timepress[ic4+1] do inc(ic4);
    ct:=Interpolate(timeavg[ic1],timeavg[ic1+1],
                    tmpavg[ic1],tmpavg[ic1+1],tm);
    at:=Interpolate(timeavg[ic1],timeavg[ic1+1],
                    tavgcl[ic1],tavgcl[ic1+1],tm);
    tcp1:=Interpolate(time1out[ic2],time1out[ic2+1],
                    tc1out[ic2],tc1out[ic2+1],tm);
    tcp2:=Interpolate(time2out[ic3],time2out[ic3+1],
                    tc2out[ic3],tc2out[ic3+1],tm);
    pr:=Interpolate(timepress[ic4],timepress[ic4+1],
                    pressout[ic4],pressout[ic4+1],tm);
    pradj:=Interpolate(timepress[ic4],timepress[ic4+1],
                    pressoutadj[ic4],pressoutadj[ic4+1],tm);
    writeln(outdat,tm:6:2,ct:8:0,at:9:0,tcp1:9:0,tcp2:8:0,pr:9:0,pradj:9:0);
  end;

end;
```

```
{--------------------------------------------------------------------}
procedure Deflect;

{     DEFLECT converts (z,y) points of any scale of a deflected       }
{ laser line to deflection versus distance from the hot wall.  Note   }
{ that deflection is assumed to be completely in the y direction.     }

label 10;

var
  testlength,angle,zratio,yratio : double;
  time,imagelength,frac,hyp,z,yd,yi,wd1,dfc,av,cl,wyds : double;
  zo,yo : array[1..60] of double;
  i,j,n : integer;
  deffile : text;

begin
  Readcomments(indat);    readln(indat,exvol); exvol:=exvol-1.0;
  Readcomments(indat);    readln(indat,testlength,testlengtherror);
    testlength:=testlength+testlengtherror;
  Readcomments(indat);    readln(indat,angle,angleerror);
    angle:=angle+angleerror;
  Readcomments(indat);    readln(indat,zratio,yratio);
  Readcomments(indat);    readln(indat,wyds,addtowavydis);
  Readcomments(indat);    readln(indat,pixelerror);
  Readcomments(indat);    readln(indat,walltemperror);
  Readcomments(indat);    readln(indat,pressureerror);

{  Read in a series of (z,y) points for the undeflected beam beginning  }
{ at the top (i.e. next to hot wall).                                   }

  Readcomments(indat);
  i:=0;
  repeat
    inc(i);
    readln(indat,zo[i],yo[i]);
    zo[i]:=zo[i]/zratio; yo[i]:=yo[i]/yratio;
  until zo[i] < 0.0;
  ndef[0]:=i-1;

  angle := angle*3.14159265/180.0;
  imagelength := sqrt(sqr(zo[1]-zo[ndef[0]])+sqr(yo[1]-yo[ndef[0]]));
  for i := 1 to ndef[0] do
  begin
    frac := sqrt(sqr(zo[1]-zo[i])+sqr(yo[1]-yo[i]))/imagelength;
    hyp := testlength*frac;                                        {in}
    walldis[0,i]:=hyp*cos(angle);                                  {in}
  end;

  n:=0;
  repeat
    inc(n);
    Readcomments(indat);   readln(indat,timeslice[n]);
    if timeslice[n] < 0.0 then goto 10;
    Readcomments(indat); i:=0; j:=2;
    repeat
      inc(i);
      readln(indat,z,yd); yd:=yd+pixelerror;
      z := z/zratio; yd := yd/yratio;
      while (z < zo[j]) and (z > 0.0) do inc(j);
      walldis[n,i]:=Interpolate(zo[j-1],zo[j],
                       walldis[0,j-1],walldis[0,j],z);
      if i=1 then wd1:=walldis[n,1];
      walldis[n,i]:=walldis[n,i]-wd1;
```

```
      yi:=Interpolate(zo[j-1],zo[j],yo[j-1],yo[j],z);
      ydefav[n,i] := yd-yi;
    until z < 0.0;
    ndef[n]:=i-1;

    if addtowavydis = 1 then wavydis[n]:=wyds+wd1 else wavydis[n]:=wyds;
    walldis[n,0]:=0.0;
    for i:=1 to ndef[n] do walldis[n,i]:=walldis[n,i]+wavydis[n];

    Readcomments(indat); i:=0; j:=2;
    repeat
      inc(i);
      readln(indat,discorr[n,i],av,cl);
      if av > 0.0001 then defcorr[n,i]:=cl/av;
    until discorr[n,i] < 0.0;
    for i := 1 to ndef[n] do
    begin
      while walldis[n,i] > discorr[n,j] do inc(j);
      dfc:=Interpolate(discorr[n,j-1],discorr[n,j],defcorr[n,j-1],
                       defcorr[n,j],walldis[n,i]);
      ydefcl[n,i]:=dfc*ydefav[n,i];
    end;

    10:
  until timeslice[n] < 0.0;
  nslice := n-1;
end;
{-------------------------------------------------------------------}
procedure TempProf; {n : byte}

{    TempProf uses deflection versus distance from the hot wall and    }
{  pressure to calculate dTdy versus distance from the wall using      }
{  the known relationship between dTdy and the refraction of the       }
{  laser.  With dTdy and the known temperature at the wall, the        }
{  temperature profile of the gas vs. distance from the wall is        }
{  calculated.                                                         }
{    Finally, if dTdy reaches zero, it is known that the               }
{  temperature should be equal to the core temperature at that         }
{  point.  Inaccuracy in the deflection data will generally cause the  }
{  integration to "miss", so the temperature profile is adjusted so    }
{  that the temperature far from the wall equals the core temp.  The   }
{  core temp is found from an energy balance.  This gives Ta1.         }
{  Another adjusted temp profile, Ta2, is obtained by matching the     }
{  measured pressure to the calculated pressure.                       }

const
  indexm1  : double = 2.719e-4;
  idensity : double = 0.0757;          {lbm/ft3}
  r        : double = 53.34;           {(ft-lbf)/(lbm-R)}

var
  i,j,nparam,ndn,ndnm,ndnp : integer;
  press,pressc,k1,Ttest,Tdif,Tdifmax : double;
  frac,Toff,Tden,ddtdy : double;
  rowbound,rowcore,tavg,wdis,tcorea1old,tcorea1dif : double;
  tcorea2old,tcorea2dif,th,tl : double;

begin
         {Get hot wall temperature (R)}
  i:=2;
  while (timeavg[i] < timeslice[n]) and (i <= nout1) do inc(i);
  T[0]:=Interpolate(timeavg[i-1],timeavg[i],
                    tavgcl[i-1],tavgcl[i],timeslice[n])+walltemperror;
```

```
        {Get pressure (lbf/ft2)}
  i:=2;
  while (timepress[i] < timeslice[n]) and (i <= nout1) do inc(i);
  press:=Interpolate(timepress[i-1],timepress[i],
                     pressout[i-1],pressout[i],timeslice[n]) + 2117.0;
  press:=press+pressureerror;

  ndn:=ndef[n]; ndnp:=ndn+1;
  for i:=0 to ndn do
  begin
    dtdy[i]:=0.0; if i <> 0 then T[i]:=0.0;
    if i <> 0 then del[i]:=walldis[n,i] - walldis[n,i-1];
  end;

{First cut on temperature profile}

  k1 := -(idensity*r)/(indexm1*press*L*Zs*144.0);          {1/R-in2}
  repeat
    Ttest:=T[1];
    dtdy[1]:=k1*T[1]*T[1]*ydefav[n,1];  dtdy[0]:=dtdy[1];       {R/in}
    T[1]:=T[0] + del[1]*dtdy[0];
    Tdif:=(Ttest-T[1])/T[1];
  until abs(Tdif) < 0.00001;
  ydefcl[n,0]:=dtdy[0]/(k1*T[0]*T[0]);
  ydefav[n,0]:=ydefcl[n,0]/defcorr[n,1];
  for i:=2 to ndn do
  begin
    repeat
      Ttest:=T[i];
      T[i] := T[i-1]+del[i]*(dtdy[i]+dtdy[i-1])/2.0;
      dtdy[i]:=k1*T[i]*T[i]*ydefav[n,i];
      Tdif:=(Ttest-T[i])/T[i];
    until abs(Tdif) < 0.0001;
  end;

{Final determination of temperature profile}

  T[ndnp]:=T[ndn]; dtdy[ndnp]:=0.0;
  del[ndnp]:=del[ndn];
  repeat
    Tdifmax:=0.0;
    for i:=2 to ndn do
    begin
      Ttest:=T[i];
      T[i] := ((T[i-1]+T[i+1])/2.0)
             +((del[i]/4.0)*(dtdy[i-1]+dtdy[i]))
             -((del[i+1]/4.0)*(dtdy[i]+dtdy[i+1]));
      dtdy[i]:=k1*T[i]*T[i]*ydefav[n,i];
      if T[i] > 0.1 then Tdif:=abs((Ttest-T[i])/T[i]);
      if Tdif > Tdifmax then Tdifmax:=Tdif;
    end;
    T[ndnp]:=T[ndn] + del[ndn]*dtdy[ndn];
    dtdy[ndnp]:=dtdy[ndn];
  until Tdifmax < 1.0e-6;

      {Save wall temperature derivative for output}
  walldtdy[n]:=dtdy[1];

     {Extend the derivative linearly until dtdy = 0.0 and extend temp.}

  if (dtdy[ndn] < -0.1) then
  begin
    ndef[n] := ndnp; ndn:=ndnp; ndnm:=ndn-1; ddtdy:=0.0;
    ydefcl[n,ndn]:=0.0;   ydefav[n,ndn]:=0.0;
```

```
    dtdy[ndn]:=0.0;
    for i:=2 to 5 do
      ddtdy:=ddtdy + (dtdy[ndnm]-dtdy[ndn-i])/
                     (walldis[n,ndnm]-walldis[n,ndn-i]);
    ddtdy:=ddtdy/4.0;
    if ddtdy < 200.0 then ddtdy:=200.0;
    walldis[n,ndn]:=walldis[n,ndnm] - (dtdy[ndnm]/ddtdy);
    T[ndn]:=T[ndnm] + (walldis[n,ndn]-walldis[n,ndnm])*dtdy[ndnm]/2.0;
  end;

    { get core temperature 1 }

  tcorea1:=tavgcl[1];   press:=press/2117.0;  Tden:=T[0]-T[ndn];
  repeat

     { Adjust the temperature profile to get Ta1}

    Toff := T[ndn]-tcorea1;
    for i:=0 to ndn do
      Ta1[i] := T[i] - Toff*(T[0]-T[i])/Tden;

    rowbound:=0.0;
    for i:=1 to ndn do
    begin
      tavg:=Avg(Ta1[i-1],Ta1[i])/tavgcl[1];
      wdis:=(walldis[n,i]-walldis[n,i-1])/(ydepth*12.0);
      rowbound:=rowbound + (press/tavg)*wdis;
    end;
    wdis:=walldis[n,ndn]/(ydepth*12.0);
    rowbound:=rowbound/wdis;
    rowcore:=1.0 + (1.0-rowbound)*(wdis/(1.0+exvol-wdis));
    tcorea1old:=tcorea1;
    tcorea1:=tavgcl[1]*(1.0 + (gam/2.0)*(rowcore-1.0))/
                     (rowcore - (gam/2.0)*(rowcore-1.0));
    tcorea1dif:=abs(tcorea1-tcorea1old)/tcorea1;
  until tcorea1dif < 0.00001;

  inc(ndef[n]);
  walldis[n,ndef[n]]:=ydepth*12.0;
  Ta1[ndef[n]]:=tcorea1; T[ndef[n]]:=T[ndef[n]-1];

    { get Ta2 by matching measured and calculated pressure }

  th:=1500.0; tl:= 100.0;  tcorea2:=800.0;
  press := press*2117.0;
  repeat

     { Adjust the temperature profile to get Ta2}
    Toff := T[ndn]-tcorea2;
    for i:=0 to ndn do
      Ta2[i] := T[i] - Toff*(T[0]-T[i])/Tden;
    Ta2[ndef[n]]:=tcorea2;
    Presscalc(n,Ta2,pressc);   pressc:=pressc+2117.0;
    tcorea2old:=tcorea2;
    if pressc > press then th:=tcorea2 else tl:=tcorea2;
    tcorea2:=Avg(th,tl);
    tcorea2dif:=abs(tcorea2-tcorea2old)/tcorea2;
  until tcorea2dif < 0.00001;
end;
{-----------------------------------------------------------------}
procedure Timesliceoutput; {n : byte}
var i : byte;
begin
  if n=1 then
```

```
    begin
      lff(outdat,5);
      writeln(outdat,'Deflection error =              ',pixelerror:8:2,' pixels');
      writeln(outdat,'Wall temperature error =        ',walltemperror:8:0,' R');
      writeln(outdat,'Test section length error =     ',Lerror:8:5,' feet');
      writeln(outdat,'Laser path length error =       ',Zserror:8:5,' feet');
      writeln(outdat,'Measured pressure error =       ',pressureerror:8:1,
                    ' lbf/ft2');
      writeln(outdat,'Width of laser sheet error = ',Testlengtherror:8:3,
                    ' inches');
      writeln(outdat,'Angle of laser sheet error = ',angleerror:8:1,
                    ' degrees');
      writeln(outdat,'Extra volume = ',exvol:6:3);
    end;

    lff(outdat,5);
    writeln(outdat,'--- Results for Time Slice ',timeslice[n]:6:2,' seconds');
    lff(outdat,2);
    writeln(outdat,'Wavy wall error = ',wavydis[n]:6:3,' inches');
    lff(outdat,1);
    writeln(outdat,' Deflection & Temperature Data vs Distance from Hot Wall');
    lff(outdat,1);
    writeln(outdat,'  walldis   Adjusted   Adjusted     Temp.   ydefav',
                   '  ydefcl    dTdy');
    writeln(outdat,'              Temp.      Temp.');
    writeln(outdat,'            match core  match press');
    writeln(outdat,'   (in)       (R)        (R)        (R)     (in) ',
                   '   (in)    (R/in)');
    lff(outdat,1);
    for i := 0 to ndef[n]-1 do
      if (i<>0) or (wavydis[n] <> 0.0) then
        writeln(outdat,walldis[n,i]:8:5,Ta1[i]:10:0,Ta2[i]:11:0,T[i]:11:0,
                       ydefav[n,i]:9:2,ydefcl[n,i]:9:2,dtdy[i]:9:0);
    writeln(outdat,walldis[n,ndef[n]]:8:5,Ta1[ndef[n]]:10:0,Ta2[ndef[n]]:11:0,
                   T[ndef[n]]:11:0);
{   for i := 0 to ndef[n]-1 do
      if (i<>0) or (wavydis[n] <> 0.0) then
        writeln(outdat,walldis[n,i]:8:5,'                              ',T[i]:11:0);
    writeln(outdat,walldis[n,ndef[n]]:8:5,'                              ',
                   T[ndef[n]]:11:0); }
end;
{------------------------------------------------------------------}
procedure Presscalc; {n:byte; temp:profile; var pressure:double}

{ PressCalc determines the pressure from a given temperature profile. }

var
  cavdepth,densitysum,tavg,press,ph,pl,densdif,tcoren : double;
  i : integer;  y, tempn : profile;

begin
  cavdepth := ydepth*12.0;
  for i := 1 to ndef[n] do
  begin
    y[i]:=walldis[n,i]/cavdepth;
    tempn[i]:=temp[i]/tavgcl[1];
  end;
  tcoren := tempn[ndef[n]];

  ph:=4.0; pl:=0.5;
  repeat
    densitysum := 0.0;
    press:=Avg(ph,pl);
    for i := 2 to ndef[n] do
```

```
    begin
      tavg := (tempn[i]+tempn[i-1])/2.0;
      densitysum := densitysum + press*(y[i]-y[i-1])/tavg;
    end;
    densitysum := densitysum + press*exvol/tcoren;
    if densitysum > (1.0+exvol) then ph:=press else pl:=press;
    densdif := abs(densitysum-(1.0+exvol));
  until densdif < 0.0005;
  pressure := (press-1.0)*2117.0;          {gauge pressure, lbf/ft2}

end;
{-----------------------------------------------------------------}
procedure Output2;
 {Output pressure and wall temperature derivative}
var i:byte;
begin
  lff(outdat,4);
  writeln(outdat,'--- Pressure (from temp. profile) and wall temperature');
  writeln(outdat,'    derivative at each time slice');
  lff(outdat,2);
  writeln(outdat,'  Time    Unadjusted   Adjusted    wall dTdy');
  writeln(outdat,'           Pressure    Pressure     (minus)');
  writeln(outdat,'            (gauge)     (gauge)');
  writeln(outdat,'  (sec)    (lbf/ft2)   (lbf/ft2)     (R/in)');
  lff(outdat,1);
  writeln(outdat,'  0.00         0           0             0');
  for i := 1 to nslice do
    writeln(outdat,timeslice[i]:6:2,pressure1[i]:11:0,
                pressurea[i]:12:0,-walldtdy[i]:13:0);
end;
{-----------------------------------------------------------------}
begin Go end.
```

```
{s-,v-}
program cond5c;
uses crt,utility,dos;
{ ******************************************************************** }
{  COND5C.PAS  (Compiled with Borland Turbo Pascal 6.0)               }
{              (ver 5.5 can be used with little modification)          }
{                                                                      }
{                                                                      }
{    --  Three-Dimensional Expanding Conduction Model --               }
{ ******************************************************************** }

const
{----Geometrical data and initial conditions - Air initial properties}

  cavheight : extended = 0.3333333;      {ft, j - perp to heated plate}
  cavwidth  : extended = 0.3333333;      {ft, i - dis between bus bars}
  cavlength : extended = 0.3333333;      {ft, k - dis between glass walls}
  laserlength : extended = 72.21;        {ft,   - distance to screen}
  irefrac   : extended = 2.719e-4;       {unitless}
  roo : extended = 0.0745;               {lbm/ft3}
  po : extended = 2117.0;                {lbf/ft2}
  cvo : extended = 133.0;                {(ft-lbf)/(lbm-R)}
  xko : extended = 0.00321;              {(ft-lbf)/(ft-sec-R)}
  alfo : extended = 3.238e-4;            {ft2/sec}
  rgas : extended = 53.34;               {(ft-lbf)/(lbm-R)}
  agm1 : extended = -0.0256211;          {gamma - 1 curve fit constants}
  bgm1 : extended =  0.431122;
  ak   : extended = -0.0335744;
  bk   : extended =  0.789609;           {k curve fit constants}
  ck   : extended =  0.244055;
  acv  : extended =  0.074404;           {cv curve fit constants}
  bcv  : extended =  0.925596;
  xznodemax = 11; ynodemax = 42; {max size of arrays, not actual nodes}

type
  nodeptr  = ^noderec;
  noderec = record  tn,told,toldi,toldii,v,rn,apo,save : extended; end;
  nodes = array[1..xznodemax,1..ynodemax,1..xznodemax] of nodeptr;
  ycolumn   = array[1..ynodemax] of extended;

var
  n : nodes;
  aa, bb, cc, rhs, dtdy, ysccl, yscav, evr, evrold, evt  : ycolumn;
  evmassgainperlevel, evmassgainpernode, evdensgainpernode : ycolumn;
  exvol,exvperlevel : extended;  {excess volume}

  ikmax, ikmaxm, ikmaxp, ik, jmax, jmaxm, jmaxm2 : byte;
  delxz, dely, xl,yl,zl, dxodt,dyodt,dzodt : extended;
  dxody,dxodz,dyodx,dyodz,dzodx,dzody,delyo2,delyt2 : extended;
  dytdz, dxtdz, dxtdy, dytdzodx, dxtdzody, dxtdyodz : extended;
  dxtdytdzodt, dxtdztdt, dtodx, dtody, dtodz, sor, dv : extended;
  rdelt, delt, tend, rtend, ti, pavg  : extended;
  rsum, rsumnormal, rsumcheck, rowrsum : extended;
  ppg,rts,walldtdy : array[1..200] of single;  pcnt : byte;

  intout : array[1..25] of longint;
  routint,outint : array[0..25] of double;
  mxstep, nout, nstep, oldnstep : longint;  ocnt,ocnt2 : byte;
  time, rtime, oldtime, newtime, cputime : double;
  hour, minute, second, fraction : word;

  outdat,indat : text;
  iout : array[1..50] of integer;
  io, iz1, iz2 : integer;
```

```
  wtime,wtemp : array[1..120] of single;             ic1 : integer;
  wttime,wta,wtb,wtc,wtd : array[1..25] of single;   ic2 : integer;
  bwtime, bwtemp : array[1..10] of single;           ic3 : integer;
  swctime, switime : array[1..12] of single;     ic4,ic5 : integer;
  swctmp : array[1..10,1..12] of single;
  switmp : array[1..5,1..12] of single;
  swcyd, swiyd : array[0..6] of single;
  swcxd : array[1..3] of single;

procedure BoundaryConditions; forward;
procedure TDMA( jend,i,k : integer); forward;
function  Powerlaw(peclet : extended) : extended; forward;
function  Xk(t : extended) : extended; forward;
function  Xcv(t : extended) : extended; forward;
procedure Grid; forward;
procedure Output; forward;
procedure Timer; forward;
procedure Init; forward;
procedure Go;

var
  i,j,k,im,jm,km,ip,jp,kp,sides,itloop,otloop          : integer;
  gamm1,tw,te,kw,ke,cvw,cve,cv,dw,de,aw,ae              : extended;
  ts,tn,ks,kn,cvs,cvn,ds,dn,fs,fn                       : extended;
  ps,pn,maxfs,minfn,dsPl,dnPl,as,an                     : extended;
  tb,tt,kb,kt,cvb,cvt,db,dt,ab,at                       : extended;
  b,tdif,maxtdif,innertdif,innermaxtdif,rcrit           : extended;
  rdif,maxrdif,center,flowchange,velchange              : extended;
  isv,jsv,ksv : integer;

begin
  Init;
  Grid;
  lfs(2); wrtln('             BEGIN  COND5C');
  repeat
    nstep := nstep+1;
    time := nstep*delt;
    rtime := (time*yl*yl)/alfo;

    BoundaryConditions;

    for i := 2 to ikmax do
      for j := 2 to jmaxm do               {save old temps and apo}
        for k := 2 to ikmax do
        begin
          n[i,j,k]^.told := n[i,j,k]^.tn;
          n[i,j,k]^.apo := n[i,j,k]^.rn*dxtdytdzodt;
        end;
    for j := 2 to jmaxm do evrold[j]:=evr[j];

    maxtdif:=9.9;  otloop:=0;
    repeat
      otloop:=otloop+1;

      for i := 2 to ikmax do
        for j := 2 to jmaxm do            {save old iteration to adjust}
          for k := 2 to ikmax do              {relaxation parameter}
            n[i,j,k]^.toldii:=n[i,j,k]^.toldi;

      for i := 2 to ikmax do
        for j := 2 to jmaxm do            {save temps for outer iteration}
          for k := 2 to ikmax do
            n[i,j,k]^.toldi:=n[i,j,k]^.tn;
```

```
{------Solve the energy equation for n[i,j,k]^.tn-----}
  {...Solve at each (i,k) and iterate...}

    itloop:=0;
    repeat
      itloop:=itloop+1;

      for i := 2 to ikmax do
        for j := 2 to jmaxm do          {save temps for inner iteration}
          for k := 2 to ikmax do
            n[i,j,k]^.save:=n[i,j,k]^.tn;
      for i := 2 to ikmax do
        for k := 2 to ikmax do
        begin
          im:=i-1; ip:=i+1; km:=k-1; kp:=k+1;
          gamm1 := agm1*n[i,2,k]^.tn + bgm1;
          tw := Avg(n[im,2,k]^.tn,n[i,2,k]^.tn);
                if i = 2 then tw := n[1,2,k]^.tn;
          te := Avg(n[i,2,k]^.tn,n[ip,2,k]^.tn);
          ts := n[i,1,k]^.tn;
          tn := Avg(n[i,2,k]^.tn,n[i,3,k]^.tn);
          tb := Avg(n[i,2,km]^.tn,n[i,2,k]^.tn);
                if k = 2 then tb := n[i,2,1]^.tn;
          tt := Avg(n[i,2,k]^.tn,n[i,2,kp]^.tn);
          kw := Xk(tw);      ke := Xk(te);      ks := Xk(ts);
          kn := Xk(tn);      kb := Xk(tb);      kt := Xk(tt);
          cvw := Xcv(tw);  cve := Xcv(te);  cvs := Xcv(ts);
          cvn := Xcv(tn);  cvb := Xcv(tb);  cvt := Xcv(tt);
          cv := (cvw+cve+cvs+cvn+cvb+cvt)/6.0;
          dw := (kw/cvw)*dytdzodx;
                if i = 2 then dw := dw*2.0;
          de := (ke/cve)*dytdzodx;
          ds := (ks/cvs)*dxtdzody*2.0;
          dn := (kn/cvn)*dxtdzody;
          db := (kb/cvb)*dxtdyodz;
                if k = 2 then db := db*2.0;
          dt := (kt/cvt)*dxtdyodz;
          fn := Avg(n[i,2,k]^.rn,n[i,3,k]^.rn)*n[i,2,k]^.v*dxtdz;
          pn := abs(fn/dn);
          if fn < 0 then minfn:=fn else minfn:=0.0;
          dnPl := dn*Powerlaw(pn);
          aw := dw;
          ae := de;
          as := ds;
          an := dnPl - minfn;
          ab := db;
          at := dt;
          b  := -(gamm1/cv)*pavg*n[i,2,k]^.v*dxtdz
                + n[i,2,k]^.apo*n[i,2,k]^.told;
          bb[2]  := aw+ae+as+an+ab+at + n[i,2,k]^.apo;
          cc[2]  := -an;
          rhs[2] := b + aw*n[im,2,k]^.tn + ae*n[ip,2,k]^.tn
                      + as*n[i,1,k]^.tn
                      + ab*n[i,2,km]^.tn + at*n[i,2,kp]^.tn;

          for j := 3 to jmaxm2 do
          begin
            jm:=j-1; jp:=j+1;
            gamm1 := agm1*n[i,j,k]^.tn + bgm1;
            tw := Avg(n[im,j,k]^.tn,n[i,j,k]^.tn);
                  if i = 2 then tw := n[1,j,k]^.tn;
            te := Avg(n[i,j,k]^.tn,n[ip,j,k]^.tn);
            tn := Avg(n[i,j,k]^.tn,n[i,jp,k]^.tn);
            tb := Avg(n[i,j,km]^.tn,n[i,j,k]^.tn);
```

```
          if k = 2 then tb := n[i,j,1]^.tn;
    tt := Avg(n[i,j,k]^.tn,n[i,j,kp]^.tn);
    kw := Xk(tw);     ke := Xk(te);
    kn := Xk(tn);     kb := Xk(tb);     kt := Xk(tt);
    cvw := Xcv(tw);  cve := Xcv(te);  cvs := cvn;
    cvn := Xcv(tn);  cvb := Xcv(tb);  cvt := Xcv(tt);
    cv := (cvw+cve+cvs+cvn+cvb+cvt)/6.0;
    dw := (kw/cvw)*dytdzodx;
          if i = 2 then dw := dw*2.0;
    de := (ke/cve)*dytdzodx;
    dn := (kn/cvn)*dxtdzody;
    db := (kb/cvb)*dxtdyodz;
          if k = 2 then db := db*2.0;
    dt := (kt/cvt)*dxtdyodz;
    fs := fn;
    fn := Avg(n[i,j,k]^.rn,n[i,jp,k]^.rn)*n[i,j,k]^.v*dxtdz;
    pn := abs(fn/dn);
    if fs > 0 then maxfs:=fs else maxfs:=0.0;
    if fn < 0 then minfn:=fn else minfn:=0.0;
    dsPl := dnPl;
    dnPl := dn*Powerlaw(pn);
    aw := dw;
    ae := de;
    as := dsPl + maxfs;
    an := dnPl - minfn;
    ab := db;
    at := dt;
    b  := -(gamm1/cv)*pavg*(n[i,j,k]^.v-n[i,jm,k]^.v)*dxtdz
          + n[i,j,k]^.apo*n[i,j,k]^.told;
    aa[j]  := -as;
    bb[j]  := aw+ae+as+an+ab+at + n[i,j,k]^.apo;
    cc[j]  := -an;
    rhs[j] := b + aw*n[im,j,k]^.tn + ae*n[ip,j,k]^.tn
                + ab*n[i,j,km]^.tn + at*n[i,j,kp]^.tn;
end;

gamm1 := agm1*n[i,jmaxm,k]^.tn + bgm1;
tw := Avg(n[im,jmaxm,k]^.tn,n[i,jmaxm,k]^.tn);
      if i = 2 then tw := n[1,jmaxm,k]^.tn;
te := Avg(n[i,jmaxm,k]^.tn,n[ip,jmaxm,k]^.tn);
tn := n[i,jmax,k]^.tn;
tb := Avg(n[i,jmaxm,km]^.tn,n[i,jmaxm,k]^.tn);
      if k = 2 then tb := n[i,jmaxm,1]^.tn;
tt := Avg(n[i,jmaxm,k]^.tn,n[i,jmaxm,kp]^.tn);
kw := Xk(tw);     ke := Xk(te);
kn := Xk(tn);     kb := Xk(tb);     kt := Xk(tt);
cvw := Xcv(tw);  cve := Xcv(te);  cvs := cvn;
cvn := Xcv(tn);  cvb := Xcv(tb);  cvt := Xcv(tt);
cv := (cvw+cve+cvs+cvn+cvb+cvt)/6.0;
dw := (kw/cvw)*dytdzodx;
      if i = 2 then dw := dw*2.0;
de := (ke/cve)*dytdzodx;
dn := (kn/cvn)*dxtdzody*2.0;
db := (kb/cvb)*dxtdyodz;
      if k = 2 then db := db*2.0;
dt := (kt/cvt)*dxtdyodz;
fs := fn;
if fs > 0 then maxfs:=fs else maxfs:=0.0;
dsPl := dnPl;
aw := dw;
ae := de;
as := dsPl + maxfs;
an := dn;
ab := db;
```

```
          at := dt;
          b  := -(gamm1/cv)*pavg*(-n[i,jmaxm2,k]^.v)*dxtdz
                + n[i,jmaxm,k]^.apo*n[i,jmaxm,k]^.told;
          aa[jmaxm]  := -as;
          bb[jmaxm]  := aw+ae+as+an+ab+at + n[i,jmaxm,k]^.apo;
          rhs[jmaxm] := b + aw*n[im,jmaxm,k]^.tn + ae*n[ip,jmaxm,k]^.tn
                      + an*n[i,jmax,k]^.tn
                      + ab*n[i,jmaxm,km]^.tn + at*n[i,jmaxm,kp]^.tn;

          TDMA( jmaxm,i,k);
        end;

      innermaxtdif := 0.0;
      for i := 2 to ikmax do
        for j := 2 to jmaxm do
          for k := 2 to ikmax do
          begin
            innertdif:=abs( n[i,j,k]^.tn-n[i,j,k]^.save);
            if innertdif > innermaxtdif then innermaxtdif:=innertdif;
          end;
    until innermaxtdif < 1.0e-6;

  {....Relax temperatures.....}

    for i := 2 to ikmax do
      for j := 2 to jmaxm do
        for k := 2 to ikmax do
          n[i,j,k]^.tn := n[i,j,k]^.toldi
                        + sor*(n[i,j,k]^.tn - n[i,j,k]^.toldi);

  {... Set extra volume temperatures...}

    for j := 2 to jmaxm do evt[j]:={Avg( n[1,j,ikmax]^.tn,1.0)}1.0;

{.... Iterate to get new pressure and density ....}

    repeat

  {----Determine average pressure}

      pavg := 0.0;
      for i := 2 to ikmaxm do
        for j := 2 to jmaxm do
          for k := 2 to ikmaxm do
            pavg := pavg + (n[i,j,k]^.rn*n[i,j,k]^.tn);
      pavg := 4.0*pavg;
      center:=0.0;
      for i:= 2 to ikmaxm do
        for j := 2 to jmaxm do
          center := center + (n[i,j,ikmax]^.rn*n[i,j,ikmax]^.tn);
      for j:= 2 to jmaxm do
        for k := 2 to ikmaxm do
          center := center + (n[ikmax,j,k]^.rn*n[ikmax,j,k]^.tn);
      center := 2.0*center;
      for j := 2 to jmaxm do
        center := center + (n[ikmax,j,ikmax]^.rn*n[ikmax,j,ikmax]^.tn);
      pavg := (pavg + center)*dv;
      for j := 2 to jmaxm do
        pavg:=pavg + exvperlevel*evr[j]*evt[j];
      pavg:=pavg/(1.0+exvol);

  {----Determine new density and correct}

      for i := 2 to ikmax do
```

```
      for j := 2 to jmaxm do
        for k := 2 to ikmax do
          n[i,j,k]^.rn := pavg/n[i,j,k]^.tn;
    for j := 2 to jmaxm do evr[j] := pavg/evt[j];

    rsum := 0.0;
    for i := 2 to ikmaxm do
      for j := 2 to jmaxm do
        for k := 2 to ikmaxm do
          rsum := rsum + n[i,j,k]^.rn;
    rsum := 4.0*rsum;
    center:=0.0;
    for i:= 2 to ikmaxm do
      for j := 2 to jmaxm do
        center := center + n[i,j,ikmax]^.rn;
    for j:= 2 to jmaxm do
      for k := 2 to ikmaxm do
        center := center + n[ikmax,j,k]^.rn;
    center := 2.0*center;
    for j := 2 to jmaxm do
      center := center + n[ikmax,j,ikmax]^.rn;
    rsum := (rsum + center)*dv;
    for j := 2 to jmaxm do rsum := rsum + exvperlevel*evr[j];
    rsumnormal:=rsum/(1.0+exvol);
    rsumcheck := abs(rsumnormal-1.0);
    for i := 2 to ikmax do
      for j := 2 to jmaxm do
        for k := 2 to ikmax do
          n[i,j,k]^.rn := n[i,j,k]^.rn/rsumnormal;
    for j := 2 to jmaxm do evr[j] := evr[j]/rsumnormal;

  until rsumcheck < 1.0e-9;


{---Determine nonacoustic velocity from continuity equation}

  {.. get density per node into extra volume ..}

    for j := 2 to jmaxm do
    begin
      evmassgainperlevel[j]:=(evr[j]-evrold[j])*exvperlevel;
      evmassgainpernode[j]:=evmassgainperlevel[j]/ik;
      evdensgainpernode[j]:=evmassgainpernode[j]/dv;
    end;

    for i := 2 to ikmax do          {calculate density of each node}
      for j := 2 to jmaxm do        {using old velocity            }
        for k := 2 to ikmax do
          n[i,j,k]^.save:=  n[i,j,k]^.apo/dxtdytdzodt
             +(n[i,j-1,k]^.v*Avg(n[i,j-1,k]^.rn,n[i,j,k]^.rn)
             - n[i,j,k]^.v  *Avg(n[i,j,k]^.rn,n[i,j+1,k]^.rn))*dtody
             - evdensgainpernode[j];

  {.. level by level or 1-D approximation ..}

    for j := 2 to jmaxm2 do
    begin
      flowchange := 0.0;
      for i := 2 to ikmax do
        for k := 2 to ikmax do
          flowchange := flowchange + (n[i,j,k]^.rn-n[i,j,k]^.save);
      flowchange := flowchange/(ikmaxm*ikmaxm);
      velchange := flowchange/dtody;
      for i := 2 to ikmax do
```

```
          for k := 2 to ikmax do
          begin
            jp:=j+1;
            n[i,j,k]^.v:=n[i,j,k]^.v
                       - velchange/Avg(n[i,j,k]^.rn,n[i,jp,k]^.rn);
            n[i,j,k]^.save:=n[i,j,k]^.save + flowchange;
            n[i,jp,k]^.save:=n[i,jp,k]^.save - flowchange;
          end;
        end;

        maxrdif := 0.0;
        for i := 2 to ikmax do
          for k := 2 to ikmax do
          begin
            rdif:=abs(n[i,j,k]^.rn-n[i,j,k]^.save);
            if rdif > maxrdif then maxrdif:=rdif;
          end;

    {....Determine outer iteration difference.....}

        maxtdif := 0.0;
        for i := 2 to ikmax do
          for j := 2 to jmaxm do
            for k := 2 to ikmax do
              begin
                tdif:=abs(n[i,j,k]^.tn-n[i,j,k]^.toldi);
                if tdif > maxtdif then
                  begin maxtdif:=tdif; isv:=i; jsv:=j; ksv:=k;  end;
              end;

        if ((n[isv,jsv,ksv]^.tn   > n[isv,jsv,ksv]^.toldi)   and
           (n[isv,jsv,ksv]^.toldi < n[isv,jsv,ksv]^.toldii)) or
           ((n[isv,jsv,ksv]^.tn   < n[isv,jsv,ksv]^.toldi)   and
           (n[isv,jsv,ksv]^.toldi > n[isv,jsv,ksv]^.toldii)) and
           (sor >= 0.2) then sor := sor-0.05
           else if sor <= 1.5 then sor := sor+0.02;

        Timer;
        if nstep > oldnstep then
        begin  oldnstep:=nstep;
          wrtln(' nstep   otloop   itloop   maxtdif'+
                '         maxrdif       sor       cputime');
        end;
        writeln(nstep:5,otloop:8,itloop:9,maxtdif:13:8,
                maxrdif:14:9,sor:9:2,cputime:11:0);
      until maxtdif < sor*1.0e-6;

      Output;
    until nstep = mxstep;

    lff(outdat,3);
    writeln(outdat,'    time     pressure    (minus)wall dTdy, cl');
    writeln(outdat,'    (sec)    (lbf/ft2)              (R/in)');
    for j := 1 to pcnt do
      writeln(outdat,rts[j]:8:3,ppg[j]:11:0,-walldtdy[j]:19:0);
    close(outdat);
  end;
  {-------------------------------------------------------------------}
  procedure BoundaryConditions;
  var
    i,j,k, ic4a, ic4b, ic5a : integer;
    wall, tm, ts, a,b,c,d, ikmaxm15, xd, yd, zd, f1 : single;
    swit, swctm, swcts : array[0..6] of single;
  begin
```

```
{.. TOP HOT WALL ..}

 while rtime >= wtime[ic1+1] do ic1 := ic1+1;
 wall := Interpolate(wtime[ic1],wtime[ic1+1],
                     wtemp[ic1],wtemp[ic1+1],rtime);

 while rtime >= wttime[ic2+1] do ic2 := ic2+1;
 a:=Interpolate(wttime[ic2],wttime[ic2+1],wta[ic2],wta[ic2+1],rtime);
 b:=Interpolate(wttime[ic2],wttime[ic2+1],wtb[ic2],wtb[ic2+1],rtime);
 c:=Interpolate(wttime[ic2],wttime[ic2+1],wtc[ic2],wtc[ic2+1],rtime);
 d:=Interpolate(wttime[ic2],wttime[ic2+1],wtd[ic2],wtd[ic2+1],rtime);

 ikmaxm15:=ikmax-1.5;
 for i := 1 to ikmax do
 begin
   xd:=(ikmax-i)/ikmaxm15;
   for k := 1 to ikmax do
   begin
     zd:=(ikmax-k)/ikmaxm15;
     f1:=1.0 + a*xd*xd + b*xd + c*zd*zd + d*zd;
     if f1 > 1.0 then f1:=1.0;
     n[i,1,k]^.tn:=f1*wall;
   end;
 end;
 for i := 2 to ikmax do
   for k := 2 to ikmax do
   begin
     if n[i,1,k]^.tn<n[i-1,1,k]^.tn then n[i,1,k]^.tn:=n[i-1,1,k]^.tn;
     if n[i,1,k]^.tn<n[i,1,k-1]^.tn then n[i,1,k]^.tn:=n[i,1,k-1]^.tn;
   end;

{.. BOTTOM WALL ..}

 while rtime >= bwtime[ic3+1] do ic3 := ic3+1;
 wall := Interpolate(bwtime[ic3],bwtime[ic3+1],
                     bwtemp[ic3],bwtemp[ic3+1],rtime);
 for i := 1 to ikmax do
   for k := 1 to ikmax do
     n[i,jmax,k]^.tn := wall;

{.. CLEAR WALL ..}

 while rtime >= swctime[ic4+1] do ic4 := ic4+1;
 for j := 1 to 5 do
 begin
   swctm[j]:=Interpolate(swctime[ic4],swctime[ic4+1],
                         swctmp[j,ic4],swctmp[j,ic4+1],rtime);
   swcts[j]:=Interpolate(swctime[ic4],swctime[ic4+1],
                         swctmp[j+5,ic4],swctmp[j+5,ic4+1],rtime);
 end;
 swctm[0]:=swctm[1]; swctm[6]:=swctm[5];
 swcts[0]:=swcts[1]; swcts[6]:=swcts[5];

 ic4a := 0; ic4b := 0;
 for j := 2 to jmaxm do
 begin
   yd:=(j-1.5)/jmaxm2;
   while yd >= swcyd[ic4a+1] do ic4a:=ic4a+1;
   tm:=Interpolate(swcyd[ic4a],swcyd[ic4a+1],
                   swctm[ic4a],swctm[ic4a+1],yd);
   ts:=Interpolate(swcyd[ic4a],swcyd[ic4a+1],
                   swcts[ic4a],swcts[ic4a+1],yd);
   for i := 2 to ikmax do
```

```
    begin
      xd:=(i-1.5)/ikmaxm15;
      while xd >= swcxd[ic4b+1] do ic4b:=ic4b+1;
      wall:=Interpolate(swcxd[ic4b],swcxd[ic4b+1],ts,tm,xd);
      n[i,j,1]^.tn := wall;
    end;
  end;

 {.. INSULATED SIDE WALL ..}

  while rtime >= switime[ic5+1] do ic5 := ic5+1;
  for j := 1 to 5 do
    swit[j]:=Interpolate(switime[ic5],switime[ic5+1],
                         switmp[j,ic5],switmp[j,ic5+1],rtime);
  swit[0]:=swit[1]; swit[6]:=swit[5];

  ic5a := 0;
  for j := 2 to jmaxm do
  begin
    yd:=(j-1.5)/jmaxm2;
    while yd >= swiyd[ic5a+1] do ic5a:=ic5a+1;
    wall:=Interpolate(swiyd[ic5a],swiyd[ic5a+1],
                      swit[ic5a],swit[ic5a+1],yd);
    for k := 2 to ikmax do
      n[1,j,k]^.tn := wall;
  end;

  for j := 1 to jmax do
    n[1,j,1]^.tn := Avg(n[2,j,1]^.tn,n[1,j,2]^.tn);

end;
{---------------------------------------------------------------}
procedure TDMA; {jend,i,k : integer}
var j,jm : integer; xt : extended;
begin
  for j := 3 to jend do
  begin
    jm := j-1;
    xt := aa[j]/bb[jm];
    bb[j] := bb[j]-cc[jm]*xt;
    rhs[j] := rhs[j]-rhs[jm]*xt;
  end;
  n[i,jend,k]^.tn := rhs[jend]/bb[jend];
  for j := jend-1 downto 2 do
    n[i,j,k]^.tn := (rhs[j]-cc[j]*n[i,j+1,k]^.tn)/bb[j];
end;
{---------------------------------------------------------------}
function Powerlaw; {(peclet : extended) : extended}
var pc : extended;
begin
  if peclet > 10.0 then Powerlaw := 0.0
  else begin pc:=1.0-0.1*peclet; Powerlaw := pc*sqr(sqr(pc)); end;
end;
{---------------------------------------------------------------}
function Xk; {(t : extended) : extended}
begin  Xk := ak*sqr(t) + bk*t + ck; end;
{---------------------------------------------------------------}
function Xcv; {(t : extended) : extended}
begin  Xcv := acv*t + bcv; end;
{---------------------------------------------------------------}
procedure Grid;
var j : integer; ztest,sum : extended;
begin
  dxodt := delxz/delt;          dyodt := dely/delt;
```

```
  dzodt := delxz/delt;          dxody := delxz/dely;
  dxodz := delxz/delxz;         dyodx := dely/delxz;
  dyodz := dely/delxz;          dzodx := delxz/delxz;
  dzody := delxz/dely;          dytdz := dely*delxz;
  dxtdz := delxz*delxz;         dxtdy := delxz*dely;
  dytdzodx := dely*delxz/delxz; dv:=delxz*dely*delxz;
  dxtdzody := delxz*delxz/dely;
  dxtdyodz := delxz*dely/delxz;
  dxtdytdzodt := delxz*dely*delxz/delt;
  dxtdztdt:=delxz*delxz*delt;
  dtodx := delt/delxz;          dtody := delt/dely;
  dtodz := delt/delxz;          delyo2 := dely/2.0;
  delyt2 := dely*2.0;

  io := 1;  iout[1] := 1;
  ztest := 0.0;
  for j := 2 to jmaxm do
  begin
    if (dely*j > ztest) and (ztest < 0.0399) then
    begin
      ztest := ztest+0.005;
      io := io+1;
      iout[io] := j;
      if dely*j > 0.0399 then ztest := dely*j;
    end
    else if (dely*j > ztest) and (ztest < 0.0999) then
    begin
      ztest := ztest+0.01;
      io := io+1;
      iout[io] := j;
      if dely*j > 0.0999 then ztest := dely*j;
    end
    else if (dely*j > ztest) and (ztest < 0.3749) then
    begin
      ztest := ztest+0.025;
      io := io+1;
      iout[io] := j;
      if dely*j > 0.3749 then ztest := dely*j;
    end
    else if (dely*j > ztest) and (ztest < 1.0) then
    begin
      ztest := ztest+0.1;
      io := io+1;
      iout[io] := j;
      if dely*j > 1.0 then ztest := dely*j;
    end;
  end;
  io := io+1;  iout[io] := jmaxm;
  io := io+1;  iout[io] := jmax;
end;
{-------------------------------------------------------------}
procedure Output;
var i,j,k,m,a,c : integer;
    percent,f1,vz,tnz,tio12y1,dis,tavg : extended;
begin
  if iz1 = 1 then
  begin
    iz1 := 0;
    percent := 100.0*time/tend;
    writeln( nstep:7,'     ',percent:6:2,' %');
  end;
  iz1 := iz1+1;
  if (iz2 = 10) or (nstep = nout) then
  begin
```

```
    iz2:=0;
    pcnt:=pcnt+1;
    rts[pcnt]:=rtime;                                                {sec}
    ppg[pcnt]:=(pavg-1.0)*po;                                   {lbf/ft2}

{Calculate 'average' laser Deflection}

    a:=ikmax; c:=ikmax;
    tio12yl := ti/(12.0*yl);                                         {R/in}
    f1 := -irefrac*cavheight*laserlength*pavg*po/(roo*rgas);        {ft2-R}
    for j := 1 to jmax do yscav[j]:=0.0;
    for k := 2 to ikmax do
    begin
      dtdy[1] := tio12yl*(n[a,2,k]^.tn-n[a,1,k]^.tn)/delyo2;         {R/in}
      dtdy[2] := tio12yl*(((n[a,2,k]^.tn+n[a,3,k]^.tn)/2.0)
                 -n[a,1,k]^.tn)/dely;                                {R/in}
      for j := 3 to jmaxm2 do
        dtdy[j] := tio12yl*(n[a,j+1,k]^.tn-n[a,j-1,k]^.tn)/delyt2;  {R/in}
      dtdy[jmaxm] := tio12yl*(n[a,jmax,k]^.tn
        -((n[a,jmaxm2,k]^.tn+n[a,jmaxm,k]^.tn)/2.0))/dely;           {R/in}
      dtdy[jmax] := tio12yl*(n[a,jmax,k]^.tn-n[a,jmaxm,k]^.tn)/delyo2;
      for j := 1 to jmax do
        yscav[j] := yscav[j] + f1*dtdy[j]*144.0/sqr(ti*n[a,j,k]^.tn); {in}
    end;
    for j := 1 to jmax do yscav[j]:=yscav[j]/ikmaxm;

{Calculate centerline temperature derivative}

    dtdy[1] := tio12yl*(n[a,2,c]^.tn-n[a,1,c]^.tn)/delyo2;           {R/in}
    dtdy[2] := tio12yl*(((n[a,2,c]^.tn+n[a,3,c]^.tn)/2.0)
               -n[a,1,c]^.tn)/dely;                                  {R/in}
    for j := 3 to jmaxm2 do
      dtdy[j] := tio12yl*(n[a,j+1,c]^.tn-n[a,j-1,c]^.tn)/delyt2;    {R/in}
    dtdy[jmaxm] := tio12yl*(n[a,jmax,c]^.tn
      -((n[a,jmaxm2,c]^.tn+n[a,jmaxm,c]^.tn)/2.0))/dely;             {R/in}
    dtdy[jmax] := tio12yl*(n[a,jmax,c]^.tn-n[a,jmaxm,c]^.tn)/delyo2; {R/in}
    walldtdy[pcnt] := dtdy[1];                                       {R/in}

{Calculate center line laser Deflection}

    for j := 1 to jmax do
      ysccl[j] := f1*dtdy[j]*144.0/sqr(ti*n[a,j,c]^.tn);             {in}
  end;
  iz2:=iz2+1;

  if nstep = nout then
  begin
    ocnt2:=ocnt2+1;
    nout := nout+intout[ocnt2];

    lff(outdat,2);
    writeln(outdat,'nstep = ',nstep:6,' time=',time:10:7,'   rtime=',
            rtime:10:4,' sec    twall = ',n[a,1,c]^.tn:7:4);
    lff(outdat,1);
    writeln(outdat,'  dis      tavg     tn(i)   Ysc av   Ysc cl',
                  '     rn(i)     v(i)      dTdy      i');
    writeln(outdat,'  (in)      (R)      (R)     (in)     (in) ',
                  '     (nd)    (in/sec)   (R/in)');
    lff(outdat,1);
    for m := 1 to io do
    begin
      j := iout[m];
      vz := n[a,j,c]^.v*(alfo/yl)*12.0;
      dis:=(delyo2+((j-2)*dely))*yl*12.0;
```

```
        if j=1 then dis:=0.0; if j=jmax then dis:=yl*12.0;
        tnz:=n[a,j,c]^.tn*ti;
        tavg:=0.0;
        for k:= 2 to ikmax do
        begin
          tavg:=tavg + Avg(n[a,j,k-1]^.tn,n[a,j,k]^.tn);
          if k = 2 then tavg:=tavg/2.0;
        end;
        tavg:=tavg*ti/(ikmax-1.5);
        writeln(outdat,dis:6:3,tavg:8:0,tnz:8:0,yscav[j]:9:2,ysccl[j]:9:2,
                n[a,j,c]^.rn:10:4,vz:9:4,dtdy[j]:9:0,j:5);
      end;

      for m := 1 to 4 do
      begin
        if m=1 then j:=2  else if m=2 then j:=5
        else if m=3 then j:=jmaxm2 div 2 else if m=4 then j:=jmaxm2;
        lff(outdat,1); dis:=(delyo2+((j-2)*dely))*yl*12.0;
        writeln(outdat,'temperatures:  j=',j:3,'  dis=',dis:6:3,
                     '    rtime=',rtime:8:3,' sec');
        writeln(outdat,'k     i ->');
        for k := 1 to ikmax do
        begin
          write(outdat,k:2,'   ');
            for i := 1 to ikmax do
            begin
              tnz := n[i,j,k]^.tn*ti;
              write(outdat,tnz:7:0);
            end;
          lff(outdat,1);
        end;
      end;
      lff(outdat,1);

      for m:=1 to 4 do
      begin
        if m=1 then j:=2  else if m=2 then j:=5
        else if m=3 then j:=jmaxm2 div 2 else if m=4 then j:=jmaxm2;
        dis:=(delyo2+((j-2)*dely))*yl*12.0;
        lff(outdat,1);
        writeln(outdat,'velocities, v: j=',j:3,'  dis=',dis:6:3,
                     '    rtime=',rtime:8:3,' sec');
        writeln(outdat,'k     i ->');
        for k := 2 to ikmax do
        begin
          write(outdat,k:2);
          for i := 2 to ikmax do
            write(outdat,n[i,j,k]^.v:8:3);
          lff(outdat,1);
        end;
      end;
    end;
end;
{--------------------------------------------------------------}
procedure Timer;
var   etime : double;
begin
  Gettime(hour,minute,second,fraction);
  newtime := 60.0*minute + second + 0.01*fraction;
  if newtime > oldtime then etime := newtime-oldtime
    else etime := newtime-oldtime+3600.0;
  oldtime:=newtime;
  cputime:=cputime+etime;
end;
```

```
{-------------------------------------------------------------------}
procedure Init;
var outname,inname : string[13]; runnumber : string[3];
  i,j,k,m : integer; bw : single;
begin
  clrscr; gotoxy(10,3);
  wrt('Enter three digit run number (ex. 007 or 124) --> ');
  runnumber:=GetStr(3,-1,-1,''); inname := 'NTACC'+runnumber+'.IN';
  outname := 'NTACC'+runnumber+'.DAT';
  OpenFile(indat,inname); if error <> 0 then Stop;
  CreateFile(outdat,outname);
  writeln(outdat,'         NTACC'+runnumber+'.DAT');

  Readcomments(indat);
  readln(indat,xl,yl,zl);
  Readcomments(indat);
  readln(indat,rdelt,rtend,sor);
  delt:=rdelt*alfo/(yl*yl);  tend:=rtend*alfo/(yl*yl);
  Readcomments(indat);
  readln(indat,jmax,ikmax);  ik:=ikmax*ikmax;
  dely:=1.0/jmax; jmax := jmax+2;  jmaxm := jmax-1;  jmaxm2 := jmax-2;
  delxz:=1.0/ikmax;
  ikmax:=((ikmax+1) div 2)+1; ikmaxm:=ikmax-1;  ikmaxp:=ikmax+1;
  if (jmax > ynodemax) or (ikmaxp > xznodemax) then begin lfs(2);
    wrtln(' *** Too many nodes ***'); stop; end;
  Readcomments(indat);
  ocnt:=0; routint[0]:=0.0;
  repeat
    ocnt:=ocnt+1;
    readln(indat,routint[ocnt]);
  until routint[ocnt] < 0;
  ocnt:=ocnt-1;
  for m := 1 to ocnt do
  begin
    outint[m]:=(routint[m]-routint[m-1])*alfo/(yl*yl);
    intout[m] := round(outint[m]/delt);
  end;
  Readcomments(indat); readln(indat,exvol); exvol:=exvol-1.0;
  exvperlevel:=exvol/jmaxm2;
  for j:=1 to jmax do begin evr[j]:=1.0; evt[j]:=1.0; end;
  writeln(outdat,' exvol = ',exvol:6:3);

  Readcomments(indat);  i:=1;
  readln(indat,wtime[1],wtemp[1]);
  ti:=wtemp[1]; wtemp[1]:=1.0;
  repeat
    i := i+1;
    readln(indat,wtime[i],wtemp[i]);
    wtemp[i]:=wtemp[i]/ti;
  until wtime[i] < 0;
  wtime[i]:=9.9e9; wtemp[i]:=wtemp[i-1];

  Readcomments(indat);  i:=0;
  repeat
    i:=i+1;
    readln(indat,wttime[i],wta[i],wtb[i],wtc[i],wtd[i]);
  until wttime[i] < 0;
  wttime[i]:=9.9e9; wta[i]:=wta[i-1]; wtb[i]:=wtb[i-1];
  wtc[i]:=wtc[i-1]; wtd[i]:=wtd[i-1];

  Readcomments(indat); i:=1;
  readln(indat,bwtime[1],bwtemp[1]);
  bw:=bwtemp[1]; bwtemp[1]:=1.0;
  repeat
```

```
    i:=i+1;
    readln( indat,bwtime[i],bwtemp[i] );
    bwtemp[i]:=bwtemp[i]/bw;
  until bwtime[i] < 0;
  bwtime[i]:=9.9e9; bwtemp[i]:=bwtemp[i-1];

  Readcomments( indat); j:=0;
  repeat
    j:=j+1;
    readln( indat,swctime[j],swctmp[1,j],swctmp[2,j],swctmp[3,j],
                  swctmp[4,j],swctmp[5,j],swctmp[6,j],swctmp[7,j],
                  swctmp[8,j],swctmp[9,j] );
    swctmp[10,j]:=swctmp[5,j];
  until swctime[j] < 0;
  swctime[j]:=9.9e9;
  for i := 1 to 10 do swctmp[i,j]:=swctmp[i,j-1];
  swcyd[0]:=0.0; swcyd[1]:=0.0475; swcyd[2]:=0.14; swcyd[3]:=0.2625;
  swcyd[4]:=0.515; swcyd[5]:=0.76; swcyd[6]:=1.0001;
  swcxd[1]:=0.0; swcxd[2]:=0.075; swcxd[3]:=1.0001;

  Readcomments( indat); j:=0;
  repeat
    j:=j+1;
    readln( indat,switime[j],switmp[1,j],switmp[2,j],switmp[3,j],
                  switmp[4,j],switmp[5,j] );
  until switime[j] < 0;
  switime[j]:=9.9e9;
  for i := 1 to 5 do switmp[i,j]:=switmp[i,j-1];
  swiyd[0]:=0.0; swiyd[1]:=0.075; swiyd[2]:=0.1825; swiyd[3]:=0.295;
  swiyd[4]:=0.55; swiyd[5]:=0.775; swiyd[6]:=1.0001;

  for i := 1 to ikmax do
    for j := 1 to jmax do
      for k := 1 to ikmax do
        new( n[i,j,k] );
  for j := 2 to jmaxm do
    for k := 2 to ikmax do                     {sets adiabatic bc}
      n[ikmaxp,j,k] := n[ikmaxm,j,k];
  for i := 2 to ikmax do
    for j := 2 to jmaxm do                     {sets adiabatic bc}
      n[i,j,ikmaxp] := n[i,j,ikmaxm];

  for i:=1 to ynodemax do begin aa[i]:=0.0; bb[i]:=0.0;
    cc[i]:=0.0; rhs[i]:=0.0; end;
  for i := 1 to ikmax do
    for j := 1 to jmax do
      for k := 1 to ikmax do
      begin
        n[i,j,k]^.rn:= 1.0;   n[i,j,k]^.apo:=dxtdytdzodt;
        n[i,j,k]^.tn:= 1.0;   n[i,j,k]^.told:=1.0;      {Initial Conditions}
        n[i,j,k]^.toldi:=1.0; n[i,j,k]^.toldii:=1.0; n[i,j,k]^.v := 0.0;
      end;
  mxstep := round( tend/delt)+1;
  nout := intout[1]; ocnt2:=1;
  ic1:=0; ic2:=0; ic3:=0; ic4:=0; ic5:=0; iz1:=0; iz2:=0;
  ppg[1]:=0.0; rts[1]:=0.0; walldtdy[1]:=0.0; pcnt:=1; pavg:=1.0;
  nstep := -1;  oldnstep:=-1;
  cputime:=0.0;  Gettime( hour,minute,second,fraction);
  oldtime:=60.0*minute+second+0.01*fraction;
  clrscr;
end;
{---------------------------------------------------------------}
begin Go end.
```

```
{ ***************************************************************** }
{   UTILITY.PAS                                                     }
{                                                                   }
{  This unit contains some routines that are used in COND5C.PAS     }
{  and TACPROC4.PAS                                                 }
{                                                                   }
{ ***************************************************************** }

unit Utility;
interface

uses crt,graph;

type
  string12 = string[12];  string20 = string[20];  string80 = string[80];
  buffer = array[1..2048] of char;

var
  comment : string80;  ch : char;  error : integer; func10,escape : boolean;

function ArcCos(x : extended) : extended;
function ArcSin(x : extended) : extended;
function Avg(x1,x2 : extended) : extended;
procedure Beep;
procedure CreateFile(var ffile : text; filename : string80);
function GetStr (maxlen,txtcolor,backcolor : shortint;
                 defstr : string80) : string80;
procedure GraphErrorproc(s : string20);
function Interpolate(x1,x2,y1,y2,xmid : extended) : extended;
procedure lff(var filename : text; n : byte);
procedure lfs(n : byte);
procedure OpenFile(var ffile : text; filename : string80);
procedure OpenFileBuf(var ffile:text; var buf:buffer; filename:string80);
function Power(Mantissa,Exponent : extended) : extended;
procedure PressaKey;
procedure Readcomments(var filename : text);
procedure Stop;
procedure Stopg;
procedure wrtln(s : string80);
procedure wrt(s : string80);
procedure WrtStr(x,y : integer; s : string80; txtcolor,backcolor : shortint);

implementation

{------------------------------------------------------------------}
function ArcCos(x : extended) : extended;
begin ArcCos:=(3.14159265/2.0)-ArcTan(x/(sqrt(1.0-x*x))) end;
{------------------------------------------------------------------}
function ArcSin(x : extended) : extended;
begin ArcSin:=ArcTan(x/(sqrt(1.0-x*x))) end;
{------------------------------------------------------------------}
function Avg(x1,x2 : extended) : extended;
begin Avg:=(x1+x2)/2.0; end;
{------------------------------------------------------------------}
procedure Beep; begin sound(350); delay(50); nosound; end;
{------------------------------------------------------------------}
procedure CreateFile(var ffile : text; filename : string80);
begin
  assign(ffile,filename); {$i-} rewrite(ffile); {$i+}  error:=IOResult;
  if error <> 0 then begin lfs(1);
    writeln('Could not create ',filename,'  Error: ',error:3); end;
end;
{------------------------------------------------------------------}
function GetStr(maxlen,txtcolor,backcolor : shortint;
```

```
                defstr : string80) : string80;
var
  printables : set of char;  s,clearit : string80;
  x,y : integer;  oldbcolor,oldtcolor : byte;  cr,fc : boolean;

begin  func10:=false; escape:=false;
  oldbcolor:=(textattr and 112) shr 4;  oldtcolor:=textattr and 15;
  fc:=true; cr:=false;  printables := [' '..'~']; x:=wherex; y:=wherey;
  if length(defstr) > maxlen then defstr[0]:=chr(maxlen);
  fillchar(clearit,81,' '); clearit[0]:=chr(maxlen);
  if txtcolor > -1 then textcolor(txtcolor);
  if backcolor > -1 then textbackground(backcolor);
  wrt(clearit); gotoxy(x,y); wrt(defstr); s:=defstr;
  repeat
    repeat until keypressed;    ch:=readkey;
    if ch in printables then
    begin
      if fc then
        begin s:=''; gotoxy(x,y); wrt(clearit); gotoxy(x,y);  end;
      if length(s) < maxlen then begin s:=s+ch; wrt(ch); end  else Beep;
      fc:=false;
    end
    else
      case ord(ch) of
        0 : begin ch:=readkey;
                  if (ch='D') and (length(s)>0) then
                    begin cr:=true; func10:=true; end
                  else Beep; end;
        8 : begin if length(s) > 0 then
                    begin fc:=false; delete(s,length(s),1);
                      gotoxy(wherex-1,wherey); wrt(' ');
                      gotoxy(wherex-1,wherey); end
                    else Beep;
            end;
        13 : if length(s) > 0 then cr:=true else Beep;
        27 : begin escape:=true; cr:=true; s:=defstr; end;
      else Beep;
      end;
  until cr;
  textbackground(oldbcolor); textcolor(oldtcolor); GetStr:=s;
end;
{--------------------------------------------------------------------}
procedure GraphErrorproc(s : string20);
begin  error := graphresult;
       if error <> 0 then begin
         restorecrtmode; writeln(s,' ',grapherrormsg(error)); Stop; end;
end;
{--------------------------------------------------------------------}
function Interpolate(x1,x2,y1,y2,xmid : extended) : extended;
begin  Interpolate := y1 + ((xmid-x1)/(x2-x1))*(y2-y1); end;
{--------------------------------------------------------------------}
procedure lff(var filename : text; n : byte);
var i : byte;
begin    for i := 1 to n do writeln(filename);  end;
{--------------------------------------------------------------------}
procedure lfs(n : byte);
var i : byte;
begin    for i := 1 to n do writeln;  end;
{--------------------------------------------------------------------}
procedure OpenFile(var ffile : text; filename : string80);
begin
  assign(ffile,filename); {$i-} reset(ffile); {$i+} error:=IOResult;
  if error <> 0 then begin lfs(1);
    writeln('Could not open ',filename,'  Error: ',error:3); end;
```

```
end;
{------------------------------------------------------------------------}
procedure OpenFileBuf(var ffile:text; var buf:buffer; filename:string80);
begin
  assign(ffile,filename); settextbuf(ffile,buf);{$i-} reset(ffile); {$i+}
  error:=IOResult;
  if error <> 0 then begin lfs(1);
    writeln('Could not open ',filename,'  Error: ',error:3); end;
end;
{------------------------------------------------------------------------}
function Power(Mantissa,Exponent : extended) : extended;
begin Power := exp(ln(mantissa)*exponent); end;
{------------------------------------------------------------------------}
procedure PressaKey;
begin   write('** Press any key to continue **');
        repeat until keypressed; ch:=readkey;      end;
{------------------------------------------------------------------------}
procedure Readcomments(var filename : text);
begin
  repeat
    readln(filename,comment);
    if length(comment) = 0 then comment := ' ';
  until comment[1] = '*';
end;
{------------------------------------------------------------------------}
procedure Stop;
begin  lfs(1); wrt(' ** Press a key to Exit ** '); ch:=readkey; halt; end;
{------------------------------------------------------------------------}
procedure Stopg;
begin    closegraph; lfs(1);
  wrt(' ** Press a key to Exit ** '); ch:=readkey; halt; end;
{------------------------------------------------------------------------}
procedure wrtln(s : string80);
begin writeln(s); end;
{------------------------------------------------------------------------}
procedure wrt(s : string80);
begin write(s); end;
{------------------------------------------------------------------------}
procedure WrtStr(x,y : integer; s : string80; txtcolor,backcolor : shortint);
var oldtcolor,oldbcolor : byte;
begin
  oldbcolor:=(textattr and 112) shr 4; oldtcolor:=textattr and 15;
  if txtcolor > -1 then textcolor(txtcolor);
  if backcolor > -1 then textbackground(backcolor);
  gotoxy(x,y); wrt(s);
  textbackground(oldbcolor); textcolor(oldtcolor);
end;
{------------------------------------------------------------------------}
end.
```

```
$nofloatcalls
      program tac2
cc
cc ************************************************************************
cc TAC2.FOR   (complied with Microsoft FORTRAN77 ver 3.31
cc
cc    This is the ThermoAcoustic Convection code originally written by
cc    Adel Salah-Eddine for his Master's Thesis.  It is a reproduction of
cc    the numerical scheme used by Spradley, et al.  This version (TAC2)
cc    includes the following deviations from the original:
cc        1) Conversion to IBM PC.
cc        2) "Clean-up" for better efficiency
cc        3) Power law for variable viscosity (instead of curve fit)
cc        4) Changes in number of nodes and time step
cc
cc ************************************************************************
cc
cc  ** Definition of primary variables **
cc
cc u : velocity,  r : density,  t : temperature,  p : pressure
cc ru : r*u,  ruu : r*u*u,  dru : d(ru)/dx,  druu : d(ruu)/dx
cc dt : d(t)/dx,  du : d(u)/dx,  dp : d(p)/dx
cc d2u and d2t : second derivative wrt x,
cc xl : cavity length (ft),  to : initial temperature (R)
cc ro : initial density (lbm/ft3),  v : initial viscosity (lbm/(ft-sec))
cc cv : constant volume heat capacity ((ft-lbf)/(lbm-R))
cc xk : thermal conductivity ((ft-lbf)/(ft-sec-R))
cc gam : heat capacity ratio, rb : gas constant ((ft-lbf)/(lbm-R))
cc
cc ****************************************************************************
cc
      implicit real * 8 (a-h,o-z)
      real * 8 u(220,2),r(220,2),t(220,2),p(220,2),druu(220),d2u(220),
     )       ruu(220),ru(220),dp(220),dru(220),dt(220),du(220),d2t(220)
cc
cc----Coefficients of the heater surface temp. curve fit equation
cc
ccw      data   aa,       bb,       cc,       dd
ccw     )     /533.0,  1106.3,  -3469.71,   19630.0/
ccw      data   ee,       ff,        gg,        hh
ccw     )     /-50652.3, 69893.7,  -55832.9, 25944.8/
ccw      data   xii,      xjj
ccw     )     /-6520.7,  686.102/
      data   aa,       bb,       cc,       dd
     )     /1066.0,    0.0,      0.0,      0.0/
      data   ee,       ff,        gg,        hh
     )     /0.0,       0.0,       0.0,       0.0/
      data   xii,      xjj
     )      / 0.0,      0.0/
cc
cc----Geometrical data and initial conditions - Air initial properties
cc
      data  xl,      to
     )     /0.33333, 533.0/
cc
      data   ro,      v,           cv,     xk,     gam,  rb
     )     /0.0757, 1.2282e-5,  133.4, 0.00318, 1.4, 53.34/
cc
      open(6,file='tac2.dat')
      write(*,200)
 200  format(/,1x,'Enter the number of nodes (maximum of 220)')
      read(*,*) imax
      write(*,210)
 210  format(/,1x,'Enter the time step')
```

```
      read(*,*) delt
      write(*,220)
 220  format(/,1x,'Enter real end time (sec)')
      read(*,*) rtend
      k1out=(int(5.0/delt)+1)*0
      k2out=int(0.00334/delt)+1
cc
cc----Calculating Re and Pr
cc
      sqrto=sqrt(rb*32.2*to)
      re=(ro*xl*sqrto)/v
      pr=(v*gam*cv)/xk
cc
cc----Number of node points and grid spacing
cc
      imaxm=imax-1
      imx2=imax/2
      imx2p1=imx2+1
      iwrt5=imax/5
      iwrt10=imax/10
      iwrt15=iwrt5+iwrt10
      iwrt30=imax/30
      if(iwrt30 .lt. 1) iwrt30=1
      delx=1.0/imax
      rdelt=delt*xl/sqrto
      intout=0.1/rdelt
      mxstep=(rtend+0.0002)/rdelt
      nout=intout
      nstep=0
cc
cc----miscellaneous constants
cc
      delxsq=delx*delx
      delx2=2.0*delx
      delxo2=delx/2.0
      gorepr=gam/(re*pr)
      gm1=gam-1.0
      re43=4.0/(3.0*re)
      f4o3=4.0/3.0
cc
cc----Initial conditions
cc
      do 1 i=1,imax
        u(i,1)=0.0
        t(i,1)=1.0
        p(i,1)=1.0
        r(i,1)=1.0
 1    continue
cc
 1000 nstep=nstep+1
      time=(nstep-1)*delt
      rtime=(time*xl)/sqrto
cc
cc----Calculate the heater surface temperature
cc
      twall=aa+bb*rtime+cc*(rtime*rtime)+dd*(rtime*rtime*rtime)
     >   +ee*(rtime**4)+ff*(rtime**5)+gg*(rtime**6)
     >   +hh*(rtime**7)+xii*(rtime**8)+xjj*(rtime**9)
      tw=twall/to
ccw      twall=2.0*to
ccw      tw=twall/to
cc
cc----Write statement
cc
```

```
      iz1=iz1+1
      if (iz1 .lt. 500) go to 110
      iz1=0
      percnt=100.0*(rtime/rtend)
      write(*,100) nstep,percnt
 100  format(1x,i9,5x,f9.4,'%')
 110  continue
      if (nstep .gt. k1out) go to 130
      iz2=iz2+1
      if (iz2 .lt. k2out) go to 130
      iz2=0
      uav=(u(imx2,1)+u(imx2p1,1))/2.0
      write(6,120) time,uav
 120  format(1x,f6.3,1x,f7.4)
 130  continue
      if(nstep .lt. nout) go to 3
      nout=nout+intout
      write(6,4) nstep,time,rtime,twall
 4    format(//,1x,'nstep =',i6,4x,'time=',f8.2,4x,'rtime=',f9.6,
     >        ' sec',4x,'twall =',1x,f8.2,/)
      write(6,21)
 21   format(2x,'i',5x,'dis',7x,'u(i,1)',5x,'p(i,1)',5x,
     >        'r(i,1)',5x,'t(i,1)',5x,'t (R)',/)
      do 60 i=1,imax,iwrt30
        trr=t(i,1)*to
        dis=(delxo2+((i-1)*delx))
        write(6,61) i,dis,u(i,1),p(i,1),r(i,1),t(i,1),trr
 61     format(1x,i3,3x,f5.3,5x,f7.5,3(5x,f6.4),4x,f6.0)
 60   continue
 3    continue
cc
cc----solve the momentum equation
cc
      ruu(1)=r(1,1)*u(1,1)*u(1,1)
      ruu(2)=r(2,1)*u(2,1)*u(2,1)
      druu(1)=(ruu(1)+ruu(2))/delx2
      dp(1)=(p(2,1)-p(1,1))/delx
      d2u(1)=(u(2,1)-3.0*u(1,1))/delxsq
        do 7 i=2,imaxm
          im=i-1
          ip=i+1
          ruu(ip)=r(ip,1)*u(ip,1)*u(ip,1)
          druu(i)=(ruu(i)-ruu(im))/delx
          if(u(i,1) .lt. 0.0) druu(i)=(ruu(ip)-ruu(i))/delx
          dp(i)=(p(ip,1)-p(im,1))/delx2
          d2u(i)=(u(ip,1)-2.*u(i,1)+u(im,1))/delxsq
 7      continue
      druu(imax)=(-1.*(ruu(imax)+ruu(imaxm)))/delx2
      dp(imax)=(p(imax,1)-p(imaxm,1))/delx
      d2u(imax)=(u(imaxm,1)-3.0*u(imax,1))/delxsq
cc
cc----Calculate r*u at the new time level
cc
        do 8 i=1,imax
          vcorr=v*(t(i,1)**0.666)
          ru(i)=r(i,1)*u(i,1)-delt*(druu(i)+dp(i)-re43*vcorr*d2u(i))
 8      continue
cc
cc----Solve the continuity equation
cc
      dru(1)=(ru(2)+ru(1))/delx2
        do 9 i=2,imaxm
          dru(i)=(ru(i+1)-ru(i-1))/delx2
 9      continue
```

```
      dru(imax)=-1.0*(ru(imax)+ru(imaxm))/delx2
cc
        do 10 i=1,imax
          r(i,2)=r(i,1)-delt*(dru(i))
          u(i,2)=ru(i)/r(i,2)
 10     continue
cc
cc----Solve the energy equation
cc
      dt(1)=(t(1,1)-tw)/delxo2
      if (u(1,1) .lt. 0.0) dt(1)=(t(2,1)-t(1,1))/delx
      du(1)=(u(1,2)+u(2,2))/delx2
      d2t(1)=f4o3*(t(2,1)-3.0*t(1,1)+2.0*tw)/delxsq
        do 11 i=2,imaxm
          im=i-1
          ip=i+1
          dt(i)=(t(i,1)-t(im,1))/delx
          if(u(i,1) .lt. 0.0) dt(i)=(t(ip,1)-t(i,1))/delx
          du(i)=(u(ip,2)-u(im,2))/delx2
          d2t(i)=(t(ip,1)-2.0*t(i,1)+t(im,1))/delxsq
 11     continue
      dt(imax)=(t(imax,1)-t(imaxm,1))/delx
      if(u(imax,1) .lt. 0.0) dt(imax)=(1.0-t(imax,1))/delxo2
      du(imax)=-1.0*(u(imax,2)+u(imaxm,2))/delx2
      d2t(imax)=f4o3*(t(imaxm,1)-3.0*t(imax,1)+2.0)/delxsq
cc
cc----Calculate the new temperature
cc
        do 12 i=1,imax
          term2=(gm1*p(i,1)*du(i))/r(i,2)
          term3=gorepr*d2t(i)/r(i,2)
          t(i,2)=t(i,1)-delt*(u(i,2)*dt(i)+term2-term3)
 12     continue
cc
cc-----Calculate the new pressure
cc
        do 13 i=1,imax
          p(i,2)=r(i,2)*t(i,2)
 13     continue
cc
      do 16 i=1,imax
        u(i,1)=u(i,2)
        p(i,1)=p(i,2)
        r(i,1)=r(i,2)
        t(i,1)=t(i,2)
 16   continue
      if(nstep .lt. mxstep) go to 1000
      stop
      end
```

```
$nofloatcalls
      program tac2g
cc ********************************************************************
cc  TAC2G.FOR  (Complied with Microsoft FORTRAN77 ver 3.31)
cc
cc    This was originally the ThermoAcoustic Convection code written by
cc    Adel Salah-Eddine for his Master's Thesis.  This version (TAC2g)
cc    includes the following deviations from the original:
cc              1) Conversion to IBM PC.
cc              2) "Clean-up" for better efficiency
cc              3) Power law for variable viscosity (instead of curve fit)
cc from TAC1    1) Changes in number of nodes and time step
cc from TAC2    1) Central differencing when u changes sign
cc from TAC2a   1) Variable grid size
cc from tac2b   1) 2nd order accurate derivatives near the wall
cc from tac2c   1) rearrangement of continuity and momentum equations
cc from tac2d   1) implicit method
cc from tac2e   1) constant grid size
cc from tac2f   1) conservative form of equations
cc ********************************************************************
cc
cc  ** Definition of primary variables **
cc
cc un,uo,uold : new,old velocity;  rn,ro : new,old density;
cc tn,to : new,old temperature;  p : pressure
cc run,ruo : new,old r*u
cc aa,bb,cc,rhs : array for tri-diagonal matrix solver
cc xl : cavity length (ft);  ti : initial temperature (R)
cc roo : initial density (lbm/ft3);  v : initial viscosity (lbm/(ft-sec))
cc cv : constant volume heat capacity ((ft-lbf)/(lbm-R))
cc xk : thermal conductivity ((ft-lbf)/(ft-sec-R))
cc rb : gas constant ((ft-lbf)/(lbm-R))
cc gam specific heat ratio;  re : Reynold's number
cc pr : Prandtl number;  alf : thermal diffusivity
cc
cc ***********************************************************************
cc
      implicit real * 8 (a-h,o-z)
      dimension un(220), uo(220), to(220), tn(220), rn(220), ro(220),
     >          p(220), ruo(220), run(220), unold(220)
      dimension aa(220),bb(220),cc(220),rhs(220)
      common /cons/ imax,imaxm,imaxm2,delt,dtdx,dt2dx,dt3dx,tdtdx,
     >              hdtdx,fdt3dx,dtdx2,tdtdx2,edtdx2,fdtdx2,gdtdx2

cc  output parameters
      common /outpt/ dis(220), iout(50), iu, io
cc
cc----Coefficients of the heater surface temp. curve fit equation
cc
ccw      data   aa,     bb,      cc,       dd,       ee,      ff
ccw     >     /126.5, -779.69, 1581.43, -880.48, -930.85, 1914.03/
cc
cc----Geometrical data and initial conditions - Air initial properties
cc
      data  xl,   ti
     >    /1.0, 533.0/
      data   roo,       v,          cv,      xk,      rb
     >    /0.0757, 1.2282e-5,   133.4, 0.00318,  53.34/
      open(6,file='tac2g.dat')
      write(*,40)
 40   format(/,1x,'Enter the time step')
      read(*,*) delt
      write(*,60)
 60   format(/,1x,'Enter real end time (sec)')
```

```
      read(*,*) rtend
      write(*,70)
 70   format(/,1x,'Enter output interval [real time] (sec)')
      read(*,*) outint
      rtend=rtend+0.001
      k1out=int(5.0/delt)+1
      k2out=int(0.00499/delt)+1
cc
cc----Miscellaneous constants
cc
      gam=1.4
      gm1=gam-1.0
      sqrto=sqrt(rb*32.2*ti)
      re=(roo*xl*sqrto)/v
      pr=(v*gam*cv)/xk
      re43=4.0/(3.0*re)
      gorepr=gam/(re*pr)
      alf=xk/(roo*cv)
cc
      call grid
      write(*,75)
 75   format(/,1x,'Enter velocity relaxation factor')
      read(*,*) a
      am=1.0-a
      uconv=a*0.01
      write(*,77)
 77   format(/,1x,'Print Udifmax?? [0,1]')
      read(*,*) iup
cc
      rdelt=delt*xl/sqrto
      intout=outint/rdelt
      mxstep=rtend/rdelt
      nout=intout
      nstep=0
cc
cc----Initial conditions
cc
      do 80 i=1,imax
        uo(i)=0.0
        to(i)=1.0
        ro(i)=1.0
        p(i)=1.0
 80   continue
      tw=2.0
cc
      write(*,90)
 90   format(//,25x,'BEGIN  TAC',/)
 1000 nstep=nstep+1
      time=(nstep-1)*delt
      rtime=(time*xl)/sqrto
cc
cc----Calculate the heater surface temperature
cc
ccw     twall=aa*(rtime**6)+bb*(rtime**5)+cc*(rtime**4)+dd*(rtime**3)
ccw    )   +ee*(rtime**2)+ff*rtime+ti-460.0
ccw     twall=twall+460.0
      twall=2.0*ti
cc
cc----Write statement
cc
      iz1=iz1+1
      if (iz1 .lt. 25) go to 110
      iz1=0
      percent=100.0*rtime/rtend
```

```
      write(*,100) nstep, percent
100   format(1x,i7,5x,f6.2,' %')
110   continue
      if (nstep .gt. k1out) go to 130
      iz2=iz2+1
      if (iz2 .lt. k2out) go to 130
      iz2=0
      uz1=uo(1)*1.0e3
      uz2=uo(iu)*1.0e3
      write(6,120) time, uz1, uz2
120   format(1x,f6.3,2(1x,f9.3))
130   continue
      if(nstep .lt. nout) go to 199
      nout=nout+intout
      write(6,135) nstep,time,rtime,twall
135   format(//,1x,'nstep =',i6,4x,'time=',f8.2,4x,'rtime=',f9.6,
     >       ' sec',4x,'twall =',1x,f8.2,/)
      write(6,140)
140   format(3x,'dis',7x,'t(i)',8x,'tc',7x,'u(i)e3',7x,'p(i)',
     >        7x,'r(i)',7x,'i',/)
      do 150 iz=1,io
        i=iout(iz)
        uz=uo(i)*1.0e3
        call conduc(rtime,dis(i),alf,tc)
        write(6,145) dis(i),to(i),tc,uz,p(i),ro(i),i
145     format(1x,2(f6.4,5x),f6.4,3x,f9.3,5x,2(f6.4,5x),i3)
150   continue
199   continue
      icloop=0
500   continue
      icloop=icloop+1
      do 510 i=1,imax
510     unold(i)=un(i)
cc
cc----solve the momentum equation for run(i)
cc
      if(un(1)*un(2) .lt. 0.0) then
        bb(1)= 1.0 + dtdx*un(1)
        cc(1)= dt3dx*un(2)
      else
        if(un(1) .gt. 0.0) then
          bb(1)= 1.0 + tdtdx*un(1)
          cc(1)= 0.0
        else
          bb(1)= 1.0 - dtdx*un(1)
          cc(1)= dtdx*un(2)
        endif
      endif
      vcre=(tn(1)**0.666)*re43
      rhs(1)= ruo(1) + (dt2dx*p(3) - tdtdx*p(2) + hdtdx*p(1))
     >               + vcre*(gdtdx2*un(2) - fdtdx2*un(1))
        do 9 i=2,imaxm
          im=i-1
          ip=i+1
          if(un(im)*un(ip) .lt. 0.0) then
            aa(i)= - dt2dx*un(im)
            bb(i)= 1.0
            cc(i)= dt2dx*un(ip)
          else
            if(un(i) .gt. 0.0) then
              aa(i)= - dtdx*un(im)
              bb(i)= 1.0 + dtdx*un(i)
              cc(i)= 0.0
            else
```

```
            aa(i)=  0.0
            bb(i)= 1.0 - dtdx*un(i)
            cc(i)= dtdx*un(ip)
          endif
        endif
        vcre=(tn(i)**0.666)*re43
        rhs(i)= ruo(i) - dt2dx*(p(ip) - p(im))
     >          + vcre*(dtdx2*un(ip) - tdtdx2*un(i) + dtdx2*un(im))
 9      continue
      if(un(imaxm)*un(imax) .lt. 0.0) then
        aa(imax)= - dt3dx*un(imaxm)
        bb(imax)= 1.0 - dtdx*un(imax)
      else
        if(un(imax) .gt. 0.0) then
          aa(imax)= - dtdx*un(imaxm)
          bb(imax)= 1.0 + dtdx*un(imax)
        else
          aa(imax)=  0.0
          bb(imax)= 1.0 - tdtdx*un(imax)
        endif
      endif
      vcre=(tn(imax)**0.666)*re43
      rhs(imax)= ruo(imax) - (hdtdx*p(imax) - tdtdx*p(imaxm)
     >                        + dt2dx*p(imaxm2))
     >          + vcre*(-fdtdx2*un(imax) + gdtdx2*un(imaxm))
      call tdma(aa,bb,cc,rhs,run)
      do 520 i=1,imax
 520    un(i)=a*un(i)+am*unold(i)
cc
cc----Solve the continuity equation for rn(i)
cc
      rn(1)= ro(1) - dt3dx*run(2) - dtdx*run(1)
        do 7 i=2,imaxm
          rn(i)= ro(i) - dt2dx*(run(i+1)-run(i-1))
 7      continue
      rn(imax)= ro(imax) + dtdx*run(imax) + dt3dx*run(imaxm)
cc
      do 8 i=1,imax
        un(i)=run(i)/rn(i)
        un(i)=a*un(i)+am*unold(i)
 8    continue
cc
cc----Solve the energy equation for tn(i)
cc
      gopern=gorepr/rn(1)
      if(un(1)*un(2) .lt. 0.0) then
        bb(1)= 1.0 + dtdx*un(1) + gopern*fdtdx2
        cc(1)= dt3dx*un(1) - gopern*gdtdx2
        rhs(1)= to(1) + un(1)*fdt3dx*tw + gopern*edtdx2*tw
     >          - gm1*p(1)*(dt3dx*un(2) + dtdx*un(1))/rn(1)
      else
        if(un(1) .gt. 0.0) then
          bb(1)= 1.0 + tdtdx*un(1) + gopern*fdtdx2
          cc(1)= - gopern*gdtdx2
          rhs(1)= to(1) + un(1)*tdtdx*tw + gopern*edtdx2*tw
     >          - gm1*p(1)*(dt3dx*un(2) + dtdx*un(1))/rn(1)
        else
          bb(1)= 1.0 - dtdx*un(1) + gopern*fdtdx2
          cc(1)= dtdx*un(1) - gopern*gdtdx2
          rhs(1)= to(1) + gopern*edtdx2*tw
     >          - gm1*p(1)*(dt3dx*un(2) + dtdx*un(1))/rn(1)
        endif
      endif
        do 11 i=2,imaxm
```

```
          im=i-1
          ip=i+1
          gopern=gorepr/rn(i)
          if(un(im)*un(ip) .lt. 0.0) then
            aa(i)= - dt2dx*un(i) - gopern*dtdx2
            bb(i)= 1.0 + gopern*tdtdx2
            cc(i)= dt2dx*un(i) - gopern*dtdx2
          else
            if(un(i) .gt. 0.0) then
              aa(i)= - dtdx*un(i) - gopern*dtdx2
              bb(i)= 1.0 + dtdx*un(i) + gopern*tdtdx2
              cc(i)= - gopern*dtdx2
            else
              aa(i)=  - gopern*dtdx2
              bb(i)= 1.0 - dtdx*un(i) + gopern*tdtdx2
              cc(i)= dtdx*un(i) - gopern*dtdx2
            endif
          endif
          rhs(i)= to(i) - gm1*p(i)*dt2dx*(un(ip) - un(im))/rn(i)
 11     continue
        gopern=gorepr/rn(imax)
        if(un(imaxm)*un(imax) .lt. 0.0) then
          aa(imax)= - dt3dx*un(imax) - gopern*gdtdx2
          bb(imax)= 1.0 - dtdx*un(imax) + gopern*fdtdx2
          rhs(imax)= to(imax) - fdt3dx*un(imax) + gopern*edtdx2
     >           + gm1*p(imax)*(dtdx*un(imax)
     >                        + dt3dx*un(imaxm))/rn(imax)
        else
          if(un(imax) .gt. 0.0) then
            aa(imax)= - dtdx*un(imax) - gopern*gdtdx2
            bb(imax)= 1.0 + dtdx*un(imax) + gopern*fdtdx2
            rhs(imax)= to(imax) + gopern*edtdx2
     >           + gm1*p(imax)*(-dtdx*un(imax)
     >                        + dt3dx*un(imaxm))/rn(imax)
          else
            aa(imax)=  - gopern*gdtdx2
            bb(imax)= 1.0 - tdtdx*un(imax) + gopern*fdtdx2
            rhs(imax)= to(imax) - tdtdx*un(imax) + gopern*edtdx2
     >           + gm1*p(imax)*(-dtdx*un(imax)
     >                        + dt3dx*un(imaxm))/rn(imax)
          endif
        endif
        call tdma(aa,bb,cc,rhs,tn)
cc
cc-----Calculate the new pressure
cc
          do 13 i=1,imax
            p(i)=rn(i)*tn(i)
 13       continue
cc
cc-----Check for time step convergence
cc
        if(icloop .eq. 1) go to 500
        udifmx=0.0
          do 15 i=1,imax
            if(dabs(unold(i)) .gt. 1.0e-5) then
              udif=dabs((un(i)-unold(i))/unold(i))
              if(udif .gt. udifmx) udifmx=udif
            endif
 15       continue
        if(iup .eq. 0) go to 560
        write(*,550) nstep,udifmx
 550    format(1x,i4,3x,f11.5)
 560    if(udifmx .gt. uconv) go to 500
```

```
cc
      do 25 i=1,imax
        ruo(i)=run(i)
        uo(i)=un(i)
        ro(i)=rn(i)
        to(i)=tn(i)
 25   continue
      if(nstep .lt. mxstep) go to 1000
      stop
      end
cc.............................................................
      subroutine tdma(aa,bb,cc,rhs,xx)
      implicit real * 8 (a-h,o-z)
      dimension aa(220),bb(220),cc(220),rhs(220),xx(220)
      common /cons/ imax,imaxm,imaxm2,delt,dtdx,dt2dx,dt3dx,tdtdx,
     >              hdtdx,fdt3dx,dtdx2,tdtdx2,edtdx2,fdtdx2,gdtdx2
      do 1 i=2,imax
        im=i-1
        xt=aa(i)/bb(im)
        bb(i)=bb(i)-cc(im)*xt
        rhs(i)=rhs(i)-rhs(im)*xt
 1    continue
      xx(imax)=rhs(imax)/bb(imax)
      do 2 i=imaxm,1,-1
        xx(i)=(rhs(i)-cc(i)*xx(i+1))/bb(i)
 2    continue
      return
      end
cc.............................................................
      subroutine grid
      implicit real * 8 (a-h,o-z)
      common /cons/ imax,imaxm,imaxm2,delt,dtdx,dt2dx,dt3dx,tdtdx,
     >              hdtdx,fdt3dx,dtdx2,tdtdx2,edtdx2,fdtdx2,gdtdx2
      common /outpt/ dis(220), iout(50), iu, io
cc
      write(*,5)
 5    format(///15x,'SET GRID CONFIGURATION  (220 node maximum)',/)
      write(*,20)
 20   format(/1x,'Enter number of nodes')
      read(*,*) imax
      delx=1.0/imax
      dis(1)=delx/2.0
        do 100 i=2,imax
 100      dis(i)=dis(i-1)+delx
      imaxm=imax-1
      imaxm2=imax-2
      dtdx=delt/delx
      dt2dx=dtdx/2.0
      dt3dx=dtdx/3.0
      tdtdx=2.0*dtdx
      hdtdx=1.5*dtdx
      fdt3dx=4.0*dtdx/3.0
      dtdx2=dtdx/delx
      tdtdx2=2.0*dtdx2
      edtdx2=8.0*dtdx2/3.0
      fdtdx2=4.0*dtdx2
      gdtdx2=4.0*dtdx2/3.0
cc
      io=0
      iu=0
      ztest=0.0
      do 200 i=1,imax
        if(dis(i) .gt. ztest .and. ztest .lt. .0399) then
          ztest=ztest+0.005
```

```
          io=io+1
          iout(io)=i
          if(dis(i) .gt. .0399) ztest=dis(i)
        elseif(dis(i) .gt. ztest .and. ztest .lt. .0999) then
          ztest=ztest+0.01
          io=io+1
          iout(io)=i
          if(dis(i) .gt. .0999) ztest=dis(i)
        elseif(dis(i) .gt. ztest .and. ztest .lt. 1.0) then
          ztest=ztest+0.1
          io=io+1
          iout(io)=i
          if(dis(i) .gt. 1.0) ztest=dis(i)
        endif
        if(dis(i) .gt. 0.49 .and. iu .eq. 0) iu=i
 200  continue
      return
      end
cc.........................................................
      subroutine conduc(rtime,dis,alf,tc)
      implicit real * 8 (a-h,o-z)
      dimension f(127)
cc from error function table in Kreith
      data f/.00000,.02256,.04511,.06762,.09008,.11246,.13476,.15695,
     )       .17901,.20094,.22270,.24430,.26570,.28690,.30788,.32863,
     )       .34913,.36936,.38933,.40901,.42839,.44749,.46622,.48466,
     )       .50275,.52050,.53790,.55494,.57162,.58792,.60386,.61941,
     )       .63459,.64938,.66378,.67780,.69143,.70468,.71754,.73001,
     )       .74210,.75381,.76514,.77610,.78669,.79691,.80677,.81627,
     )       .82542,.83423,.84270,.85084,.85865,.86614,.87333,.88020,
     )       .88679,.89308,.89910,.90484,.91031,.91553,.92050,.92524,
     )       .92973,.93401,.93806,.94191,.94556,.94902,.95228,.95538,
     )       .95830,.96105,.96365,.96610,.96841,.97059,.97263,.97455,
     )       .97635,.97804,.97962,.98110,.98249,.98379,.98500,.98613,
     )       .98719,.98817,.98909,.98994,.99074,.99147,.99216,.99279,
     )       .99338,.99392,.99443,.99489,.99532,.99566,.99600,.99634,
     )       .99668,.99702,.99724,.99747,.99769,.99792,.99814,.99828,
     )       .99843,.99857,.99872,.99886,.99895,.99904,.99913,.99922,
     )       .99931,.99937,.99942,.99948,.99953,.99959,1.0/
      arg=dis/(2.0*dsqrt(rtime*alf))
      i=1+idint(0.0001+arg*50.0)
      if (i .gt. 126) erf=1.0
      if (i .gt. 126) go to 10
      erf=((arg-0.02*(dble(i)-1.0))/0.02)*(f(i+1)-f(i)) + f(i)
 10   tc=2.0-erf
      return
      end
```

****
READER

Basic Program for the HP Computer for reading thermocouples and pressure transducer voltages and converting them to temperatures and pressure

Notes: 1) Thermocouples are read on Channels 6 to 8.
2) This BASIC code is case sensitive. Commands are all caps and variables have only first letter cap.

****

```
10   REM
20   REM  READ THERMOCOULES AND PRESSURE TRANSDUCER
30   REM
40   REAL Volts(8,200),Time(200),Temp(7,200)
50   REAL Press(200)
60   CREATE BDAT "DATA1',200,48
70   ASSIGN @Path TO "DATA1"
80   CLEAR 709
90   OUTPUT 709;"ARAF6AL8VR1SD0VA0S01"
100  OUTPUT 709;"AC000"
110  ENTER 709;Ref
120  Idata=20
130  FOR N=1 TO Idata
140    FOR M=1 TO 8
150      Volts(M,N)=Ref
160    NEXT M
170  NEXT N
180  BEEP
190  WAIT 4
200  BEEP
210  Ti=TIMEDATE
220  FOR I=1 TO Idata
230    Time(I)=TIMEDATE
240    FOR J=6 TO 8
250      OUTPUT 709;"AS"
260      ENTER 709;Volts(J,I)
270    NEXT J
280    IF Volts(6,I)>.0005 THEN GOTO 300
290  NEXT I
300  Isave=I
310  Loop=0
320  FOR I=Isave+1 TO Idata
330    Loop=Loop+1
340    IF Loop=5 THEN
350      Loop=0
360      Jstart=4
370      OUTPUT 709;"AF4"
380    END IF
390    IF Loop=1 THEN
400      Jstart=6
410      OUTPUT 709;"AF6"
420    END IF
430    Time(I)=TIMEDATE
440    FOR J=Jstart TO 8
450      OUTPUT 709;"AS"
460      ENTER 709;Volts(J,I)
470    NEXT J
480  NEXT I
490  CLEAR 709
```

```
500  BEEP
510  P0=-5.115307103E-2
520  P1=2.485028007E+4
530  P2=-3.826622822E+5
540  P3=9.966105673E+7
550  P4=-1.082062357E+10
560  P5=6.039285524E+11
570  P6=-1.910899962E+13
580  P7=3.478234730E+14
590  P8=-3.399102821E+15
600  P9=1.382851398E+16
610  PRINT
620  FOR I=1 TO Idata
630    Time(I)=INT((Time(I)-Ti)*1000
640    FOR J=4 TO 7
650      Vt=Volts(J,I)-Ref
660      T1=P6+Vt*(P7+Vt*(P8+Vt*P9))
670      T2=P3+Vt*(P4+Vt*(P5+Vt*T1))
680      Tf=P0+Vt*(P1+Vt*(P2+Vt*T2))
690      Temp(J,I)=INT((Tf+273.0)*1.8)
700    NEXT J
710    Mivolt=Volts(8,I)*1000000.0
720    Press(I)=INT((.0557*(Mivolt-139)+.5))
730  NEXT I
740  FOR I= 1 TO Idata
750    OUTPUT @PATH,I;Time(I),Temp(4,I),Temp(5,I),Temp(6,I),Temp(7,I),Press(I)
760  NEXT I
770  FOR I=1 TO Idata
780    PRINT Time(I);
790    FOR J=4 TO 7
800      PRINT TAB(15+(J-4)*8);
810      IF Temp(J,I)>510 THEN PRINT Temp(J,I);
820    NEXT J
830    PRINT TAB(48);Press(I)
840  NEXT I
850  OUTPUT 709;"SD1"
860  END
```

VITA

D. Scott Crocker was born in Fulton, Kentucky to Edward and Gayle Crocker on April 16, 1961. He graduated from Jackson Central-Merry High School, Jackson, Tennessee in 1979. In June, 1983 he received a Bachelor of Science degree in Mechanical Engineering at the University of Tennessee at Knoxville. He then received a Master of Science degree in Mechanical Engineering at Georgia Institute of Technology, Atlanta, Georgia in December 1984. He worked for McDonnell Douglas in Huntsville, Alabama until September, 1987. In August of 1991 he received a Doctor of Philosophy degree in Mechanical Engineering at the University of Tennessee at Knoxville and began employment with Computational Fluid Dynamics Research Corporation (CFDRC) in Huntsville, Alabama.