NASA Technical Paper 1895

Computer Program for Pulsed Thermocouples With Corrections for Radiation Effects

Herbert A. Will

SEPTEMBER 1981

NASA

NASA Technical Paper 1895

Computer Program for Pulsed Thermocouples With Corrections for Radiation Effects

Herbert A. Will
Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1981

Summary

A pulsed thermocouple is used for measuring gas temperatures above the melting point of common thermocouples. This is done by allowing the thermocouple to heat until it approaches its melting point and then turning on the protective cooling gas. This method requires a computer to extrapolate the thermocouple data to the higher gas temperatures. In earlier work by this author the extrapolation was done by using a first-order exponential curve fit to predict the final thermocouple wire temperature. Since radiation effects were neglected, the gas temperature was not computed. Hand calculations had to be used to estimate the gas temperature. This report describes a method that includes the effect of radiation in the extrapolation. Computations of gas temperature are provided, along with the estimate of the final thermocouple wire temperature. Results from tests on high-temperature combustor research rigs are presented.

Introduction

An earlier investigation by the author (ref. 1) described the use of a pulsed thermocouple to measure gas temperatures above the melting point of common thermocouples. This method of measuring temperature is intended for the measurement of temperatures at the exit of experimental aircraft combustors at temperatures to 2400 K and pressures to 4 MPa (40 atm). The previous investigation described an approach that uses a thermocouple cooled by a small jet of inert gas. When a measurement is to be made, the cooling jet is turned off and the thermocouple is allowed to heat up to near its melting point. When the temperature of the thermocouple approaches its melting point, the cooling is reapplied. The data are then fitted to a first-order exponential function. The final temperature that the thermocouple would have attained is then calculated by extrapolation.

The computer program (ref. 1) did not take into account the fact that at the higher temperatures the heating curve deviates from a true exponential. This deviation is the result of radiant energy (obeying Stephan's T^4 law) being absorbed or emitted by the thermocouple wire.

The analysis described in this report takes into account the T^4 radiation terms in the differential equation describing the temperature of the thermocouple wire as a function of time. The report describes the solution of this differential equation for time as a function of temperature. This solution cannot be inverted (except numerically) to give temperature as a function of time. A computer program is described that fits measured data to the theoretical curve based on this more complete analysis. The computer program uses the gradient-expansion method (ref. 2) to fit the data to the theoretical function. The program computes final thermocouple wire temperature and final gas temperature.

This report also presents typical input and results for the computer program. Data and results are discussed from tests in two combustor test facilities.

Theory

This section describes the theoretical equations necessary to compute gas temperatures with a pulsed thermocouple. Most of the time the thermocouple is protected with a jet of cooling gas, as shown in figure 1. When a temperature measurement is to be made, the cooling gas is turned off and the thermocouple output is sampled at a high rate and recorded. Just before the thermocouple reaches its melting point the cooling is reapplied to protect the thermocouple wire. The gas temperature can then be calculated by extrapolation from the initial heating curve. For the extrapolation to be valid, it must be based on a theoretical heating curve. The derivation of the theoretical equation is described here. All symbols are defined in appendix A.

The equation that describes the pulsed-thermocouple wire temperature can be derived from the basic heat transfer relations (ref. 3). Assume a bare wire thermocouple with infinitely long leads in a hot gas stream. This assumption causes the conduction effects to be neglected. Very little error is introduced if we neglect the transfer of heat to the junction by conduction along the wire for carefully designed probes. Thus in the absence of conduction, heat can be transferred to the wire by convection of the gas, by radiation from the gas, and by radiation from the duct walls. Also heat can be transferred away from the wire by radiation.

The rate of heat storage in the wire will be equal to the rate of heat entering the wire minus the rate of the heat leaving the wire. The rate of heat storage q_s per

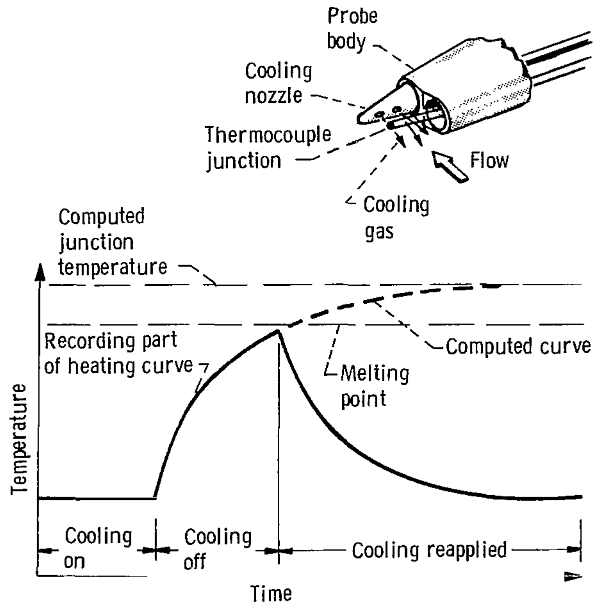

Figure 1. - Thermocouple heating curve.

unit length is given by

$$q_s = q_c + q_r \tag{1}$$

where q_c is the rate of heat convected per unit length to the wire by the gas and q_r is the net heat radiated per unit length to the wire.

The rate of heat storage per unit length of the wire is given by (ref. 3)

$$q_s = \rho C \frac{\pi D^2}{4} \frac{dT_w}{dt} \tag{2}$$

where ρ is the wire density, C is the specific heat of the wire, T_w is the wire temperature, t is the time, and D is the wire diameter.

The rate of heat transfer to the wire by convection q_c is given by (ref. 3)

$$q_c = \pi \, \mathrm{Nu} \, K_g \, P_{sc} \, (T_g - T_w) \tag{3}$$

where Nu is the Nusselt number, K_g is the thermal conductivity of the gas, P_{sc} is the probe shape constant, and T_g is the gas temperature. For an infinitely long wire in crossflow P_{sc} is unity. The probe shape constant was introduced to take into account the fact that the presence of a probe to support the wire will cause a reduction in the effective Nusselt number of the thermocouple. In practice the P_{sc} must be determined experimentally and generally falls in the range 0.8 to 1.0.

The rate of heat transfer by radiation q_r is given by (ref. 3)

$$q_r = \sigma \epsilon_w \left[(1 - \alpha) T_d^4 + \epsilon_g T_g^4 - T_w^4 \right] \pi D \tag{4}$$

where σ is the Stefan-Boltzmann constant, ϵ_w is the emissivity of the wire, α is the effective absorptivity of the gas, T_d is the duct temperature, and ϵ_g is the emissivity of the gas. The first term in equation (4) represents the heat received by the wire from the hot walls of the duct. The second term represents radiant heat received from the gas. The third term represents radiant heat emitted from the wire.

Combining equations (1) to (4) gives

$$dt = \frac{-K_1 \, dT_w}{T_w^4 + K_2 T_w - K_3} \tag{5}$$

where

$$K_1 = \frac{\rho C D}{4 \sigma \epsilon_w} \tag{6}$$

$$K_2 = \frac{\mathrm{Nu} K_g P_{sc}}{D \sigma \epsilon_w} \tag{7}$$

and

$$K_3 = K_2 T_g + \left[(1 - \alpha) T_d^4 + \epsilon_g T_g^4 \right] \tag{8}$$

To solve equation (5), we integrate both sides of the equation. The integration is easier if we factor the denominator. The roots of a fourth-order equation can be found by algebraic methods (ref. 4). The roots of the equation are

$$T_w = \alpha_1 \pm i\beta, \alpha_2, \alpha_3 \tag{9}$$

where

$$\alpha_1 = \frac{1}{2} \sqrt{Y_1} \tag{10}$$

$$\beta = \frac{1}{2} \left(Y_1 + \frac{2K_2}{\sqrt{Y_1}} \right)^{1/2} \tag{11}$$

$$\alpha_2 = \frac{1}{2}\left[\sqrt{Y_1} + \left(-Y_1 + \frac{2K_2}{\sqrt{Y_1}}\right)^{1/2}\right] \quad (12)$$

$$\alpha_3 = -\frac{1}{2}\left[\sqrt{Y_1} + \left(-Y_1 + \frac{2K_2}{\sqrt{Y_1}}\right)^{1/2}\right] \quad (13)$$

$$Y_1 = \left[\frac{K_2^2}{2} + \left(\frac{K_2^4}{4} + \frac{64K_3^3}{27}\right)^{1/2}\right]^{1/3} + \left[\frac{K_2^2}{2} - \left(\frac{K_2^4}{4} + \frac{64K_3^3}{27}\right)^{1/2}\right]^{1/3} \quad (14)$$

Equation (5) can then be rewritten as

$$dt = \frac{-K_1\, dT_w}{(T_w - \alpha_1 - i\beta)(T_w - \alpha_1 + i\beta)(T_w - \alpha_2)(T_w - \alpha_3)} \quad (15)$$

or

$$dt = \left[\frac{H_1}{T_w - \alpha_2} + \frac{H_2}{T_w - \alpha_3} + \frac{H_3}{T_w - \alpha_1 - i\beta} + \frac{H_3^*}{T_w - \alpha_1 + i\beta}\right] dT_w \quad (16)$$

where

$$H_1 = \frac{-K_1}{(\alpha_2 - \alpha_3)\left[(\alpha_2 - \alpha_1)^2 + \beta^2\right]} \quad (17)$$

$$H_2 = \frac{-K_1}{(\alpha_3 - \alpha_2)\left[(\alpha_3 - \alpha_1)^2 + \beta^2\right]} \quad (18)$$

$$H_3 = \frac{-K_1}{(\alpha_1 - \alpha_2 + i\beta)(\alpha_1 - \alpha_3 + i\beta)(2i\beta)} \quad (19)$$

If the denominator of H_3 is multiplied out, we get

$$H_3 = \frac{-K_1}{E + iF} \quad (20)$$

where

$$E = -2\beta^2(2\alpha_1 - \alpha_2 - \alpha_3) \quad (21)$$

and

$$F = 2\left[\beta(\alpha_1 - \alpha_2)(\alpha_1 - \alpha_3) - \beta^3\right] \quad (22)$$

Thus H_3 can be rewritten as

$$H_3 = \frac{-K_1 E}{E^2 + F^2} + i\frac{K_1 F}{E^2 + F^2} \equiv H_{3A} + iH_{3B} \quad (23)$$

Equation (16) can then be integrated to get

$$t = H_1 \ln(\alpha_2 - T_w) + H_2 \ln(T_w - \alpha_3) + H_{3A}\ln\left[(T_w - \alpha_1)^2 + \beta^2\right] + 2H_{3B}\tan^{-1}\left(\frac{\beta}{T_w - \alpha_1}\right) + H_4 \quad (24)$$

where H_4 is a constant of integration.

Equation (24) shows the theoretical relationship between the wire temperature T_w and the time t. In general all the parameters in the equation are known except for the gas temperature T_g, the probe shape constant P_{sc}, and the integration constant H_4. After a measurement a set of wire temperature readings are known. The procedure used finds the values of T_g, P_{sc}, and H_4 that result in the best fit of the temperature data to the theoretical equation (eq. 24)). The next section describes the computer program written to fit equation (24) to the data.

Description of Computer Program

The FORTRAN IV computer program described in this report is designed to calculate gas temperature by using data taken from a separate pulsed-thermocouple controller. A listing of the program and its various subroutines is shown in appendix B. The program input requirement is a set of wire temperatures taken at regular time intervals, the Mach number, the total pressure, the wall temperature, and the probe shape constant. The computer program output is the extrapolated wire temperature and the computed gas temperature. In addition, if the probe shape constant has not been entered, the computer program will calculate and output PSC, the probe shape constant.

The program uses a curve-fitting procedure from reference 2 called the gradient-expansion method to fit the theory to the input data. Two parameters, gas temperature TGAS and possibly probe shape constant PSC, are adjusted for best fit of the theory to the data. These parameters are adjusted until the sum of the squares of the differences between the measured wire temperature and the theoretical wire temperature is a minimum. The error, which is called CHISQR, is defined by

$$\text{CHISQR} = \sum_{i=1}^{n} \left[\left(T_{\text{data}}\right)_i - \left(T_{\text{theory}}\right)_i \right]^2 \tag{25}$$

where T_{data} is the measured wire temperature, T_{theory} is the corresponding theoretical wire temperature, and n is the number of measured data points. Note that the theoretical wire temperatures must be evaluated point by point at the same values of the time parameter used for the measured data.

Both the gradient-expansion procedure and the evaluation of CHISQR require computation of theoretical wire temperature at every measurement time. In addition, the gradient-expansion method requires values for $\partial T_w / \partial T_g$ and $\partial T_w / \partial P_{sc}$ at every measurement time. These requirements create a difficulty because the analytical solution to the differential equation expresses time as a function of wire temperature in equation (24). The equation cannot easily be inverted to yield the needed wire temperature as a function of time and its derivatives. As a result a great amount of the computer time is devoted to numerically inverting the equation and evaluating the derivatives. Since theoretical wire temperature values at the measurement times are not available directly from equation (24), they are calculated by interpolating in a table of wire temperature-time pairs that do satisfy equation (24). This table must be regenerated whenever equation parameters are changed.

This procedure must be repeated once for every evaluation of wire temperature and twice for every evaluation of the derivatives. The derivatives are approximated by computing the differences in wire temperature that result for two values of the parameters TGAS and PSC: one value slightly above the present value and one value slightly below the present value.

The main computer program takes care of reading the input data, calling the curve-fitting routines, deciding when the curve fit is good enough, and writing the results. Initially input data of Mach number, pressure, and duct temperature are read in as well as 1000 readings of thermocouple wire temperature. The temperatures represented by these numbers are taken at equal time intervals before and during the temperature rise. The first 100 readings represent the thermocouple wire temperature while the cooling air is on. The rest of the 900 temperature readings are taken during the temperature rise of the thermocouple wire when the cooling air is turned off. If the cooling air is turned on again before the 900 readings are taken, the remaining readings are zero.

After the data are read in, a call to subroutine STCFIT determines the best estimate of the temperature ramp starting time. This is necessary because the theoretical curve is always forced to pass through this point.

With the starting time determined, the curve-fitting process begins. Repetitve calls to CURFIT and FDERIV result in adjustments to several parameters such that CHISQR is decreased. With every adjustment in the parameter values a call to CONGEN is needed to evaluate the constants in equation (24). The parameters adjusted include the gas temperature TGAS; the probe shape constant PSC; and FLAMDA, a parameter whose value controls the curve-fitting process. The probe shape constant is adjusted only if its value is not included in the input data. If the PSC is to be adjusted, the variable NTERMS is set equal to 2 by the computer program; otherwise NTERMS is set equal to 1 and only TGAS is adjusted. Thus the main program recalls FDERIV and CURFIT until the decrease in CHISQR is less then 1 percent. This value of 1 percent was chosen by trial-and-error methods to provide a wire temperature within 1 or 2 K of the ultimate wire temperature without using an unreasonable amount of computer time.

Subroutine CURFIT

Subroutine CURFIT makes a least-squares fit to a nonlinear function by using the gradient-expansion algorithm described in appendix C. The algorithm is really two curve-fitting techniques combined into one program. One of the techniques works well when the variables are far from the correct values, and the other works well when they are close to the final values. A parameter λ (called FLAMDA in the program) is used to change the curve-fitting routine gradually from one technique to the other.

The subroutine works by starting with FLAMDA = 0.001 (when FLAMDA is less than 1 the fitting technique that works close to the minimum is dominant—see appendix C). The error χ^2 (appendix C) between the measured and theoretical data is called both CHISQ1 and CHISQR in the program. CHISQ1 is an initial value of χ^2 calculated once when the subroutine is entered. The program makes changes in the wire temperature, the probe shape constant, and FLAMDA until a new value of χ^2 (called CHISQR) starts to decrease, at which time FLAMDA is divided by 10 and the subroutine returns to the calling program. It is the responsibility of the calling program to check CHISQR to see if the change in CHISQR since the last call to CURFIT is small enough to stop the program. If it is not, subroutine CURFIT should be called again without changing the value of the current FLAMDA.

Subroutine FDERIV

Subroutine FDERIV computes data needed by the curve-fitting routine CURFIT. The data needed are the derivatives of the wire temperature with respect to both gas temperature and the probe shape constant. Also needed are theoretical values of wire temperature evaluated at the measured time (the times corresponding to the measured wire temperatures). The derivatives are determined from (ref. 5)

$$\frac{\partial T_w}{\partial T_g} = \frac{T_w(T_g+1, \mathrm{PSC}) - T_w(T_g-1, \mathrm{PSC})}{2} \tag{26}$$

$$\frac{\partial T_w}{\partial \mathrm{PSC}} = \frac{T_w(T_g, \mathrm{PSC}+0.001) - T_w(T_g, \mathrm{PSC}-0.001)}{2(0.001)} \tag{27}$$

If the probe shape constant is not to be calculated (NTERMS = 1), only equation (26) will be calculated. The theoretical values of wire temperature are generated from equation (24) with a call to subroutines TABL and INTRP.

The subroutine returns a 1000- by 3-element array. The derivative of the wire temperature with respect to the gas temperature at time I is returned in array DERIV(I,1). The derivative of the wire temperature with respect to the probe shape constant at time I is returned in array DERIV(I,2). The table of the computed wire temperatures at time I is returned in array DERIV(I,3).

Function XICALC

Function XICALC computes the sum of the squares of the differences between the measured wire temperature and the theoretical wire temperature (from the numerically inverted equation (24)). The sum of the squares of the differences will be

$$\mathrm{XICALC} = \sum_{I=START}^{RANGE} \left[T_w(I) - T_{\mathrm{theory}}(I)\right]^2 \tag{28}$$

The program first calls subroutine CONGEN to generate new constants for equation (24) since the gas temperature and the probe shape constant may have changed. Subroutine TABL is then called to generate a table of theoretical temperatures and times. The interpolation necessary is done by this subroutine and not by subroutine INTRP because the output of this routine is a single number, the error XICALC, and not an entire table of numbers.

Subroutine TABL

The purpose of subroutine TABL is to generate values of theoretical wire temperatures and times for subroutine INTRP. Subroutine CURFIT, FDERIV, and function XICALC require a value of theoretical wire temperature at every measurement time. These wire temperatures must be obtained by inverting equation (24). However, because of the form of equation (24) a numerical inversion will have to be done. A call to subroutine TABL generates a table of temperature-time pairs that satisfy equation (24). Then a call to INTRP interpolates in this table to get temperatures at the measurement times.

To generate the interpolation table, a set of temperatures is needed to put into equation (24) to obtain computed times. The values of computed time that result from equation (24) should be as close as possible to the measured times for accurate interpolation by subroutine INTRP. The set of temperatures is determined one at a time, starting with a known point on the theoretical curve. Each succeeding temperature is computed from the previous one by using a linear approximation to the theoretical curve (fig. 2). The linear approximation will have a slope equal to the slope of the theoretical

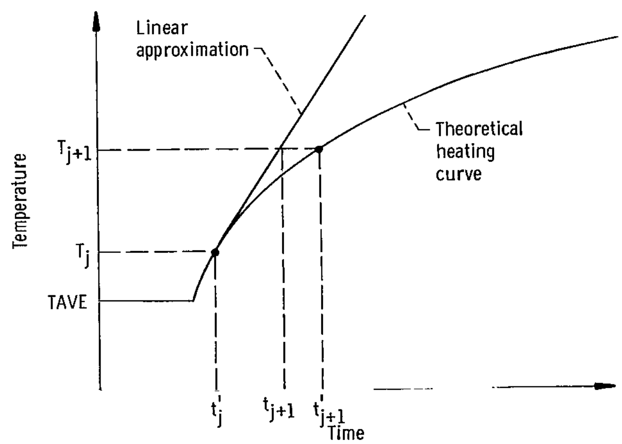

Figure 2. - Graphical representation of linear approximation to theoretical wire heating curve. The t_j are measured times and the t'_j are computed times from equation (24).

curve at the previous temperature. Thus each succeeding temperature will be

$$T_{j+1} = T_j + \frac{t_{j+1} - t_j'}{(dt/dT_w)_{T_w = T_j}} \tag{29}$$

where $j = 1,2,3,\ldots, n$ measured data points. The times corresponding to the measured data points are t_j. The times t_j' are computed by evaluating equation (24) with $T_w = T_j$. The derivative of equation (24) is

$$\frac{dt}{dT_w} = \frac{-H_1}{\alpha_2 - T_w} + \frac{H_2}{T_w - \alpha_3} + \frac{2[H_{3A}(T_w - \alpha_1) - H_{3B}\beta]}{(T_w - \alpha_1)^2 + \beta^2} \tag{30}$$

In the program $t_{j+1} - t_j'$ is defined as DELTIM and

$$\text{DELTMP} \equiv \frac{\text{DELTIM}}{(dt/dT_w)_{T_w = T_j}} \tag{31}$$

The program starts by setting $T_j = T_1$ = TAVE, which is the temperature on the theoretical curve; and $t_j' = t_1'$ is equal to MSTIME*START. The next temperature T_{j+1} is evaluated by setting $t_{j+1} = t_2$ = MSTIME*(START + 1) in equation (29). What results is a table of theoretical time-temperature pairs that do satisfy equation (24), where the times are not exactly equal to the measurement times. The array of times is called TIMC, and the array of temperatures is called TC in the program. A linear interpolation will need to be done because temperatures at the exact measurement times are needed.

Subroutine INTRP

Subroutine INTRP is used to correct the table of theoretical temperatures (array TC) generated by subroutine TABL. Subroutine INTRP performs a linear interpolation between the calculated data points so that the calculated times (and corresponding temperatures) fall exactly on the measured time. The resulting interpolated values of temperature are stored in array TC.

Subroutine STCFIT

Subroutine STCFIT determines the starting point of the thermocouple temperature rise. The starting point is defined as the intersection of two straight lines. One line is the best fit through the data before

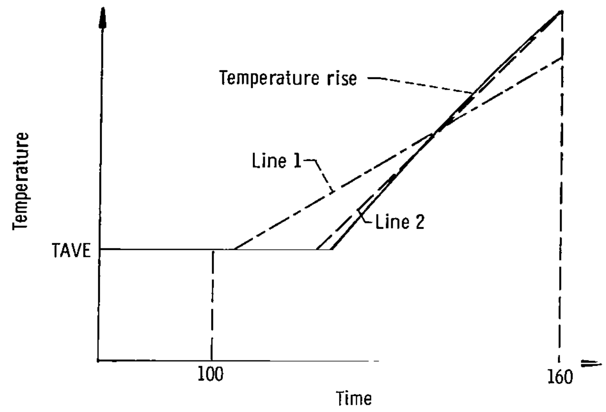

Figure 3. - Search process for subroutine STCFIT.

the cooling is turned off. This line is called TAVE. The other line is the best fit through approximately the first 50 points of the temperature rise. Since a solenoid is used to turn the cooling air on and off, there will be some delay between when the power is removed and when the cooling air actually stops flowing. The solenoid power is turned off at data point 100, and the starting point search ranges between data points 100 and 130.

The starting point of the search process is shown in figure 3. A standard least-squares fit to a straight line of the data from point 100 to point 160 is performed. In general, point 100 is not the true starting point; so this line (line 1 in fig. 3) will not intersect the TAVE line at point 100. In fact, if the starting point of the data for the least-squares line is varied from 100 to 130, the intersection of the least-squares line (line 2) with TAVE will approach the true starting point and then back away. Therefore the intersection point will have a maximum as the starting point is varied. The output of this routine is this maximum value of the starting point. This represents the best approximation to the start of the ramp.

Subroutine CONGEN

Subroutine CONGEN computes the constants necessary to evaluate equation (24). Constants K_1, K_2, and K_3 are evaluated by using equations (6) to (8). The wire emissivity ϵ_w for clean platinum was found to be (ref. 6)

$$\epsilon_w \approx 0.085 + (0.76\text{E}{-4})T_{wf} \tag{32}$$

where T_{wf} is the final wire temperature in K. The other parameters used for platinum (type R) thermocouple wire are (ref. 7):

Wire density, kg/m^3 .... 0.2078×10^5
Stefan-Boltzmann constant,
$J/K^4 \sec m^2$ .... 0.56697×10^{-7}
Wire specific heat, J/kg K .... 0.1427×10^{-3}
Gas effective absorptivity 0
Gas effective emissivity 0
Wire diameter, m .... 0.8128×10^{-3}

$$K_g = (0.3007\ \text{E}-3) * \text{TGAS}^{0.78}\ \text{J/(sec K m)} \quad (33)$$

$$\text{Nu} = 188.41 * (\sqrt{\text{WDIA} * \text{MN} * \text{P}}) * \text{TGAS}^{-0.6} * \left|1 + 0.2 * (\text{MN})^2\right|^{-1/4} \quad (34)$$

where WDIA is the wire diameter, MN is the Mach number, P is the pressure in pascals, and T_g is in K. The gas effective absorptivity and emissivity are assumed to be zero. This corresponds to a transparent gas and the worst case for radiation effects.

The subroutine also computes α_1, α_2, α_3, β, H_1, H_2, H_{3A}, and H_{3B} from equations (10) to (23). The value of H_4 is computed by putting the initial conditions into equation (24) and solving for H_4. The initial temperature is the average cooled temperature TAVE. The initial time is the measurement time interval MSTIME times START.

Function EVALTM

Function EVALTM evaluates equation (24) to obtain a calculated time for an input of wire temperature. The input wire temperature must be between the initial average cooled temperature TAVE and α_2 in order to avoid taking the logarithm of a negative number. Values of α_1, α_2, α_3, β, H_1, H_2, H_{3A}, and H_4 must have been previously calculated with a call to the CONGEN subroutine.

Subroutine MATINV

Subroutine MATINV does an inversion of a 1- or 2-degree matrix. For a 1-degree matrix only a simple reciprocal is needed. For a 2-degree matrix the adjoint matrix is calculated. Then each element is divided by the determinant to form the inverse matrix. The original matrix is then replaced by its inverse.

Tests and Results

A pulsed-thermocouple system was tested in a combustor rig at the Air Force Wright Aeronautical Laboratory (AFWAL) as part of a joint AF–NASA program on instrumentation. The system included a probe, a sample-and-hold voltmeter, a microcomputer-based controller, and a digital recorder, as shown in figure 4. Figure 5 shows the probe that was put into the combustor. The probe consisted of a water-cooled shell with a replaceable platinum (type R) thermocouple. Compressed-air cooling for the thermocouple was controlled by a fast-acting solenoid valve. The thermocouple voltage was converted to digital form by a sample-and-hold digital voltmeter. A microcomputer was used to control the voltmeter and turn the cooling air on and off. The time between data points (called MSTIME) was controlled at 0.0042 second. This value was chosen so that most of the ramp would be included in the 1000 data points. If a different probe with a different time constant were used, this MSTIME would have to be changed.

A full curve including the final wire temperature could be recorded for each pulse because the gas stream of the combustor configuration under test was not hot enough to require the cooling air to come on. The data were first processed by the computer program to compute the probe shape constant. The average computed probe shape constant for 20 pulses at fixed combustor conditions was 0.91, with a maximum deviation of 0.09. This deviation is the result of the fact that the burning process is not constant during the pulse and thus results in a temperature that can vary during the pulse by as much as 3.2 percent.

With the average probe shape constant of 0.91 the data were curve fit 60 percent of the way up the

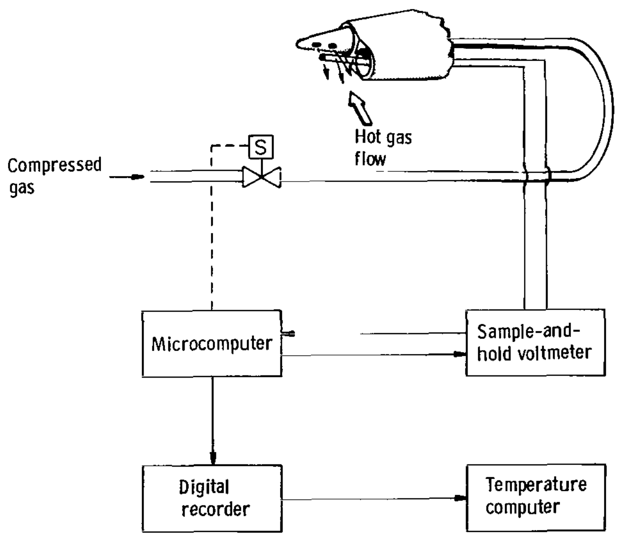

Figure 4. - Block diagram of pulsed-thermcouple system.

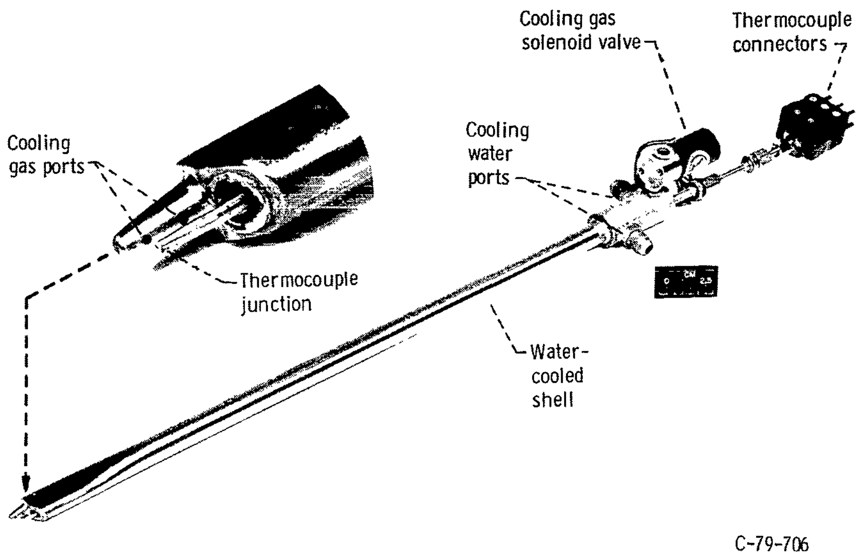

Figure 5. - Thermocouple probe.

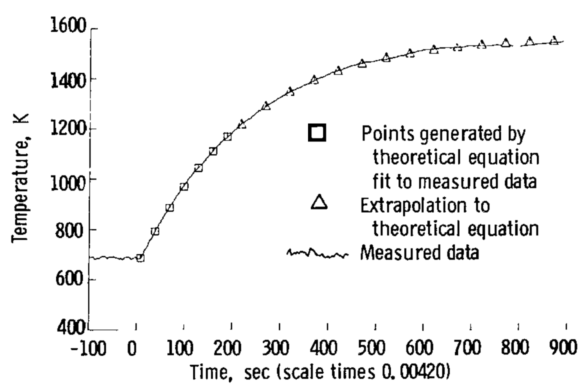

Figure 6. - Results with final temperature well below wire melting point.

curve. It is estimated that at least 60 percent of the curve could be measured at the highest expected gas temperatures. A typical result is shown in figure 6. The solid line is the measured data (a total of 1000 data points). The triangles and squares represent the theoretical curve. The squares represent the portion of the curve that was used in the computation. The triangles represent the portion of the curve that was

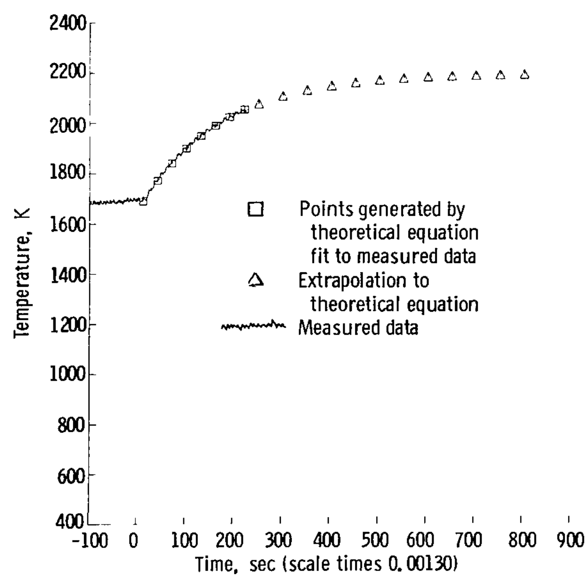

Figure 7. - Results with final temperature near wire melting point.

extrapolated by using the theoretical curve. The computed final wire temperature varied from 1525 K to 1581 K, with an average of 1561 K for the 20 readings. The actual final wire temperature varied from 1525 K to 1575 K because of fluctuations in the burning. A comparison between the final wire temperature computed using 60 percent of the ramp and the actual final wire temperature measured for the 20 readings showed a maximum deviation of 3 percent.

The average of the 20 computed gas temperatures was 1691 K, with a maximum deviation of 47 K, or 2.7 percent. The difference of 130 K between the computed wire temperature and the gas temperature is the radiation error. It is estimated that the radiation error can be computed to within about 20 percent, which for this case would be ±26 K.

Results for a pulsed-thermocouple probe different from the probe just described were obtained during a high-temperature combustor test at the Lewis Research Center as shown in figure 7. The probe shape constant for this geometry was determined at lower temperatures than shown in figure 7 to be 0.96. The gas temperature for the data shown in figure 7 was 2300 K, and the final computed wire temperature was 2190 K. The wire melts at 2215 K. The protective compressed air was set to turn on at about 2000 K in order to assure a long thermocouple life.

Concluding Remarks

The pulsed thermocouple was developed as an instrument to determine high gas temperatures. The pulsed feature is needed at temperatures above the melting point of common thermocouples or when streaking of a combustion process is occurring. The cooling gas was found to adequately protect the thermocouple during this high-temperature operation.

The computer program for computing gas temperature was designed to take the T^4 radiation error into account. The program requires as input the Mach number, the wall temperature, and the total pressure in addition to the thermocouple data. Tests at temperatures below the melting point of platinum thermocouples show that the pulsed-thermocouple system can compute the gas temperature to within about 4 percent with as little as 60 percent of the temperature step as input data.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, December 15, 1980

Appendix A
Symbols

Mathematical symbol	Computer symbol	Definition
a	-----	parameter of function χ^2
C	SPHT	specific heat of wire
D	WDIA	wire diameter, m
$H_1, H_2, H_{3A}, H_{3B}, H_4$	H1,H2,H3A, H3B,H4	intermediate constants
K_g	-----	thermal conductivity of gas, J/sec K m
K_1, K_2, K_3	K1,K2,K3	intermediate constants
-----	MN	Mach number
Nu	NU	Nusselt number
P	P	pressure, Pa
P_{sc}	PSC	probe shape constant
q_c	-----	rate of heat transferred by convection into surface of wire, J/sec m
q_r	-----	rate of heat transferred by radiation, J/sec m
q_s	-----	rate of heat storage, J/sec m
T	-----	temperature, K
-----	TAVE	average temperature
T_d	TDUCT	duct temperature
T_g	TGAS	gas temperature
T_w	TWIRE	wire temperature
T_{wf}	TWF	final wire temperature
t	-----	time
X, Y	-----	general independent variables
Y_1	-----	intermediate constant
α	ALPHAG	effective absorptivity of gas
$\alpha_1, \alpha_2, \alpha_3$	ALPHA1, ALPHA2 ALPHA3	intermediate constants
β	BETA	intermediate constant
ϵ_g	EGAS	emissivity of gas
ϵ_w	E1 + E2*T	emissivity of wire
σ	SIGMA	Stefan-Boltzmann constant, J/K^4 sec m^2
χ^2	CHISQR, CHISQ1	least-squares error

Appendix B
Computer Programs

```
C       ROUTINE FOR CURVE FITTING DATA FROM A PULSED THERMOCOUPLE.
C       THERMOCOUPLE DATA SHOULD BE CONSTANT FOR THE FIRST 100
C       DATA POINTS (THERMOCOUPLE COOLED). THE THERMOCOUPLE IS THEN
C       HEATED ON AN EXPONENTIAL HEATING CURVE (900 DATA POINTS).
C       THE PROGRAM NEGLECTS CONDUCTION ERRORS.
C
C
C
C       INPUTS REQUIRED ARE:
C        WIRE TEMPERATURE (1000 DATA POINTS) (K).
C        MACH NUMBER.
C        PRESSURE (Pa).
C        DUCT TEMPERATURE (K).
C        PROBE SHAPE CONSTANT.
C
C       ALSO THE FOLLOWING PARAMETERS MUST BE SET TO
C       THE PROPER VALUE DEPENDING ON THE TYPE OF
C       THERMOCOUPLE USED:
C        MSTIME = TIME BETWEEN MEASUREMENTS (THIS PROGRAM).
C        WDIA = WIRE DIAMETER. (SUBROUTINE CONGEN).
C        WDENS = WIRE DENSITY. (SUBROUTINE CONGEN).
C        SPHT = WIRE SPECIFIC HEAT. (SUBROUTINE CONGEN).
C        WIRE EMISSIVITY. (SUBROUTINE CONGEN).
C        EGAS = EMISSIVITY OF GAS. (SUBROUTINE CONGEN).
C        ALPHAG = ABSORPTIVITY OF GAS. (SUBROUTINE CONGEN).
C
C
C
        REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        INTEGER START,RANGE
C
        REAL TC(1000,2),TIMC(1000,2),DERIV(1000,3)
C
        DATA MSTIME/0.42E-2/
C       MSTIME IS IN SECONDS.
C
        INTEGER I,NTERMS
        REAL CHISQO,FLAMDA,TGAS,TWF,PSC,CHISQR,X
C
        COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        COMMON /BLK2/START,RANGE
C
```

```
      COMMON /BLK3/TC,TIMC,DERIV
C
C
C
C     READ TEMPERATURE DATA (TWIRE)
C     IN DEG. K . (1000 DATA POINTS).
C
      DO 20 I=1,1000
 20   READ(1,80) TWIRE(I)
C
C     READ INPUT DATA
C
      WRITE(7,30)
 30   FORMAT(1X,'INPUT MACH NUMBER')
      READ(5,40) MN
 40   FORMAT(F7.4)
      WRITE(7,50)
 50   FORMAT(1X,'INPUT PRESSURE IN Pa.')
      READ(5,60) P
 60   FORMAT(F11.3)
      WRITE(7,70)
 70   FORMAT(1X,'INPUT DUCT TEMPERATURE IN DEG. K.')
      READ(5,80) TDUCT
 80   FORMAT(F9.2)
      WRITE(7,90)
 90   FORMAT(1X,'INPUT PROBE SHAPE CONSTANT.')
      READ(5,100) PSC
 100  FORMAT(F6.3)
C
C     THE NEXT 3 STATEMENTS ARE NEEDED ONLY
C     FOR THE EXAMPLE IN THIS REPORT.
C
      WRITE(7,101)
 101  FORMAT(1X,'MACH NUMBER TEMPERATURE DEG. K.')
      READ(5,80) MTMP
C
C     AVERAGE COOLED WIRE TEMPERATURE.
C
      TAVE = 0.0
      DO 110 I=1,99
 110  TAVE = TAVE+TWIRE(I)
      TAVE = TAVE/99.
C
C     DETERMINE START OF RAMP.
C
 115  CALL STCFIT
C
C     TEMPERATURE OVER MELTING POINT?
C
 118  DO 120 RANGE=100,1000
      IF (TWIRE(RANGE).LE.400.) GO TO 130
 120  CONTINUE
```

```
 130    RANGE = RANGE-1
C
C       CURVE FIT.
C
        CHISQO = 0.
        FLAMDA = 0.001
        TWF = TWIRE(RANGE)
        TGAS = TWF
        NTERMS = 1
 135    IF (PSC.NE.0.) GO TO 140
        NTERMS = 2
        PSC = 0.8
C
C       "FDERIV" COMPUTES THE DERIVITIVE OF TWIRE WITH RESPECT
C       TO TGAS & PSC. ALSO IT RETURNS VALUES OF CALCULATED
C       THEORETICAL WIRE TEMPERATURE AS A FUNCTION OF TIME.
C
 140    CALL FDERIV(TGAS,PSC,NTERMS,TWF)
C
C       "CURFIT" MODIFIES TGAS AND PSC TO OBTAIN THE BEST
C       MATCH BETWEEN THE THEORETICAL CURVE AND THE ACTUAL DATA.
C
        CALL CURFIT(NTERMS,PSC,TGAS,CHISQR,FLAMDA,TWF)
C
C       "CHISQR" IS THE ERROR BETWEEN THE THEORETICAL CURVE AND
C       THE ACTUAL MEASURED DATA. IF THERE IS LESS THAN A
C       ONE PERCENT CHANGE IN THE ERROR SINCE THE LAST
C       CALL TO CURFIT THEN THE PROGRAM IS FINISHED.
C
        X = ABS((CHISQR-CHISQO)/CHISQR)
        IF (X.LT.0.01) GO TO 150
        CHISQO = CHISQR
        TWF = ALPHA2
        GO TO 140
 150    WRITE(7,160) TGAS
 160    FORMAT(1X,'GAS TEMPERATURE = ',F9.2,' K')
        WRITE(7,170) ALPHA2
 170    FORMAT(1X,'FINAL WIRE TEMPERATURE = ',F9.2,' K')
        WRITE(7,180) PSC
 180    FORMAT(1X,'PROBE SHAPE CONSTANT = ',F6.3)
        STOP 123
        END



C
C       SUBROUTINE CURFIT(NTERMS,PSC,CHISQR,FLAMDA,TWF)
C
C
```

```
C        PURPOSE
C        THIS SUBROUTINE MAKES A LEAST SQUARES CURVE FIT TO
C        A NON-LINEAR FUNCTION.
C
C        TIME = SET OF INTEGERS TAKEN AS INDEPENDENT VARIABLE.
C        TWIRE = ARRAY OF WIRE TEMPERATURE READINGS TAKEN
C                AS DEPENDENT VARIABLE.
C        START = INTEGER VALUE OF TIME FOR START OF DATA.
C        RANGE = INTEGER VALUE OF TIME FOR END OF DATA.
C        NTERMS = NUMBER OF PARAMETERS (MAX. = 2).
C        TGAS = PARAMETER 1: GAS TEMPERATURE.
C        PSC  = PARAMETER 2: PROBE SHAPE CONSTANT.
C        A = ARRAY OF PARAMETERS.
C        FLAMDA = PROPORTION OF GRADIENT SEARCH INCLUDED.
C        TWF = ESTIMATED FINAL WIRE TEMPERATURE.
C        CHISQR = CHI SQUARE FOR FIT.
C
         SUBROUTINE CURFIT(NTERMS,PSC,TGAS,CHISQR,FLAMDA,TWF)
C
C
C
         REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1   H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
         INTEGER START,RANGE
C
         REAL TC(1000,2),TIMC(1000,2),DERIV(1000,3)
C
         REAL BE(2),AL(2,2),PSC,TGAS,CHISQ1,CHISQR,FLAMDA,TWF,
     1   A(2),B(2),ARRAY(2,2)
C
         INTEGER I,J,K,NTERMS,ERFLAG
C
C
         COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1   H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
         COMMON /BLK2/START,RANGE
C
         COMMON /BLK3/TC,TIMC,DERIV
C
C
C
C        AL = ALPHA MATRIX.
C        BE = BETA MATRIX.
C
 10      DO 20 J=1,NTERMS
         BE(J) = 0.
         DO 20 K=1,J
 20      AL(J,K) = 0.
C
C        TRUNCATE TGAS SINCE SMALL CHANGES IN TGAS
```

```
C        CAUSE UNNECESSARY ITERATION
C
         A(1) = AINT(TGAS)
         A(2) = PSC
 30      DO 60 I=START,RANGE
         DO 50 J=1,NTERMS
         BE(J) = BE(J)+(TWIRE(I)-DERIV(I,3))*DERIV(I,J)
         DO 40 K=1,J
 40      AL(J,K) = AL(J,K)+DERIV(I,J)*DERIV(I,K)
 50      CONTINUE
 60      CONTINUE
         DO 70 J=1,NTERMS
         DO 70 K=1,J
 70      AL(K,J) = AL(J,K)
C
C        EVALUATE CHI SQUARE AT STARTING POINT
C
         CHISQ1 = XICALC(A(1),A(2),ERFLAG,TWF)
C
 80      DO 100 J=1,NTERMS
         DO 90  K=1,NTERMS
C
C        CALCULATE ALPHA PRIME MATRIX (CALLED ARRAY)
C        AND ALSO INVERT IT.
C
         ARRAY(J,K)=AL(J,K)
 90      ARRAY(J,J)=ARRAY(J,J)*(1.+FLAMDA)
 100     CONTINUE
         CALL MATINV(ARRAY,NTERMS)
 110     B(2) = A(2)
         DO 130 J=1,NTERMS
         B(J) = A(J)
         DO 120 K=1,NTERMS
 120     B(J) = B(J) + BE(K)*ARRAY(J,K)
 130     CONTINUE
C
C        TRUNCATE B(1) & B(2) TO CONSIDER ONLY INTEGER VALUES
C        OF TEMPERATURE AND ONLY 2 SIGNIFICANT FIGURES FOR PSC.
C
         B(1) = AINT(B(1))
         B(2) = B(2)*100.
         B(2) = AINT(B(2))
         B(2) = B(2)/100.
C
C        CALCULATE CHISQR FOR NEW PARAMETER VALUES.
C
         CHISQR = XICALC(B(1),B(2),ERFLAG,TWF)
C
C        ERFLAG=6 IF ALPHA2 IS TOO LOW.
C
         IF (ERFLAG.EQ.6) GO TO 140
         IF (CHISQ1-CHISQR) 140,150,150
```

```
C
C       IF CHISQR INCREASED, INCREASE FLAMDA.
C
 140    FLAMDA = 10.0*FLAMDA
        GO TO 80
C
C       IF CHISQR DECREASED, DECREASE FLAMDA & SET NEW
C       VALUES FOR TGAS & PSC.
C
 150    TGAS = B(1)
        PSC = B(2)
        FLAMDA = FLAMDA/10.
 160    RETURN
        END
```

```
C
C       SUBROUTINE FDERIV(TGAS,PSC,NTERMS,TWF)
C
C
C       PURPOSE
C        COMPUTE THE DERIVATIVE OF TWIRE WITH RESPECT TO BOTH
C        TGAS & PSC AND ALSO EVALUATE THE THEORETICAL EQUATION.
C
C
C       TGAS = ESTIMATED GAS TEMPERATURE (K).
C       PSC = PROBE SHAPE CONSTANT.
C       NTERMS = NUMBER OF TERMS (1 OR 2).
C       TWF = ESTIMATED FINAL WIRE TEMPERATTURE (K).
C
C       COMMENTS
C        THIS PROGRAM COMPUTES THE DERIVATIVE OF TWIRE WITH
C        RESPECT TO TGAS AND STORES THE VALUES IN THE ARRAY
C        DERIV(I,1) WHERE I=1,1000. THE DERIVATIVE OF TWIRE
C        WITH RESPECT TO PSC IS STORED IN DERIV(I,2). THE
C        THEORETICAL EQUATION EVALUATED AT EACH MEASURED TIME
C        [MSTIME*(TIME INTEGER)] IS STORED IN DERIV(I,3).
C
C
C
        SUBROUTINE FDERIV(TGAS,PSC,NTERMS,TWF)
C
C
        REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        INTEGER START,RANGE
C
```

```
      REAL TC(1000,2),TIMC(1000,2),DERIV(1000,3)
C
      REAL DELTA(2),T,PC,TGAS,PSC,X,Y,Z,TWF
C
      INTEGER F,FF,ERFLAG,I,L1
C
      DATA DELTA/1.0,0.001/
C
      COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1 H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
      COMMON /BLK2/START,RANGE
C
      COMMON /BLK3/TC,TIMC,DERIV
C
C
C     COMPUTE DATA FOR DERIVATIVE OF TWIRE WITH
C     RESPECT TO TGAS.
C
C     COMPUTE DATA POINTS FOR T=TGAS+DELTA(1)
C     AND PC=PSC.
C
 10   T = TGAS + DELTA(1)
      PC = PSC
C
C     GENERATE CONSTANTS.
C
      CALL CONGEN(T,PC,ERFLAG,TWF)
      IF (ERFLAG.NE.0) STOP 997
C
C     GENERATE A TABLE OF COMPUTED TIMES AND TEMPERATURES.
C     TABL PUTS DATA INTO TC AND TIMC
C
      CALL TABL(1)
C
C     COMPUTE DATA POINTS FOR T=TGAS-DELTA(1)
C     AND PC=PSC.
C
 20   T = TGAS - DELTA(1)
      PC = PSC
C
C     GENERATE CONSTANTS.
C
      CALL CONGEN(T,PC,ERFLAG,TWF)
      IF (ERFLAG.NE.0) STOP 996
C
C     GENERATE ANOTHER TABLE.
      CALL TABL(2)
C
C     INTERPOLATE BETWEEN DATA POINTS SO THAT THE
C     CALCULATED TIMES (TIMC) CORRESPOND TO THE
C     MEASURED TIMES (I*MSTIME).
```

```
C
C       INTERPOLATE FOR T=TGAS+DELTA(1).
C
        CALL INTRP(1,1)
C
C       INTERPOLATE FOR T=TGAS-DELTA(1).
C
        CALL INTRP(2,1)
C
C       CALCULATE DERIVATIVE OF TWIRE WITH RESPECT TO
C       TGAS AND STORE IT IN DERIV(I,1).
C
 30     DERIV(START,1) = 0.
        L1 = START + 1
        DO 35 I=L1,RANGE
        DERIV(I,1)=(TC(I,1)-TC(I,2))/(2.*DELTA(1))
 35     CONTINUE
        IF (NTERMS.EQ.1) GO TO 60
C
C       COMPUTE DATA FOR DERIVATIVES OF TWIRE WITH
C       RESPECT TO PSC.
C
C       COMPUTE DATA POINTS FOR T=TGAS AND
C       PC=PSC+DELTA(2).
C
 40     T = TGAS
        PC = PSC + DELTA(2)
C
C       GENERATE CONSTANTS.
C
        CALL CONGEN(T,PC,ERFLAG,TWF)
        IF (ERFLAG.NE.0) STOP 995
C
C       GENERATE A TABLE OF COMPUTED TIMES AND TEMPERATURES.
C
        CALL TABL(1)
C
C       COMPUTE DATA POINTS FOR T=TGAS AND
C       PC=PSC-DELTA(2).
C
        T = TGAS
        PC = PSC - DELTA(2)
C
C       GENERATE CONSTANTS.
C
        CALL CONGEN(T,PC,ERFLAG,TWF)
        IF (ERFLAG.NE.0) STOP 994
C
C       GENERATE ANOTHER TABLE.
C
        CALL TABL(2)
```

```
C
C       INTERPOLATE BETWEEN DATA POINTS FOR PC=PSC+DELTA(2).
C
        CALL INTRP(1,2)
C
C       INTERPOLATE FOR PC=PSC-DELTA(2)
C
        CALL INTRP(2,2)
C
C       CALCULATE DERIVATIVE OF TWIRE WITH RESPECT
C       TO PSC AND STORE IN DERIV(I,2).
C
 50     DERIV(START,2) = 0.
        DO 55 I = L1,RANGE
        DERIV(I,2)=(TC(I,1)-TC(I,2))/(2.*DELTA(2))
 55     CONTINUE
C
C       GENERATE A TABLE OF ONLY THE FUNCTION (TWIRE VS. TIME).
C
C       GENERATE CONSTANTS FOR TGAS & PSC.
C
 60     CALL CONGEN(TGAS,PSC,ERFLAG,TWF)
        IF (ERFLAG.NE.0) STOP 993
C
C       COMPUTE A TABLE.
C
        CALL TABL(1)
C
C       INTERPOLATE BETWEEN DATA POINTS
C
        CALL INTRP(1,3)
C
C       STORE THE INTERPOLATED FUNCTION (TWIRE VS. TIME)
C       INTO DERIV(I,3).
C
 70     DO 75 I=1,1000
        DERIV(I,3) = TC(I,1)
 75     CONTINUE
        RETURN
        END




C
        FUNCTION XICALC(TGAS,PSC,ERFLAG,TWF)
C
C       PURPOSE
C        TO COMPUTE CHI SQUARE FOR PRESENT PARAMETER VALUES.
C
```

```
C          TGAS = ESTIMATED GAS TEMPERATURE (K).
C          PSC = PROBE SHAPE CONSTANT.
C          TWF = ESTIMATED FINAL WIRE TEMPERATURE (K).
C
C
C
C
C
C
        REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        INTEGER START,RANGE
C
        REAL TC(1000,2),TIMC(1000,2),DERIV(1000,3)
C
        REAL X,Y,TCM,ER
C
        INTEGER ERFLAG,J,I,L1,L2,K
C
C
        COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        COMMON /BLK2/START,RANGE
C
        COMMON /BLK3/TC,TIMC,DERIV
C
C
C       GENERATE CONSTANTS FOR THEORETICAL EQUATION.
C
 10     CALL CONGEN(TGAS,PSC,ERFLAG,TWF)
        IF (ERFLAG.NE.0) RETURN
C
C       GENERATE TABLE OF THEORETICAL DATA POINTS OF
C       TEMPERATURE VS. TIME. THE VALUES OF TIME ARE ONLY
C       APPROXIMATELY EQUAL TO THE ACTUAL MEASURED TIMES.
C
        CALL TABL(1)
        J = START
        L1 = START+1
        L2 = RANGE+1
        ER = 0.
C
C       INTERPOLATE SO THAT TIMES FOR THEORETICAL DATA
C       CORRESPOND TO DATA FOR ACTUAL MEASURED TIMES.
C
 55     DO 90 I=L1,RANGE
        X = MSTIME*FLOAT(I)
        DO 60 K=J,L2
        IF (X.LT.TIMC(K,1)) GO TO 70
 60     CONTINUE
```

```
 70     J = K-1
C
C       IF NO CHANGE IN TC(J,1) NO INTERPOLATION NECESSARY.
C
        IF (TC(K,1).EQ.TC(J,1)) GO TO 85
        TCM = TC(J,1)+(X-TIMC(J,1))*(TC(K,1)-TC(J,1))/
     1  (TIMC(K,1)-TIMC(J,1))
C
C       CALCULATE XI SQUARED. THIS IS INCLUDED INSIDE
C       INTERPOLATION LOOP FOR CONVIENENCE.
C
 80     ER = ER+(TWIRE(I)-TCM)**2
        GO TO 90
 85     TCM = TC(J,1)
        GO TO 80
 90     CONTINUE
        XICALC=ER
        RETURN
        END
```

```
C
        SUBROUTINE TABL(F)
C
C       PURPOSE
C        GENERATES A TABLE OF THEORETICAL TEMPERATURE VS.
C        TIME DATA POINTS. THE COMPUTED TIME CORRESPONDS
C        CLOSELY WITH THE MEASURED TIME BUT NOT EXACTLY.
C
C        F = SECOND INDEX ON TC AND TIMC. (F = 1 OR 2).
C
C
C
C
C
        REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        INTEGER START,RANGE
C
        REAL TC(1000,2),TIMC(1000,2),DERIV(1000,3)
C
        REAL DELTIM,DELTMP,TCALC,ZY
C
        INTEGER F,J,I,L1,L2
C
        COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
```

```
C
      COMMON /BLK2/START,RANGE
C
      COMMON /BLK3/TC,TIMC,DERIV
C
C
 100  TC(START,F) = TAVE
      TCALC = TAVE
C
C     MSTIME = ACTUAL TIME BETWEEN MEASURED DATA POINTS
C              OF TWIRE.
C
      DELTIM = MSTIME
C
C
C     FIX FIRST POINT.
C
      TIMC(START,F) = EVALTM(TCALC)
C
C     INITIALIZE POINT COUNTER FOR ACTUAL MEASURED TIMES.
C
      J = START
      Z = ALPHA2-0.01
C
C     COMPUTE TABLE OF TEMPERATURES
C
 110  L1 = START+1
      L2 = RANGE+1
      DO 140 I=L1,L2
C
C     COMPUTE TCALC AT POINT I - 1.
C
C     IF TCALC > Z WE ARE ON TOP FLAT PORTION OF CURVE
C     AND TEMPERATURE WILL NOT CHANGE ANY MORE.
C
      IF (TCALC.GT.Z) GO TO 130
C
C     COMPUTE DERIVATIVE OF TIME WITH RESPECT TO
C     TEMPERATURE FOR THEORETICAL CURVE.
C
      Y = -H1/(ALPHA2-TCALC) + H2/(TCALC-ALPHA3) +
     1    2.0*(H3A*(TCALC-ALPHA1)-H3B*BETA)/
     1    ((TCALC-ALPHA1)**2+BETA**2)
C
C     DELTIM = CHANGE IN TIME FROM THE LAST DATA NECESSARY
C     TO MAKE THE CURRENT DATA POINT FALL APPROXIMATELY ON
C     THE ACTUAL MEASURED TIME.
C     DELTMP = THE CORRESPONDING CHANGE IN TEMPERATURE.
C     NOTE THAT THE THEORETICAL FUNCTION (EVALTM) GIVES
C     TIME BACK FOR AN INPUT OF TEMPERATURE.
C
```

```
      DELTMP = DELTIM/Y
      TCALC = TCALC+DELTMP
      TC(I,F) = TCALC
      TIMC(I,F) = EVALTM(TCALC)
C
C     DETERMINE THE CHANGE IN TIME NECESSARY FOR THE NEXT
C     CALCULATED DATA POINT TO FALL ON THE NEXT MEASURED
C     DATA POINT.
C
 120  J = J+1
      DELTIM = MSTIME*FLOAT(J+1)-TIMC(I,F)
      IF (DELTIM.LE.0.0) GO TO 120
      GO TO 140
C
C
 130  J = J+1
      TC(I,F) = ALPHA2
      TIMC(I,F) = MSTIME*FLOAT(J)
 140  CONTINUE
      RETURN
      END
```

```
C
C     SUBROUTINE INTRP(F,FF)
C
C
C     PURPOSE
C      TO INTERPOLATE BETWEEN DATA POINTS CALCULATED FROM
C      THE THEORETICAL EQUATION. USED BY SUBROUTINE FDERIV.
C
C
C     F = SPECIFIES THE SECOND INDEX (1 OR 2) FOR TC AND TIME.
C     FF = SPECIFIES THE SECOND INDEX (1,2, OR 3) FOR DERIV.
C
C     UPON RETURN TC(I,F) HAS THE INTERPOLATED VALUES.
C
C
      SUBROUTINE INTRP(F,FF)
C
C
      REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1 H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
      INTEGER START,RANGE
C
      REAL TC(1000,2),TIMC(1000,2),DERIV(1000,3)
C
```

```
      REAL X
C
      INTEGER F,FF,J,I,K,L1
C
      COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1 H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
      COMMON /BLK2/START,RANGE
C
      COMMON /BLK3/TC,TIMC,DERIV
C
C
 10   L1 = START+1
C
C     DERIV(I,FF) HAS NOT BEEN USED YET AND
C     IS USED AS TEMPORARY STORAGE
C
 160  DO 170 I=1,1000
 170  DERIV(I,FF) = TC(I,F)
      J = START
 175  DO 200 I=L1,RANGE
      X = MSTIME*FLOAT(I)
      K = J-1
 180  K = K+1
      IF (X.GE.TIMC(K,F)) GO TO 180
      J = K-1
      IF (DERIV(K,FF).EQ.DERIV(J,FF)) GO TO 190
C
      TC(I,F) = DERIV(J,FF) + (X-TIMC(J,F))*
     1          (DERIV(K,FF)-DERIV(J,FF))/
     1           (TIMC(K,F)-TIMC(J,F))
C
      GO TO 200
 190  TC(I,F) = DERIV(J,FF)
 200  CONTINUE
      RETURN
      END




C
C     SUBROUTINE STCFIT
C
C
C     PURPOSE
C      TO CALCULATE THE STARTING LOCATION OF THE RAMP. THIS
C      LOCATION IS CALLED "START".
C
```

```
C        INPUT IS TEMPERATURE DATA "TWIRE". THE DATA POINTS
C        I=100 TO 160 WILL BE CURVE FIT TO A STRAIGHT LINE
C        Y = M*X + B WHERE Y=TWIRE(I) AND X=I=TIME.
C
C
         SUBROUTINE STCFIT
C
C        THIS ROUTINE CALCULATES THE STARTING LOCATION
C        OF THE RAMP.
C
         REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1   H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
         INTEGER START,RANGE
C
         INTEGER I,J,K,IX
         REAL SUMX,SUMY,SUMXX,SUMYY,SUMXY,B,M,X
C
         COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1   H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
         COMMON /BLK2/START,RANGE
C
C
 10      START = 100
C
C        USE K TO INCREMENT STARTING INDEX FOR TWIRE FROM
C        100 TO 130.
C
         DO 50 K=100,130
C
C        INITIALIZE VARIABLES
C
         SUMX = 0.
         SUMY = 0.
         SUMXX = 0.
         SUMYY = 0.
         SUMXY = 0.
         J = 0
C
C        PERFORM STANDARD LEAST SQUARES CURVE FIT TO A
C        STRAIGHT LINE.
C
 20      DO 30 I=K,160
         J = J+1
         SUMXY = SUMXY+(FLOAT(I))*TWIRE(I)
         SUMY = SUMY+TWIRE(I)
         SUMX = SUMX+FLOAT(I)
 30      SUMXX = SUMXX+(FLOAT(I))**2
C
C        CURVE FIT DATA TO   Y = M*X + B
C
```

```
      M=((FLOAT(J))*SUMXY-SUMX*SUMY)/((FLOAT(J))*SUMXX-SUMX*SUMX)
      B = (SUMY-M*SUMX)/(FLOAT(J))
C
C     STRAIGHT LINE CURVE FIT DONE.
C
C
C     DETERMINE IF LAST X WAS A MAXIMUM?
C     IF NO CONTINUE.
C     IF YES PROGRAM IS DONE AND "START" IS SET EQUAL
C     TO X AS THE STARTING LOCATION OF THE RAMP.
C
C     SET Y = TAVE
C
 40   X = (TAVE-B)/M
      IX = IFIX(X)
      IF (IX.GT.START) START = IX
 50   CONTINUE
      RETURN
      END




C
C     SUBROUTINE CONGEN(TGAS,PSC,ERFLAG,TWF)
C
C     PURPOSE
C      THIS PROGRAM GENERATES THE CONSTANTS NEEDED FOR THE
C      THEORETICAL TIME VS. TEMPERATURE EQUATION (EQU. 24).
C
C      TGAS = ESTIMATED GAS TEMPERATURE (K).
C      PSC = PROBE SHAPE CONSTANT.
C      ERFLAG = AN ERROR FLAG => SET=0 IF NO ERROR.
C      TWF = ESTIMATED FINAL WIRE TEMPERATURE.
C
C
      SUBROUTINE CONGEN(TGAS,PSC,ERFLAG,TWF)
C
C
      REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1 H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
      INTEGER START,RANGE
C
      REAL NU,K1,K2,K3,PSC,WDIA,WDENS,SPHT,SIGMA,
     1 E1,E2,EGAS,ALPHAG,X,AZ,E,BZ,F,Y
C
      DATA WDIA/0.8128E-3/, WDENS/0.20785E+5/, SPHT/0.1427E+3/,
     1 SIGMA/0.56697E-7/, E1/0.85E-1/, E2/0.76E-4/,
     1 EGAS/0.0/, ALPHAG/0.0/
```

```
C
C       TEMPERATURE = DEG. K.
C       WDIA = WIRE DIAMETER (METERS).
C       WDENS = WIRE DENSITY (Kgm/m**3).
C       SPHT = WIRE SPECIFIC HEAT (J/(Kgm,K)).
C       SIGMA = STEPHAN BOLTZMAN CONSTANT (J/(SEC.,K**4,m**2)).
C       EMISSIVITY OF WIRE = E1+E2*TWF. NO UNITS ON E1.
C                                     E2 HAS UNITS OF 1/DEG. K.
C       EGAS = EMISSIVITY OF GAS.
C       ALPHAG = ABSORPTIVITY OF GAS.
C       P = PRESSURE (PASCAL).
C
C
        INTEGER ERFLAG
C
C       AZ,BZ,E,F ARE TEMPORARY VARIABLES.
C       AZ & E ARE AT THE SAME LOCATION TO SAVE SPACE.
C
        EQUIVALENCE (AZ,E),(BZ,F)
C
        COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
        COMMON /BLK2/START,RANGE
C
C
 10     ERFLAG = 0
        MNN = MN
C
C       THE FOLLOWING STATEMENT IS NEEDED ONLY
C       FOR THE EXAMPLE IN THIS REPORT.
C       THE MACH NUMBER WAS MEASURED DOWN STREAM OF THE PULSED
C       THERMOCOUPLE SITE WHERE THE GAS WAS COOLER. THIS NEXT
C       STATEMENT CONVERTS THE MACH NUMBER AT THE LOWER
C       TEMPERATURE TO THAT AT THE PULSED THERMOCOUPLE SITE.
C
        MNN = MN*SQRT(TGAS/MTMP)
C
C       COMPUTE NUSSELT NUMBER.
C
 15     NU = 188.41*(SQRT(MNN*P*WDIA))/
     1  ((TGAS**0.6)*((1.+.2*MNN**2)**.25))
C
        K1 = WDIA*WDENS*SPHT/(4.*SIGMA*(E1+E2*TWF))
C
        K2 = (TGAS**.78)*NU*PSC*3.007E-4/
     1  (WDIA*SIGMA*(E1+E2*TWF))
C
        K3 = K2*TGAS+(1.-ALPHAG)*(TDUCT**4)+EGAS*(TGAS**4)
C
C
C       COMPUTE ALPHA1,ALPHA2,ALPHA3,BETA,H1,H2,H3A,H3B.
```

```
C
C
C        SCALE NUMBERS DOWN BY A FACTOR OF 10**-20 TO
C        PREVENT OVERFLOW.
C
 20      AZ = (K2**2)*(1.0E-20)*(K2**2)/4. +
     1   K3*(64.0E-20)*(K3**2)/27.0
C
         BZ = AZ
         Y = 1./3.
         AZ = (SQRT(AZ)*(1.0E+10) + (K2**2)/2.)**Y
         BZ = (K2**2)/2.0 - SQRT(BZ)*(1.0E+10)
         X = ABS(BZ)
C
C        ERROR IF BZ IS POSITIVE
C
         IF (X.EQ.BZ) STOP 2
         BZ = X**Y
C
C        EVALUATE ALPHA'S AND BETA.
C
         Y = AZ-BZ
         ALPHA1 = SQRT(Y)/2.
         BETA = Y+2.*K2/SQRT(Y)
         BETA = SQRT(BETA)/2.
         X = SQRT(2.*K2/SQRT(Y)-Y)
         ALPHA2 = -SQRT(Y)/2.+X/2.
         ALPHA3 = -SQRT(Y)/2.-X/2.
C
         H1 = -K1/(((ALPHA2-ALPHA1)**2+BETA**2)*(ALPHA2-ALPHA3))
         H2 = -K1/(((ALPHA3-ALPHA1)**2+BETA**2)*(ALPHA3-ALPHA2))
         E = -2.*(BETA**2)*(2.*ALPHA1-ALPHA2-ALPHA3)
         F = 2.*(BETA*(ALPHA1-ALPHA2)*(ALPHA1-ALPHA3)-BETA**3)
         X = E**2+F**2
         H3A = -K1*E/X
         H3B = K1*F/X
C
C        TIME = H1*ALOG(ALPHA2-T) + H2*ALOG(T-ALPHA3) +
C               H3A*ALOG((T-ALPHA1)**2+BETA**2) +
C               2.0*H3B*ATAN(BETA/(T-ALPHA1)) + H4
C
C        H4 = CONSTANT TO BE DETERMINED.
C        TAVE = AVERAGE INITIAL WIRE TEMPERATURE.
C        MSTIME IS TIME SCALE FACTOR.
C        MSTIME*(TIME INTEGER) IS TIME SINCE START OF DATA.
C        TIME = 0 AT DATA POINT I=0
C        TIME = MSTIME AT DATA POINT I=1 etc.
C
C        SET ERROR FLAG IF ALPHA2 IS LESS THAN TAVE SINCE IT
C        WOULD REQUIRE TAKING THE LOG OF A NEGATIVE NUMBER.
C
 30      IF (ALPHA2.GT.TAVE) GO TO 40
```

```
      ERFLAG = 6
      RETURN
C
 40   X = TAVE-ALPHA1
C
C     COMPUTE INTEGRATION CONSTANT H4 BY SETTING T=TAVE
C     AT THE STARTING TIME AND SOLVING FOR H4.
C
      H4 = MSTIME*FLOAT(START) - (H1*ALOG(ALPHA2-TAVE) +
     1  H2*ALOG(TAVE-ALPHA3) +
     1  H3A*ALOG((TAVE-ALPHA1)**2+BETA**2) +
     1  2.0*H3B*ATAN2(BETA,X))
C
      RETURN
      END
```

```
C
C     FUNCTION EVALTM(T)
C
C
C     PURPOSE
C      EVALUATE THE THEORETICAL EQUATION (TEXT EQU. 24)
C      FOR TIME AS A FUNCTION OF TEMPERATURE.
C
C     T = INPUT TEMPERATURE (K).
C
C
      FUNCTION EVALTM(T)
C
C
      REAL TWIRE(1000),TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
      REAL T,X
C
      COMMON /BLK1/TWIRE,TAVE,ALPHA1,ALPHA2,ALPHA3,BETA,
     1  H1,H2,H3A,H3B,H4,TDUCT,MSTIME,MN,P,MNN,MTMP
C
C
 10   X = T-ALPHA1
      EVALTM = H1*ALOG(ALPHA2-T)+
     1  H2*ALOG(T-ALPHA3)+
     1  H3A*ALOG((T-ALPHA1)**2+BETA**2)+
     1  2.0*H3B*ATAN2(BETA,X)+H4
      RETURN
      END
```

```
C
C       SUBROUTINE MATINV
C
C       PURPOSE:
C               INVERT A 1 OR 2 DEGREE MATRIX.
C
        SUBROUTINE MATINV (ARRAY,NORDER)
C
C       ARRAY = INPUT MATRIX WHICH IS REPLACED BY ITS INVERSE.
C       NORDER= DEGREE OF MATRIX.
C
        REAL ARRAY(2,2),DET,X
        INTEGER NORDER,I,J
C
 10     IF (NORDER.EQ.1) GO TO 20
        IF (NORDER.EQ.2) GO TO 30
        STOP 800
C
C       CALCULATE INVERSE OF ONE DEGREE MATRIX.
C
 20     ARRAY(1,1) = 1./ARRAY(1,1)
        RETURN
C
C       CALCULATE DETERMINANT FOR SECOND DEGREE MATRIX.
C
 30     DET = ARRAY(1,1)*ARRAY(2,2)-ARRAY(1,2)*ARRAY(2,1)
        IF (DET.EQ.0) STOP 801
C
C       CALCULATE ADJOINT MATRIX
C
        X = ARRAY(1,1)
        ARRAY(1,1) = ARRAY(2,2)
        ARRAY(2,2) = X
        ARRAY(1,2) = -ARRAY(1,2)
        ARRAY(2,1) = -ARRAY(2,1)
C
C       CALCULATE THE INVERSE OF SECOND DEGREE MATRIX.
C
        DO 50 I=1,2
        DO 40 J=1,2
        ARRAY(I,J) = ARRAY(I,J)/DET
 40     CONTINUE
 50     CONTINUE
C
C
        RETURN
        END
```

Appendix C
Gradient-Expansion Method

This appendix describes the least-squares fit to a nonlinear function that uses the gradient-expansion algorithm taken from Bevington (ref. 2). The objective of the process is to search for the values of parameters in the theoretical equation that will minimize the sum of the squares of the difference between the data points and the theoretical nonlinear function. This sum to be minimized is defined as

$$\chi^2 = \sum_{i=1}^{m} \left[Y_i - Y(X_i)\right]^2 \tag{C1}$$

where m is the number of data points, Y_i is the dependent variable, X_i is the independent variable, and $Y(X)$ is the theoretical function with unknown parameters a_j.

The quantity χ^2 is regarded as a function of the parameters a_j of the fitting function $Y(X)$. There are m data points (X_i, Y_i). The idea is to choose the values of the n parameters a_j so that χ^2 is a minimum.

The first approach is to take the gradient of χ^2

$$\nabla\chi^2 = \sum_{j=1}^{n} \frac{\partial\chi^2}{\partial a_j}\hat{a}_j \tag{C2}$$

where the $\hat{a}_j$ are unit vectors. The gradient of χ^2 gives the direction of the maximum rate of increase of χ^2. We want to increment the parameters from some starting value χ_0^2 so that χ^2 decreases. Hence we write

$$\delta a_j = -(\nabla\chi_0^2)_j\, \Delta a_j$$
$$= -\left(\frac{\partial\chi_0^2}{\partial a_j}\right)\Delta a_j \tag{C3}$$

The Δa_j are size constants that must be supplied. The parameters a_j are incremented by δa_j and the process repeated. The minus sign insures that the increments are in a direction opposite to the gradient so that they are in the direction of most rapid decrease of χ^2. However, the method tends not to work well near the actual minimum—it is better further away.

Another approach is to expand the fitting function $Y(x)$ as a first-order Taylor series in the parameters

$$Y(X) = Y_0(X) + \sum_{j=1}^{n} \frac{\partial Y_0(X)}{\partial a_j}\delta a_j \tag{C4}$$

where $Y_0(X)$ is the value of $Y(X)$ at the starting point for the expansion. Then

$$\chi^2 = \sum_{i=1}^{m} \left[Y_i - Y_0(X_i) - \sum_{j=1}^{n} \frac{\partial Y_0(X_i)}{\partial a_j}\delta a_j\right]^2 \tag{C5}$$

We now want to minimize χ^2 as a function of the increments δa_j; so we take $\partial\chi^2/\partial\delta a_k$ and set it equal to zero

$$\sum_{i=1}^{m} 2\left[Y_i - Y_0(X_i)\right]\frac{\partial Y_0(X_i)}{\partial a_k} = \sum_{j=1}^{n} \delta a_j \sum_{i=1}^{m} 2\frac{\partial Y_0(X_i)}{\partial a_j}\frac{\partial Y_0(X_i)}{\partial a_k} \tag{C6}$$

This gives a set of n linear equations for the n quantities δa_j. Define

$$\beta_k \equiv -\frac{1}{2}\frac{\partial\chi_0^2}{\partial a_k} = \sum_{i=1}^{m} |Y_i - Y(X_i)| \frac{\partial Y_0(X_i)}{\partial a_k} \tag{C7}$$

$$\alpha_{jk} = \sum_{i=1}^{m} \frac{\partial Y_0(Y_i)}{\partial a_j}\frac{\partial Y_0(X_i)}{\partial a_k} \tag{C8}$$

and

$$\chi_0^2 = \sum_{i=1}^{m} |Y_i - Y_0(X_i)|^2 \tag{C9}$$

thus

$$\beta_k = \sum_{j=1}^{n} \delta a_j \alpha_{jk} \qquad k = 1,2,\ldots,n \tag{C10}$$

This can be put into the form of a matrix equation

$$\boldsymbol{\beta} = \boldsymbol{\delta a} \cdot \boldsymbol{\alpha}$$

or (C11)

$$\boldsymbol{\beta} \cdot \boldsymbol{\alpha}^{-1} = \delta \mathbf{a}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\delta a}$ are column matrices with n elements and $\boldsymbol{\alpha}$ is an n-by-n symmetric square matrix. This method tends to work well near the actual minimum but poorly far from the minimum.

By combining the two methods it is possible to obtain an algorithm that works well far from the minimum and also close to it. To combine the two methods, one writes (ref. 7)

$$\boldsymbol{\beta} = \boldsymbol{\alpha}' \cdot \boldsymbol{\delta a} \tag{C12}$$

where

$$\alpha'_{jk} = \alpha_{jk} \qquad \text{for} \qquad j \neq k \tag{C13}$$

and

$$\alpha'_{jj} = \alpha_{jj}(1 + \lambda) \qquad \text{for } \lambda \geq 0 \tag{C14}$$

where λ is an arbitrary parameter that changes the method from the Taylor series to the gradient method. If λ is near zero, the method is the same as the Taylor series approach. If λ is large, the diagonal terms dominate and the equations are essentially

$$\beta_j = \lambda \delta a_j \alpha_{jj}$$

or

$$\delta a_j = \frac{1}{\lambda \alpha_{jj}} \beta_j = -\frac{1}{2\lambda \alpha_{jj}} \frac{\partial \chi_0^2}{\partial a_j} \tag{C15}$$

$$= \frac{-1}{2\lambda \alpha_{jj}} (\nabla \chi_0^2)_j$$

which result in the gradient method.

This technique can be used by starting with an arbitrary small value of λ, such as 0.001. If the computed δa_j causes χ^2 to increase instead of decrease, the initial guess at the a_j is not good enough, and χ^2 is too far from the minimum for the second method to work. Then λ is increased by a factor of 10 and a new set of δa_j is found. Each time λ is increased the algorithm is more like just taking the gradient, which works well for a_j far from $(a_j)_{min}$. This continues until χ^2 starts to decrease, at which time λ is divided by 10 at each iteration. By this time the minimum will have been found.

Appendix D

Typical Program Input and Results

This appendix provides an example of data used by the computer program. The following data were put into the computer program:

INPUT MACH NUMBER 0.0286
INPUT PRESSURE IN Pa. 99805.
INPUT DUCT TEMPERATURE IN DEG. K. 396.0
INPUT PROBE SHAPE CONSTANT 0.0
MACH NUMBER TEMPERATURE DEG. K. 415.8

The following data were put out by the computer program:

GAS TEMPERATURE = 1707.00 K
FINAL WIRE TEMPERATURE = 1565.79 K
PROBE SHAPE CONSTANT = 0.850

The following data were not put out by the computer program but may be useful:

CHISQR = 0.267E+05
TAVE = 677.0
ALPHA1 = 1183.2
ALPHA2 = 1565.8
ALPHA3 = −3932.2
BETA = 3218.3
H1 = −0.902
H2 = 0.259
H3A = 0.321
H3B = −0.260
H4 = 0.072

The 1000 data points of thermocouple wire temperature are shown in the following listing:

THERMOCOUPLE TEMPERATURE DATA (K)

I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)
1	674.7	51	676.7	101	678.6	151	824.3
2	675.7	52	674.7	102	678.6	152	827.9
3	676.7	53	676.7	103	677.6	153	828.8
4	678.6	54	675.7	104	678.6	154	835.1
5	675.7	55	673.8	105	678.6	155	835.1
6	677.6	56	674.7	106	681.5	156	837.8
7	677.6	57	672.8	107	682.4	157	842.2
8	679.5	58	675.7	108	684.3	158	842.2
9	679.5	59	671.8	109	687.2	159	844.0
10	676.7	60	671.8	110	692.0	160	851.1
11	675.7	61	673.8	111	694.8	161	854.7
12	675.7	62	673.8	112	701.5	162	856.5
13	675.7	63	673.8	113	703.4	163	861.8
14	674.7	64	672.8	114	708.2	164	866.2
15	677.6	65	671.8	115	713.8	165	867.1
16	676.7	66	669.9	116	715.7	166	869.7
17	673.8	67	670.9	117	717.6	167	871.5
18	675.7	68	674.7	118	723.3	168	877.7
19	672.8	69	673.8	119	726.1	169	878.5
20	672.8	70	674.7	120	730.8	170	882.1
21	670.9	71	675.7	121	729.8	171	886.5
22	671.8	72	676.7	122	734.5	172	889.1
23	671.8	73	677.6	123	738.2	173	890.8
24	671.8	74	677.6	124	738.2	174	892.6
25	671.8	75	680.5	125	741.0	175	896.1
26	671.8	76	679.5	126	744.8	176	898.7
27	672.8	77	680.5	127	746.6	177	902.2
28	669.9	78	679.5	128	750.3	178	903.1
29	669.9	79	680.5	129	752.2	179	906.5
30	671.8	80	680.5	130	754.0	180	909.2
31	672.8	81	680.5	131	757.8	181	912.6
32	671.8	82	684.3	132	762.4	182	916.1
33	673.8	83	684.3	133	765.2	183	916.1
34	673.8	84	683.4	134	769.8	184	918.7
35	678.6	85	683.4	135	773.4	185	920.4
36	678.6	86	682.4	136	775.3	186	922.1
37	678.6	87	686.3	137	779.8	187	925.6
38	680.5	88	685.3	138	783.5	188	928.2
39	682.4	89	682.4	139	787.2	189	928.2
40	680.5	90	679.5	140	789.9	190	931.6
41	679.5	91	678.6	141	790.8	191	934.2
42	682.4	92	682.4	142	795.4	192	935.0
43	681.5	93	679.5	143	799.0	193	935.0
44	678.6	94	681.5	144	801.7	194	937.6
45	682.4	95	678.6	145	807.2	195	940.2
46	683.4	96	677.6	146	809.9	196	944.5
47	682.4	97	678.6	147	812.6	197	944.5
48	681.5	98	679.5	148	814.4	198	947.9
49	678.6	99	678.6	149	816.2	199	950.4
50	676.7	100	678.6	150	821.6	200	953.0

THERMOCOUPLE TEMPERATURE DATA (K)

I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)
201	957.3	251	1070.3	301	1176.3	351	1256.5
202	959.8	252	1076.0	302	1178.6	352	1258.0
203	960.7	253	1073.5	303	1181.8	353	1259.5
204	964.9	254	1077.6	304	1181.8	354	1259.5
205	964.9	255	1079.2	305	1182.6	355	1260.3
206	967.4	256	1080.8	306	1183.3	356	1261.1
207	973.4	257	1083.3	307	1187.2	357	1264.1
208	974.2	258	1086.5	308	1188.0	358	1262.6
209	975.9	259	1088.1	309	1189.6	359	1264.1
210	977.6	260	1090.5	310	1192.7	360	1264.8
211	981.8	261	1093.8	311	1191.9	361	1264.8
212	981.8	262	1093.8	312	1193.4	362	1267.1
213	983.5	263	1097.8	313	1195.8	363	1268.6
214	986.0	264	1099.4	314	1198.9	364	1270.2
215	986.8	265	1103.4	315	1199.7	365	1270.9
216	989.3	266	1104.2	316	1202.8	366	1271.7
217	989.3	267	1106.6	317	1202.8	367	1272.4
218	991.9	268	1109.0	318	1205.9	368	1274.7
219	994.4	269	1110.6	319	1205.9	369	1277.0
220	996.9	270	1114.6	320	1208.2	370	1276.2
221	997.7	271	1117.0	321	1208.9	371	1277.7
222	999.4	272	1117.8	322	1211.3	372	1280.0
223	1004.4	273	1121.0	323	1213.6	373	1282.3
224	1006.1	274	1121.8	324	1215.1	374	1282.3
225	1009.4	275	1124.2	325	1216.7	375	1284.5
226	1010.3	276	1128.2	326	1217.4	376	1286.8
227	1014.4	277	1129.8	327	1218.2	377	1287.5
228	1016.9	278	1132.2	328	1219.8	378	1288.3
229	1019.4	279	1135.3	329	1218.2	379	1288.3
230	1021.9	280	1136.9	330	1220.5	380	1289.8
231	1026.0	281	1137.7	331	1222.8	381	1290.5
232	1028.5	282	1140.1	332	1225.9	382	1292.0
233	1029.3	283	1143.3	333	1225.9	383	1294.3
234	1032.6	284	1146.4	334	1229.0	384	1294.3
235	1036.7	285	1145.6	335	1229.8	385	1297.3
236	1039.2	286	1148.8	336	1231.3	386	1297.3
237	1040.8	287	1151.2	337	1232.1	387	1298.8
238	1040.8	288	1151.9	338	1234.3	388	1301.1
239	1045.0	289	1153.5	339	1236.7	389	1301.8
240	1046.6	290	1156.7	340	1238.2	390	1301.1
241	1049.1	291	1158.3	341	1238.2	391	1303.3
242	1049.9	292	1162.2	342	1242.0	392	1303.3
243	1053.2	293	1162.2	343	1242.0	393	1304.0
244	1054.8	294	1164.6	344	1245.8	394	1306.3
245	1055.6	295	1166.1	345	1248.9	395	1305.5
246	1058.9	296	1168.5	346	1247.3	396	1305.5
247	1059.7	297	1168.5	347	1250.4	397	1307.8
248	1064.6	298	1170.0	348	1251.9	398	1309.3
249	1065.4	299	1173.2	349	1253.4	399	1312.3
250	1068.7	300	1173.9	350	1255.0	400	1312.3

THERMOCOUPLE TEMPERATURE DATA (K)

I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)
401	1312.3	451	1370.0	501	1398.6	551	1434.2
402	1312.3	452	1370.8	502	1397.8	552	1435.6
403	1316.8	453	1372.2	503	1400.8	553	1436.3
404	1317.5	454	1373.7	504	1400.8	554	1436.3
405	1318.2	455	1374.4	505	1399.3	555	1436.3
406	1321.2	456	1375.2	506	1402.2	556	1437.8
407	1322.7	457	1375.2	507	1403.7	557	1437.8
408	1324.2	458	1375.9	508	1403.7	558	1438.5
409	1324.9	459	1375.9	509	1405.1	559	1438.5
410	1325.7	460	1378.8	510	1404.4	560	1440.0
411	1327.2	461	1379.6	511	1405.8	561	1440.7
412	1328.7	462	1380.3	512	1405.8	562	1441.4
413	1329.4	463	1381.8	513	1408.0	563	1442.8
414	1330.1	464	1381.0	514	1408.8	564	1444.3
415	1332.4	465	1383.2	515	1408.0	565	1445.0
416	1333.8	466	1381.8	516	1411.7	566	1443.6
417	1333.8	467	1383.9	517	1410.9	567	1445.0
418	1335.3	468	1383.2	518	1411.7	568	1446.5
419	1336.8	469	1383.9	519	1412.4	569	1445.7
420	1338.3	470	1384.7	520	1413.9	570	1445.0
421	1339.8	471	1383.9	521	1413.9	571	1442.1
422	1342.0	472	1383.9	522	1416.8	572	1443.6
423	1342.0	473	1384.7	523	1413.9	573	1445.0
424	1343.5	474	1384.7	524	1418.2	574	1443.6
425	1345.7	475	1384.7	525	1418.2	575	1445.7
426	1346.4	476	1386.9	526	1417.5	576	1446.5
427	1347.2	477	1386.9	527	1416.8	577	1446.5
428	1347.9	478	1386.1	528	1418.2	578	1447.2
429	1348.7	479	1388.3	529	1417.5	579	1447.2
430	1348.7	480	1389.8	530	1415.3	580	1447.9
431	1353.8	481	1388.3	531	1416.8	581	1448.6
432	1353.8	482	1387.6	532	1417.5	582	1448.6
433	1353.8	483	1389.1	533	1420.4	583	1450.8
434	1353.1	484	1390.5	534	1419.7	584	1450.8
435	1355.3	485	1392.0	535	1420.4	585	1451.5
436	1355.3	486	1392.7	536	1421.1	586	1452.2
437	1358.3	487	1392.0	537	1422.6	587	1454.4
438	1359.0	488	1394.2	538	1422.6	588	1454.4
439	1358.3	489	1395.6	539	1424.0	589	1453.7
440	1358.3	490	1396.4	540	1425.5	590	1455.1
441	1359.7	491	1394.2	541	1426.2	591	1455.1
442	1360.5	492	1395.6	542	1426.9	592	1455.8
443	1362.7	493	1396.4	543	1427.7	593	1457.3
444	1363.4	494	1396.4	544	1427.7	594	1457.3
445	1365.6	495	1397.1	545	1428.4	595	1456.6
446	1364.9	496	1395.6	546	1431.3	596	1455.1
447	1367.8	497	1397.8	547	1431.3	597	1456.6
448	1369.3	498	1396.4	548	1431.3	598	1458.0
449	1368.6	499	1398.6	549	1434.2	599	1458.7
450	1369.3	500	1397.8	550	1432.7	600	1458.7

THERMOCOUPLE TEMPERATURE DATA (K)

I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)
601	1460.2	651	1478.9	701	1498.3	751	1516.1
602	1459.4	652	1477.4	702	1498.3	752	1516.8
603	1460.2	653	1478.9	703	1499.7	753	1514.7
604	1460.9	654	1478.9	704	1500.4	754	1516.1
605	1462.3	655	1477.4	705	1501.8	755	1516.1
606	1463.1	656	1480.3	706	1499.7	756	1516.1
607	1463.1	657	1480.3	707	1503.3	757	1516.8
608	1463.1	658	1480.3	708	1501.8	758	1518.3
609	1463.1	659	1478.9	709	1503.3	759	1518.3
610	1463.8	660	1481.0	710	1502.6	760	1519.0
611	1464.5	661	1479.6	711	1503.3	761	1519.7
612	1466.7	662	1481.8	712	1504.0	762	1521.1
613	1465.9	663	1482.5	713	1504.7	763	1519.7
614	1468.1	664	1482.5	714	1504.7	764	1520.4
615	1468.1	665	1483.9	715	1505.4	765	1520.4
616	1468.8	666	1482.5	716	1506.1	766	1521.8
617	1468.8	667	1483.2	717	1506.8	767	1521.8
618	1468.1	668	1484.6	718	1506.1	768	1520.4
619	1468.1	669	1484.6	719	1507.6	769	1523.3
620	1467.4	670	1485.3	720	1508.3	770	1522.6
621	1468.8	671	1484.6	721	1508.3	771	1523.3
622	1468.1	672	1486.8	722	1506.8	772	1523.3
623	1468.8	673	1487.5	723	1507.6	773	1524.7
624	1471.0	674	1486.8	724	1509.7	774	1524.0
625	1471.7	675	1487.5	725	1509.7	775	1525.4
626	1471.0	676	1488.9	726	1509.7	776	1525.4
627	1472.4	677	1489.7	727	1509.7	777	1525.4
628	1473.8	678	1489.7	728	1509.7	778	1524.7
629	1473.8	679	1490.4	729	1511.1	779	1524.7
630	1475.3	680	1490.4	730	1511.1	780	1524.7
631	1473.8	681	1491.8	731	1510.4	781	1526.1
632	1473.1	682	1490.4	732	1511.1	782	1526.1
633	1475.3	683	1492.5	733	1511.8	783	1526.8
634	1473.8	684	1492.5	734	1511.1	784	1528.3
635	1474.6	685	1492.5	735	1511.1	785	1527.6
636	1476.0	686	1493.2	736	1511.1	786	1529.0
637	1476.0	687	1493.2	737	1511.8	787	1530.4
638	1476.7	688	1496.1	738	1511.8	788	1530.4
639	1478.2	689	1495.4	739	1512.6	789	1528.3
640	1478.2	690	1494.7	740	1513.3	790	1529.7
641	1478.2	691	1496.8	741	1512.6	791	1529.7
642	1477.4	692	1498.3	742	1514.0	792	1530.4
643	1478.2	693	1497.5	743	1514.7	793	1529.7
644	1478.2	694	1497.5	744	1514.7	794	1531.1
645	1478.2	695	1498.3	745	1514.0	795	1531.8
646	1478.2	696	1497.5	746	1513.3	796	1531.1
647	1476.0	697	1498.3	747	1513.3	797	1527.6
648	1477.4	698	1499.0	748	1516.1	798	1531.1
649	1478.9	699	1498.3	749	1514.7	799	1532.6
650	1477.4	700	1499.0	750	1514.0	800	1533.3

THERMOCOUPLE TEMPERATURE DATA (K)

I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)	I	TWIRE(I)
801	1534.7	851	1536.1	901	1541.8	951	1543.3
802	1533.3	852	1536.1	902	1542.5	952	1543.3
803	1531.8	853	1537.6	903	1543.3	953	1543.3
804	1533.3	854	1536.1	904	1542.5	954	1544.7
805	1531.1	855	1539.0	905	1542.5	955	1543.3
806	1531.1	856	1539.0	906	1542.5	956	1543.3
807	1531.1	857	1536.8	907	1541.8	957	1544.7
808	1531.1	858	1534.7	908	1541.8	958	1544.7
809	1530.4	859	1534.7	909	1544.0	959	1544.7
810	1531.8	860	1534.7	910	1540.4	960	1545.4
811	1531.1	861	1535.4	911	1539.0	961	1544.7
812	1531.1	862	1536.1	912	1537.6	962	1544.7
813	1529.7	863	1536.1	913	1539.0	963	1544.7
814	1531.1	864	1536.1	914	1539.0	964	1544.7
815	1531.1	865	1536.8	915	1539.0	965	1545.4
816	1531.1	866	1537.6	916	1536.8	966	1545.4
817	1531.8	867	1538.3	917	1538.3	967	1547.5
818	1531.8	868	1538.3	918	1539.0	968	1544.0
819	1531.1	869	1537.6	919	1539.0	969	1544.7
820	1532.6	870	1539.7	920	1539.0	970	1544.7
821	1534.0	871	1539.0	921	1539.0	971	1544.0
822	1533.3	872	1535.4	922	1538.3	972	1545.4
823	1534.0	873	1537.6	923	1539.0	973	1544.0
824	1533.3	874	1536.8	924	1539.0	974	1544.7
825	1533.3	875	1537.6	925	1540.4	975	1543.3
826	1534.7	876	1537.6	926	1539.0	976	1544.0
827	1535.4	877	1537.6	927	1540.4	977	1545.4
828	1533.3	878	1538.3	928	1540.4	978	1543.3
829	1534.7	879	1536.8	929	1541.1	979	1542.5
830	1534.0	880	1537.6	930	1541.8	980	1542.5
831	1534.0	881	1539.0	931	1540.4	981	1541.1
832	1534.7	882	1538.3	932	1542.5	982	1539.7
833	1534.7	883	1538.3	933	1542.5	983	1539.0
834	1535.4	884	1539.7	934	1544.7	984	1539.0
835	1534.7	885	1539.7	935	1542.5	985	1539.0
836	1532.6	886	1537.6	936	1542.5	986	1539.7
837	1533.3	887	1537.6	937	1542.5	987	1539.7
838	1534.7	888	1539.0	938	1544.0	988	1539.0
839	1534.0	889	1539.7	939	1543.3	989	1539.0
840	1536.1	890	1539.7	940	1543.3	990	1539.0
841	1534.7	891	1541.1	941	1544.0	991	1537.6
842	1534.7	892	1540.4	942	1545.4	992	1538.3
843	1535.4	893	1541.1	943	1543.3	993	1539.0
844	1534.7	894	1540.4	944	1543.3	994	1539.7
845	1533.3	895	1539.7	945	1543.3	995	1539.0
846	1535.4	896	1541.1	946	1545.4	996	1540.4
847	1535.4	897	1541.8	947	1544.0	997	1541.1
848	1534.0	898	1541.1	948	1544.7	998	1539.7
849	1534.7	899	1541.8	949	1544.7	999	1540.4
850	1535.4	900	1541.8	950	1544.0	1000	1541.1

References

1. Glawe, George, E.; Will, Herbert A.; and Krause, Lloyd N.: A New Approach to the Pulsed Thermocouple for High Gas Temperature Measurements. NASA TM X-71883, 1976.

2. Bevington, Philip R.: Data Reduction and Error Analysis for the Physical Sciences. McGraw-Hill Book Co., Inc., 1969, p. 235.

3. Scadron, Marvin D.; and Warshawsky, Isidore: Experimental Determination of Time Constants and Nusselt Numbers for Bare-Wire Thermocouples in High-Velocity Air Streams and Analytic Approximation of Conduction and Radiation Errors. NACA TN-2599, 1952, p. 7.

4. Davis, H. T.: College Algebra. Prentice-Hall, Inc., 1942, pp. 281-282.

5. Bevington, Philip R.: Data Reduction and Error Analysis for the Physical Sciences. McGraw-Hill Book Co., Inc., 1969, p. 241.

6. Glawe, George E.; and Shepard, Charles E.: Some Effects of Exposure to Exhaust-Gas Streams on Emittance and Thermoelectric Power of Bare-Wire Platinum Rhodium-Platinum Thermocouples. NACA TN-3253, 1954, p. 29.

7. Scadron, Marvin D.; and Warshawsky, Isidore: Experimental Determination of Time Constants and Nusselt Numbers for Bare-Wire Thermocouples in High-Velocity Air Streams and Analytic Approximation of Conduction and Radiation Errors. NACA TN-2599, 1952, pp. 48 and 54.

8. Marquadt, D. W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters. J. Soc. Indust. Appl. Math., vol. 11, no. 2, June 1963, pp. 431-441.

1. Report No. NASA TP-1895	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle COMPUTER PROGRAM FOR PULSED THERMOCOUPLES WITH CORRECTIONS FOR RADIATION EFFECTS		5. Report Date September 1981
		6. Performing Organization Code 505-32-82
7. Author(s) Herbert A. Will		8. Performing Organization Report No. E-615
		10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Paper
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A pulsed thermocouple is used for measuring gas temperatures above the melting point of common thermocouples. This is done by allowing the thermocouple to heat until it approaches its melting point and then turning on the protective cooling gas. This method requires a computer to extrapolate the thermocouple data to the higher gas temperatures. In earlier work by this author the extrapolation was done by using a first-order exponential curve fit to predict the final thermocouple wire temperature. Since radiation effects were neglected, the gas temperature was not computed. Hand calculations had to be used to estimate the gas temperature. This report describes a method that includes the effect of radiation in the extrapolation. Computations of gas temperature are provided, along with the estimate of the final thermocouple wire temperature. Results from tests on high-temperature combustor research rigs are presented.

17. Key Words (Suggested by Author(s))		18. Distribution Statement	
Computer programs; Temperature measurement; Thermocouples; Jet engine; Extrapolation; Combustion temperature		Unclassified - unlimited STAR Category 61	
19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	41	A03

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1981

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

3 1 1U,G, 092481 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LABORATORY
ATTN: TECHNICAL LIBRARY (SUL)
KIRTLAND AFB NM 87117

NASA

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return