Grant No. NAG9 - 301

INVESTIGATION OF HEAT TRANSFER IN ZIRCONIUM POTASSIUM PERCHLORATE AT LOW TEMPERATURE:

A STUDY OF THE FAILURE MECHANISM OF THE NASA STANDARD INITIATOR

Final Technical Report

For the Period

August 1, 1988 - August 31, 1989

Submitted by

Dr. Philip L. Varghese

The University of Texas at Austin

Center for Energy Studies

Austin, TX 78712

A STUDY OF THE FAILURE MECHANISM OF THE NASA STANDARD INITIATOR

by

Dr. Philip L. Varghese

&

Walid A. El-katerji

The University of Texas at Austin

Center for Energy Studies

Austin, TX 78712

ACKNOWLEDGEMENTS

This grant, "Investigation of Heat Transfer in Zirconium Potassium Perchlorate at Low Temperature" (NAG9-301), is a continuation of Grant NAG9-201, "Investigation of Energy Transfer in the Ignition Mechanism of a NASA Standard Initiator". These grants were provided by the Lyndon B. Johnson Center of the National Aeronautics and Space Administration and monitored by Mr. Keith Van Tassel.

ABSTRACT

The objective of this work was to study the reasons for the failure of pyrotechnic initiators at very low temperatures (10 - 100 K). A two-dimensional model of the NASA standard initiator was constructed to model heat transfer from the electrically heated stainless steel bridgewire to the zirconium potassium perchlorate explosive charge and the alumina charge cup. Temperature dependent properties were used in the model to simulate initiator performance over a wide range of initial temperatures (10 K - 500 K). A search of the thermophysical property data base showed that pure alumina has a very high thermal conductivity at low temperatures. It had been assumed to act as a thermal insulator in all previous analyses. Rapid heat transfer from the bridgewire to the alumina at low initial temperatures was shown to cause failure of the initiators if the wire did not also make good contact with the zirconium potassium perchlorate charge.

The model is able to reproduce the results of the tests that had been conducted to investigate the cause for failure. It also provides an explanation for previously puzzling results and suggests simple design changes that will increase reliability at very low initial temperatures.

Table Of Contents

List Of Figures

List of Tables

List of Symbols

A	Area.
Ar	Arrhenius pre-exponential factor.
c_p	Specific heat.
Ea	Activation energy.
H_R	Enthalpy of reaction.
I	Current.
k	Thermal conductivity.
L	Length.
$\dot{q}$	Energy rate per unit volume.
$\dot{Q}$	Energy rate.
R	Electrical resistance.
R_c	Contact resistance.
R_u	Universal gas constant.
t	Time.
T_c	Charge temperature.
T_{in}	Initial system temperature.
T_w	Wire temperature.
V	Volume.
Z	Reaction rate.
α	Thermal diffusivity.
ε	Emissivity.
γ	Power transfer coefficient.
ρ	Contact resistivity.
ρ_m	Mass density.
σ	Stefan-Boltzmann constant.

Chapter 1

INTRODUCTION

The NASA Standard Initiator (NSI), shown in Fig. 1, is an electrochemical device which plays a critical role in the initiation of pyrotechnic events in the National Space Transportation System and in Shuttle payload applications. The NSI is activated by electrically heating a thin metal bridgewire lying at the bottom of an alumina cup. Energy transfer from the resistive wire to the zirconium potassium perchlorate (ZPP) charge causes it to ignite, initiating a chemical chain reaction which results in a small detonation.

Recently some initiators failed the formal lot acceptance tests at low temperatures (21 K). Subsequent investigations yielded an unacceptable failure rate even at higher temperatures (144 K) . Extensive testing and numerical modelling done by NASA failed to pinpoint the precise cause of failure.[1] In these investigations, it was thought the wire was behaving adiabatically in the initiators that failed. Contact between the wire and the charge was assumed lost, and energy transfer to the alumina was neglected since the alumina was assumed to have a low thermal conductivity at all temperatures. However, experimental tests performed by NASA revealed that the bridgewire burnout time for the NSIs that failed was comparable to that for NSIs that successfully fired.

The purpose of this research was to construct a detailed numerical model for

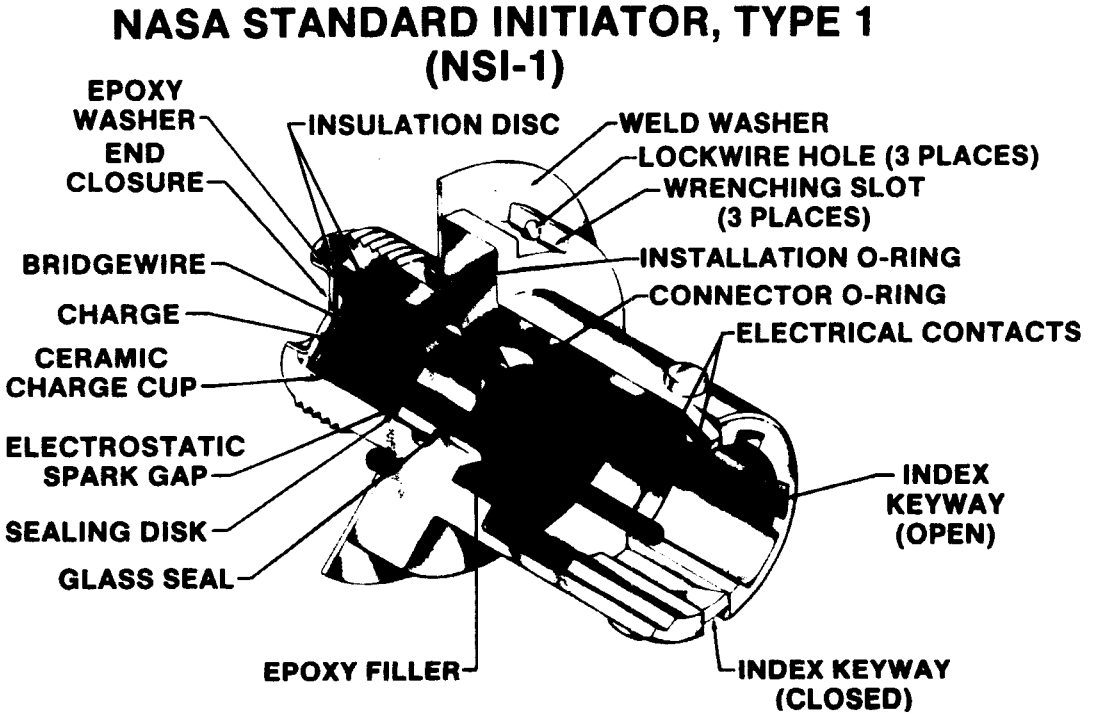


Figure 1 The NASA Standard Initiator.

the heat transfer and ignition mechanism in the NSI. The model was used to investigate the reasons for the NSI failures. Background information is presented in Chapter 2. Chapter 3 provides a description of the investigations previously done by NASA and the conclusions drawn from each analysis. This chapter is provided separately since the NASA report containing this information was not published, and thus is not widely available. Chapter 4 contains the description of the one dimensional model and the results obtained from it. Chapter 5 describes the two dimensional model and the corresponding results. Finally, Chapter 6 contains the conclusions and recommendations.

Chapter 2

BACKGROUND

Introduction to pyrotechnics

Pyrotechnics are chemical compounds that combust exothermically when heated. The energy released is any combination of light, heat, or hot gases. The energy liberated from the combustion of pyrotechnics can be used to produce light (e.g., fireworks), operate assemblies (e.g., external tank separation), or initiate larger explosive devices (e.g., initiators).

A solid pyrotechnic mixture is composed of an oxidizing agent, fuel, and binder. The oxidizer is an oxygen-rich solid that decomposes and releases oxygen when heated. The fuel, which is usually a temperature sensitive material, combines with the liberated oxygen and combusts producing an oxide and heat. The thermochemical energy released from the decomposition of the oxidizer and the combustion of the fuel heats the surrounding unreacted propellant to its autoignition temperature so that it ignites. Thus a self sustaining chemical chain reaction is initiated. The binder, which is usually present in small percentages, holds the components together in a macroscopically homogeneous blend.[2]

Initiators are pyrotechnic devices that convert input energy (from electrical, mechanical, pneumatic, or laser sources) to chemical and kinetic energy output (in the form of heat, pressure, shock waves). Because of their good reliability, high

power to weight ratio, and long shelf life, electrical explosive initiators are the most commonly used as the primary components in the actuation of pyrotechnic devices. The active components in electrical explosive initiators are a resistive wire that is surrounded by the main pyrotechnic charge. In a typical firing mode, an electrical current is passed through the wire. Heat dissipated from the wire raises the charge temperature. When the temperature of the charge exceeds its autoignition temperature, the pyrotechnic mixture starts to release its own thermochemical energy and the charge ignites. Once ignited, the charge should produce enough thermal energy to drive the chemical reaction to completion thus consuming all the available pyrotechnic mixture.[3]

NASA Standard Initiator

The NASA Standard Initiator (NSI) is an electrical explosive pyrotechnic device that converts electrical energy to a pressurized hot gaseous output. The limited high temperature and pressure gases generated may be used to operate small assemblies or to initiate reactions in larger explosives when more energy is needed. The NSI is used on all electrically initiated pyrotechnic events in the National Space Transportation System (NSTS) elements, as well as in shuttle payload and commercial applications.

The active components in the NSI are a resistive wire that is surrounded by a pyrotechnic charge. The wire, which is made of stainless steel (SS-304), is 3 mm long and has a diameter of 50 μm, and thus a nominal resistance of 1.05 Ω. The wire is welded to metal pins at the bottom of an alumina charge cup. The zirconium

potassium perchlorate (ZPP) pyrotechnic charge is pressed into the charge cup at a pressure of 10,000 psi.[4] Here zirconium is the fuel, potassium perchlorate is the oxidizer, and viton is used as the binder.

The purpose of this research was to construct a detailed model of the energy transfer in the NSI. Since most NSI parameters can only be obtained by destructive testing, this model will help evaluate the NSI performance under the various test conditions without the destruction of a prohibitively large number of initiators. First a literature review was conducted to determine if similar initiator problems have been encountered. Experimental, numerical, and analytical results were found.

Literature Search

Leslie and Dietzel[5] performed experimental work on the effect of raising the bridgewire of an initiator from the header surface by imbedding the wire in the charge. Initiators with wires flush with the header material were expected to have more heat loss to the header, and high thermal contact resistance between the wire and its surrounding after the initiator was subjected to thermal and/or mechanical shocks. The initiator used was similar to the NSI with titanium hydride-potassium perchlorate (TiH_x - $KClO_4$, $0.6 < x < 1.9$) used as the charge mix, and a 0.046 mm diameter Tophet C wire was used as the bridgewire. Initiators with bridgewire heights varying from a minimum of 0 mm (flush) to a maximum of 0.43 mm (raised) were loaded with the charge and then subjected to thermal and mechanical shocks. The thermal shock test consisted of placing the initiators in a

forced air circulation chamber maintained at 358 K for five hours, immediately followed by one maintained at 208 K for three hours. After the cycle was repeated seven times, the initiators were returned to ambient temperature. The mechanical shock test, which was repeated twice, consisted of applying a shock of 1500 ± 225 g for a duration of 200 μs in a direction that forced the charge away from the bridgewire. The initiators were subjected to Electrothermal Response Tests (ETR) after fabrication, after thermal shock test, and after both thermal and mechanical shock tests. The ETR tests, which were performed at ambient temperature, consisted of supplying the wire with a 450 mA current for 60 ms. The power transfer coefficient γ (W/K), which is the inverse of the thermal contact resistance between the wire and its surroundings, was determined by monitoring the wire temperature through variation in its resistance. Higher values of γ between the wire and the charge imply more intimate contact between the two and are thus desired.

Results of the ETR tests showed that γ values after loading varied from 2100 μW/K for flush wires to about 2400 μW/K for wires raised 0.43 mm. However, those values varied from 2100 μW/K (flush) to 2700 μW/K (raised 0.43 mm) after the thermal shock, and from 1900 μW/K (flush) to 3200 μW/K (raised 0.43 mm) after both thermal and mechanical shocks. As can be seen, the γ values after loading for the raised wires were not significantly higher than those flush with the header. Those results after both mechanical and thermal shocks indicated that the initiators with raised wires have significantly higher γ values than the ones with flush wires. Therefore, after thermal and mechanical shocks, the initiators with raised wires are more likely to initiate than the flush ones since raised

wires preserve the intimacy between the wire and the charge. It should be mentioned that the header material used in this investigation was glass, and that if ceramic was used the γ values for the flush wire cases would have been significantly higher (6500 μW/K was reported by one of the references), since the thermal conductivity of ceramic is higher than that of glass. By raising the bridgewire a distance of only 0.43 mm, the header material used would not have an effect on the values of γ since the wire would be in contact with the charge only.

Lieberman and Villa[6] performed experimental and numerical work to determine if wire deformation would affect initiator performance. The initiators considered had an initial bridgewire resistance of 1 Ω. The bridgewire was laid over an alumina insulator. To improve the performance of the initiator, a groove was machined into the alumina near the center of the wire length and filled with charge powder. This would decrease the heat loss from the wire to the alumina by insulating the wire from the header, and increase the area of contact between the wire and the charge (20/80% by weight B/$CaCrO_4$). ETR tests (described above) revealed an unacceptable bridgewire resistance increase (>10%) in 15% of the initiators tested. Further tests showed an increase in wire resistance of 6% for the entire lot after charge loading. The purpose of this work was to determine if this resistance shift would have an effect on the performance of the initiator. When a Scanning Electron Microscope (SEM) was used to examine the charges and wires of some initiators, it was noticed that the bridgewire had deformed into the machined groove in the areas around the edges of the groove. Necking, pitting and shearing of the wire were observed. When prepared bridgewire samples were

examined using the SEM, no internal porosity or fracture planes were observed. Thus pitting and shearing of the wire were assumed to be surface deformations that have no serious effects on the initiator performance. Loading sensitivity tests were performed to determine if the resistance increase and bridgewire appearance were dependent on loading pressure. Results obtained using SEM and ETR analyses did not correlate bridgewire resistance and appearance with the loading pressure. Additionally, low temperature initiation tests were conducted. Initiators with high wire resistances test fired at low temperatures (219K) functioned reliably, and some subjected to thermal cycling also functioned normally.

Thermal analysis of the initiator was then used to assess possible ignition failure due to premature bridgewire burnout. The combustion of the charge was modelled by the Arrhenius equation, and all thermophysical properties were assumed constant. The firing mode used was a constant current 3.5 A. Three different models were developed. The first assumed no bridgewire necking, perfect contact between the wire and the charge, and no contact between the wire and the alumina. The second model assumed necking to occur in the wire, perfect contact between the wire, charge, and alumina. The third model was similar to the second, except it assumed no contact between the wire and the alumina. Results from the models showed that melting of the wire always occurred around the center of the wire, and that maximum necking values for ignition should be between 50 and 64%. Ignition of the charge also occurred around the center of the wire. Model results eliminated premature wire burnout as a possible failure source. It was

concluded that the resistance and appearance of initiators with a machined groove change did not have a significant effect on initiator performance.

Donaldson[7] performed an analytical study to determine the ignition temperature and ignition time of a wire imbedded in a pyrotechnic mixture. He used a cylindrical coordinate system model with variations in the radial direction. The wire was assumed to be infinite in the z direction. The thermophysical properties of the wire and charge were assumed to be temperature independent. Further, perfect thermal contact was assumed at the wire/charge interface. Therefore, the temperature of the wire was assumed equal to that of the charge at the interface. The wire was assumed to be heated by a constant current, and temperature gradients in the wire were neglected. The Arrhenius rate law was used to model the chemical reaction in the charge.

The charge ignition temperature derived was dependent on the current through the wire, wire diameter, kinetic rate parameters, and thermophysical properties of wire and charge. The ignition temperature is thus a system property, and not a property of the propellant only. The time to reach the above ignition temperature was then obtained mathematically. The model was then used to simulate which consisted of an Evanohm wire with a $TiH_2/KClO_4$ charge. The times to ignition obtained from the analytical model were evaluated for various wire diameters and were found to agree well with experimentally determined ones.

Selberg and Johansson[8] constructed a one dimensional (radial) energy transfer model of a wire surrounded by explosive material. The explosive charge

was assumed to be at an initial temperature T_0, while the temperature of the wire T_w was higher than T_0. The wire was assumed to have a uniform initial temperature distribution. Perfect thermal contact between the wire and the charge was assumed, and constant thermophysical properties were used. Thermal decomposition of the charge was modelled by the Arrhenius equation. At time 0, the wire was placed in contact with the pyrotechnic charge, initiating the transfer of thermal energy from the wire to the powder. The initial temperature of the wire was incrementally increased until one that produced ignition of the charge was achieved. This temperature, labelled minimum initial wire temperature, was determined for wires with various radii (r), and was found to be inversely proportional to the radius of the wire ($T_{min} = a + b/r$, where a and b are constants). Additionally, the minimum energy delivered to the wire was plotted versus the square of the wire radius, and an equation relating the minimum energy supplied to the wire per unit length with the wire radius was developed ($E_{min} = c\, r^2 + d\, r$, where c and d are constants).

No previous work was found that included the thermal contact resistance, heat loss to the header, and temperature dependent thermophysical properties in the model. Work done by Leslie and Dietzel, and by Lieberman and Villa seem to indicate that the problem of heat loss to the header was recognized, but no supporting numerical work was found.

A search for the thermophysical properties of the materials pertinent to the NSI behavior (stainless steel 304, alumina, zirconium, potassium perchlorate) was also conducted. The results of the search are summarized in Appendix B through

curvefit equations for the thermal conductivity and the specific heat of the various materials.

Finally, a search was performed for the Arrhenius rate parameters for the reaction between Zr and $KClO_4$. Since no data was found for that reaction, the rate parameters for the $Zr/KClO_4$ reaction were assumed to be those of the reaction between Zr and O_2. Belle and Mallett[9] performed experiments on the reaction of zirconium with oxygen. Zirconium bars were machined into cylindrical rods. The prepared samples were placed in an evacuated chamber at temperatures ranging from 848 K to 1223 K. A pre-measured amount of oxygen gas was then introduced to the chamber. The amount of oxygen that has reacted with the zirconium was determined by taking the difference between the initial amount of gas in the chamber and that remaining after a certain amount of time had elapsed. The rate constant Z, the Arrhenius pre-exponential factor Ar, and the activation energy z for the reaction were then determined through various plots of the results obtained. A value of 197.6 ± 4.2 kJ/mol was determined for E_a, and 3.9×10^6 (ml O_2/cm^2 of metal surface)^3/sec was determined for the pre-exponential factor.

Chapter 3

PREVIOUS WORK

When the high failure rate of initiators at low temperature was discovered an extensive series of investigations was initiated to determine the reason for failure. This work, described below, was undertaken at NASA Johnson Space Center, and Hi Shear Corporation of Torrance, California with consulting support from other NASA Centers, other government agencies, and commercial organizations.

Scanning Electron Microscopy and Energy Dispersive Spectroscopy

These tests were conducted to determine if charge contamination or morphology varied between lots that had a high failure rate and those that experienced no failures. A scanning electron microscope was used to examine the NSI pyrotechnic charges. Some differences in mix morphology (particle size, shape, and size distribution) were found between lots from different suppliers but there was no correlation between the firing performance and the morphology of the charge. When charge samples from some failed initiators were examined, it was noticed that the potassium perchlorate ($KClO_4$) at the wire/charge interface had fused. Since $KClO_4$ begins to decompose at temperatures below its melting point it was concluded that reaction had been initiated in these initiators but that a self-propagating condition was not achieved.

Charges from initiators that failed to ignite, as well as some which were not

yet tested for ignition, were examined using energy dispersive spectroscopy. It was found that some charge samples were contaminated, but the trace quantities of silicon, iron, and chromium detected would have no effect on the initiator performance. It was also concluded that the contaminants were probably introduced into the samples while removing the charge from the initiators.

Zirconium Oxidation Level

The objective of this test was to determine if the oxidation level of zirconium affected the sensitivity of the charge. Because zirconium is a very sensitive metal, it is shipped, delivered, and stored in an aqueous solution. Before it is mixed with potassium perchlorate, the zirconium is dried in an oven while exposed to air. The drying procedure was not tightly controlled between various zirconium samples: drying temperature varied from 65° to 93°C, and drying time varied from a few hours to several days. It was thought that the extended drying time at higher temperatures might lead to excessive surface oxidation of the zirconium, thus making the charge mixture less sensitive to heat. Four samples from each of the three different suppliers of zirconium were subjected to various drying conditions, and then tested for oxidation levels. The results showed that even though the oxidation level increased with the time of exposure and drying temperature, the changes in oxidation levels of the various samples were negligible. It was concluded that variation of the drying parameters did not have a significant effect on the sensitivity of the pyrotechnic charge.

Zirconium Sources

This test was conducted to determine if zirconium samples from different suppliers behaved differently. Zirconium samples from two vendors were mixed with potassium perchlorate and loaded in the NSI. The initiators were then test fired at 22 K. Comparable ignition performance between the initiators with different zirconium samples was obtained. However, zirconium from one vendor had shown excellent ignition performance at low temperatures in some lots but the highest failure rate (85% failure at 22 K) in another lot. It was thus concluded that zirconium was not the source of the problem.

Reconsolidation Test and Pellet Expulsion Test

It was postulated that the charge pellet was contracting away from the bridgewire and walls of the charge cup and breaking free. A series of tests were conducted in which the pellet was forced against the bridgewire by placing a dead weight on it. An 8 gram ram was loaded on the charge face, and the initiators were test fired (PIC firing mode) at 22 K. The failure rate experienced with the deadweight was comparable to that experienced without it at the same test temperature. Hence it was concluded that pellet did not break free at low temperatures.

Additionally, the force required to push the charge out of the alumina cup was measured at room temperature and liquid nitrogen temperature (294 K and 77 K). The results obtained at 77 K varied from 10 - 110 N (2 - 25 lbf), while results at 294 K varied from 90 - 190 N (20 - 42 lbf). The diminished force was

consistent with differential contraction but since a force of at least 10 N (2 lbf) was required even at low temperature, it was concluded that the mixture could not be moving freely in the cup and pulling away from the bridgewire.

Propellant Movement Test

This experiment has not been completed yet so no results are available. The objective of this test is to determine if movement of the charge is occurring at low temperatures, and if this movement is causing the initiators to fail. Holes that admit one optical fibre will be drilled through the charge cup. The initiators will then be placed in between a light source and a light sensor. Movement of the charge cup away from the wire will be detected by monitoring the reading of the light sensor. An increase in the light detector reading indicates that the charge has pulled away from the wire, since a light path has been established.

Sealing Washer Test

The bonding strength of the sealing washers used by the two different NSI manufacturers was tested. These tests were performed to determine if the bonding strength of the sealing washers at low temperatures (liquid nitrogen) was less than that at ambient temperatures. Charge pellets of some initiators were removed from the charge cup and replaced with plugs of teflon that had dimensions identical to those of the inside charge cup, and then assembled with the two different kinds of washers. Initiators were then tested at room and liquid nitrogen temperatures. The bonding strength of the sealing washers was determined by measuring the minimum force that would break the washers. Test results showed that the washers

used by the manufacturer whose initiators experienced failure were stronger (at both ambient and liquid nitrogen temperatures) than those used by the other. It was concluded that the washers were not the cause of failure.

Firing Mode

The objective of this test was to determine if the firing mode has an effect on the failure rate. The prototype initiators were tested (qualified) by firing them with a constant current of 5 A (CC mode). During a shuttle mission initiators are fired by the Pyrotechnic Initiator Controller, which consists of a 680 μF capacitor charged to 38 V discharging into a nominal resistance of 1 Ω (PIC mode). The bridgewire resistance is originally 1 Ω and increases with temperature; the maximum resistance prior to melting is approximately 1.6 Ω. To simulate this firing mode initiators were requalified using a Standard Firing Unit consisting of a 1000 μF capacitor charged to 20 V (SFU mode). Figure 2 is a plot of the power dissipated by the wire versus time for all three firing modes. The PIC firing mode had the highest failure rate: 85% at 22 K, and the SFU and CC firing modes had comparable failure rates at 22 K (40 and 42% respectively).

The bridgewire destruct times measured for the CC and SFU modes (~2ms at 21 K) were an order of magnitude larger than the destruct time for the PIC mode (0.2 - 0.4 ms at 21 K). It was thought that smaller currents applied for longer times should deliver more energy than larger currents for shorter times, which might explain the improved performance with the CC and SFU firing modes. Since the

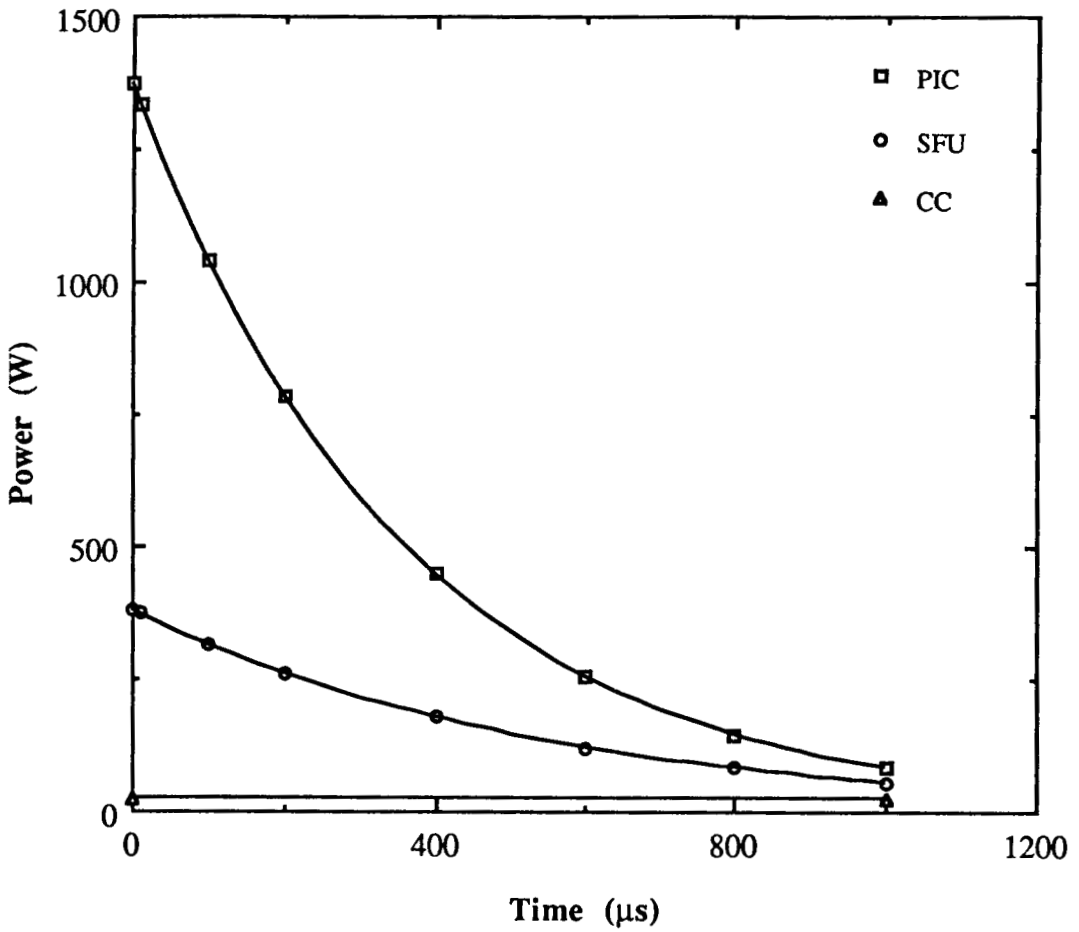


Figure 2 Power dissipated by bridgewire vs. time for all three firing modes.

wire destruct time was the shortest for the PIC mode, and since the maximum wire temperature is limited to the wire melting temperature, the PIC was expected to have the smallest energy delivery from the wire to the pyrotechnic charge. NASA set up

instrumentation to measure current and voltage across the bridgewire and pins during firing so that the energy delivered to the wire could be measured for each firing mode. It was assumed that this energy was transferred from the wire to the charge mixture. The tests at 21 K showed that maximum energy was transferred in the bridgewire in the PIC mode (77 mJ), and the minimum in the CC mode (56 mJ). Hence the firing mode with the highest failure rate had the most energy transfer to the bridgewire. Therefore, it was concluded that increased energy delivery into the bridgewire (and hence the charge) much above the threshold required for firing did not improve the performance of the NSI.

Electrothermal Response Test

The objective of the Electrothermal Response Test (ETR) was to measure the thermal contact resistance between the wire and the charge mix. Initiators were tested using ETR at both ambient temperature and 77 K; the same units were then test fired at 21 K. The tests showed that thermal contact resistance increased at lower temperatures. This is expected because of differential thermal contraction. However two different lots with approximately the same mean value of γ (1256 and 1284 μW/K) had dramatically different failure rates (0% and 85% respectively). Additionally, initiators within a given lot that failed did not have the lowest values of γ. It was thus concluded that increased thermal contact resistance between wire and charge was not the cause of failure. This inconclusive result was perhaps the most puzzling since inhibited energy transfer from bridgewire to charge mixture appeared to be the most logical reason for failure.

Heat Transfer Analysis

A simple two-dimensional axisymmetric numerical model of the NSI was constructed to study transient heat transfer between the bridgewire and the mix and to determine the sensitivity of the NSI to the initial temperature of the system. Chemical reactions were not modelled and ignition was assumed to occur if the charge temperature reached 590 K. Thermophysical properties of the ZPP mixture were assumed constant independent of temperature. Temperature dependent specific heat and thermal conductivity of the bridgewire were included in the model. The axisymmetric model was justified on the basis that the alumina charge cup had negligible contact area with the bridgewire and was a good thermal insulator. Hence it was assumed to have little influence on initiator performance. The thermal conductivity of the charge mix was set at a nominal value of 24.9 W/m K (14.4 Btu/hr ft °F) at 100% packing density, which could be reduced or increased to allow for voids and uncertainty in property values. The model simulated the SFU and CC firing modes. The effect of thermal contact resistance was modeled by using a contact area factor that measured the "effective" contact area between the wire and the charge (uniformly distributed to preserve axial symmetry).

The model predicted ignition at all initial temperatures for contact area factors greater than 10^{-6}. Such small contact areas were considered unlikely, implying that thermal decoupling of wire and charge was unlikely to be the cause of failure. Additionally, the measured bridgewire burnout times of initiators that failed corresponded to contact factors of about 60% in the model. With an initial temperature of 106 K, ignition times computed from the model ranged from 0.06 to

0.13 ms for the SFU firing mode depending on contact area factor and mixture thermal conductivity assumed. Bridgewire burnout times for this mode ranged from 0.13-0.62 ms for the cases studied. In the CC firing mode ignition times ranged from 1.24 to 1.67 ms and bridgewire burnout times varied between 1.7 and 5.7 ms. For a bridgewire completely insulated from the charge, burnout times were predicted to be 0.125 ms and 1.7 ms in the SFU and CC firing modes respectively. Experimental data at this initial temperature gave bridgewire burnout times in the range 0.36 - 0.66 ms for the SFU firing mode, and 4.42 - 6 ms for the CC firing mode. Initiators that failed to ignite in the SFU mode had comparable bridgewire burnout times (0.46 - 0.64 ms). From these results it was concluded that thermal decoupling was not the cause of failure.

Chapter 4

ONE DIMENSIONAL MODEL

Model Description

In the first phase of this work a highly simplified one-dimensional axisymmetric model of the NSI was constructed. It was assumed that this simple model would reproduce the qualitative features of the ignition process. Heat transfer between the bridgewire and charge mixture was assumed to occur by conduction only. Calculations of the radiation heat transfer (Appendix A) showed that the radiation heat transfer was likely to be insignificant because the bridgewire fused shortly after it became hot enough to radiate. The wire was assumed to be heated by a constant current of 5 A to simulate the CC firing mode.

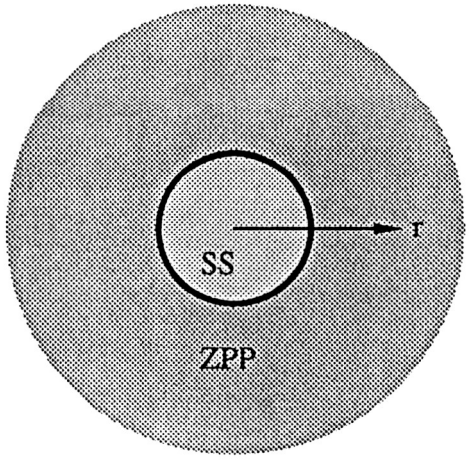


Figure 3 Geometry of one-dimensional model.

The geometry of the model used is shown in Fig. 3. The wire was assumed to infinitely long in the z direction, and the effect of differential thermal contraction was modelled by a thermal contact resistance (R_c) between the wire surface and the charge mixture surrounding it. The variation of contact resistance with temperature was neglected. The use of a thermal contact resistance permitted the model to account for a temperature difference between wire surface and charge adjacent to it. Temperature gradients within the bridgewire were neglected (Biot number = 0.1 «1).

The combustion reaction is assumed to occur in a single step according to the following reaction:

$$2\,Zr + KClO_4 \rightarrow 2\,ZrO_2 + KCl$$

The kinetics of the chemical reaction are modelled by the Arrhenius equation:

$$Z = Ar\, e^{(-E_a/R_u T)}$$

where Z is the reaction rate at temperature T, Ar is the pre-exponential factor, E_a is the activation energy, R_u is the universal gas constant, and T is the absolute temperature.

The volumetric thermochemical energy released is given by:

$$\dot{q}_{chem}\ (W/m^3) = \rho_m\, Z\, \Delta H_R = \rho_m\, \Delta H_R\, Ar\, e^{(-E_a/RT)},$$

where ρ_m is the mean mass density of the propellant, ΔH_R is the enthalpy of reaction provided in the NSI-1 design and performance specifications (1395 cal/g mixture = 1870 MJ/kmol of $KClO_4$),[10] and Z is the reaction rate. Thermophysical properties were assumed to be constant for these calculations and many properties had to be estimated because data was not available at the time.

The one dimensional model was developed because the packaged computer code available (PATRAN) could not handle variable time steps which are required to resolve the rapid temperature rise that occurs once chemical reactions begin to release significant amounts of energy. An explicit finite difference scheme which is forward in time and centered in space (FTCS)[11] was employed on the conservation of energy equation for the ZPP region:

$$\frac{\partial T_c}{\partial t} = \alpha_c \left[\frac{\partial^2 T_c}{\partial r^2} + \frac{1}{r} \frac{\partial T_c}{\partial r} \right] + \frac{\dot{q}_{chem}}{\rho_c \, c_{pc}}$$

with the boundary conditions $T_c\,(t, r \rightarrow \infty) = T_{in}$

$$m_w c_w \frac{\partial T_w}{\partial t} = RI^2 - \frac{(T_w - T_c)\, A}{R_c} \qquad \text{at interface.}$$

and the initial condition $T_w = T_c = T_{in}$ at $t = 0$

where T_{in} is the initial temperature of the system. The subscripts w and c refer to the wire and the charge respectively. For this method to be stable, the diffusion number $\equiv \alpha \Delta t/(\Delta r)^2 \leq 1/2$, where α is the thermal diffusivity. The model was

verified by comparing with results from PATRAN for the transient heat conduction problem with no chemical reactions.

Since the activation energy was not known, it was assumed to be $0.13\Delta H_R$ (= 243 kJ/mol) for the base case and the pre-exponential factor, Ar, was set to 1.82×10^{17} s^{-1}. This was found to give ignition at about 600 K in reasonable accord with experimental observations.[12] The thermal contact resistance between wire and charge inferred from differential thermal analysis measurements on samples of initiators varied between ~5×10^{-5} m^2 K/W at 300 K to ~3×10^{-4} m^2 K/W at 100 K.[13] As a base case the contact resistance was set at a low value of 10^{-7} m^2 K/W. This value is smaller than the ones obtained by the ETR measurements since the contact resistivity in the model is assigned for the whole length of the wire since the model is two dimensional. Table 1 summarizes the values of thermophysical properties used in the calculations.

Results

Some results obtained with this simple model are presented in Figs. 4-6. On these figures the temperature of the bridgewire is shown with a dashed line, and the temperatures of the first two nodes within the ZPP are shown in solid lines. Each node in the charge represents a zone 60 μm thick and zone temperature was calculated at the center of each zone. The zone thickness was adjusted to satisfy the stability criterion mentioned previously. The first node is one half the standard thickness so that the temperature at the wire-charge interface could be computed. Using the base case parameters we determined the time to ignition as a function of

initial temperature. The results are shown in Fig. 4 for initial temperatures of 300 K, 100 K, and 10 K which are the approximate temperatures at which initiators

Table 1 Properties used in Preliminary Calculation

Bridgewire Material: SS-304 Diameter, D = 50 μm

ZPP charge: Stoichiometric composition (by weight) Zr: 0.568, $KClO_4$: 0.432

	SS 304	Zr	$KClO_4$	ZPP
Density (kg/m^3)	8030	6530	2500	4600
Specific Heat (J/kg-K)	500	280	200	250
Thermal Conductivity (W/m-K)	-	22.7	1*	13
Melting Temperature (K)	1700	-	-	-

* No thermal transport data were found in a preliminary search. The value above is a crude estimate by analogy with similar compounds (KNO_3, NH_4ClO_4) for which data could be found.

have been tested. Ignition is assumed to occur after the first two nodes show a steep rise in temperature. The model predicts that ignition time increases from approximately 1.5 ms at room temperature to 10 ms at 100 K and 27 ms at 10 K.

Tests were performed on initiator samples when they were first qualified for use on the Space Shuttle. The function time was measured by recording the time for the pressure to rise in a closed chamber. The function time was 2 ms at 110 K, and 1 - 3 ms at 10 K for a constant current of 5 A (CC firing mode).[14] The function time was 3 ms at room temperature for a constant firing current of 3.5 A. In some cases the time to "first pressure" and time to maximum pressure were recorded separately, but the latter was identified with the function time of the initiator. The function times in the numerical model corresponds to ignition of the first ZPP node and are thus expected to be smaller than the experimental measurements. Thus the function times predicted by the simple model are too large, but within the expected accuracy. The discrepancy of about an order of magnitude at low temperatures can be attributed to the assumption of constant properties and to errors in assumed property values.

Thermal contact resistance is expected to increase at low temperatures because of differential thermal contraction. The effect of contact resistance is displayed in Fig. 5 which shows the time to ignition for an initial temperature of 10 K and activation energy of $0.13\Delta H_R$. Increasing the contact resistance by a factor of 100 to 10^{-5} m^2 K/W increases the temperature difference between the wire and the charge but reduces the time to ignition. This somewhat surprising result can be explained by noting that the wire heats up much faster when the contact resistance is poor. This higher temperature source can heat up the surrounding charge to ignition temperature faster than the corresponding case at low contact resistance. If

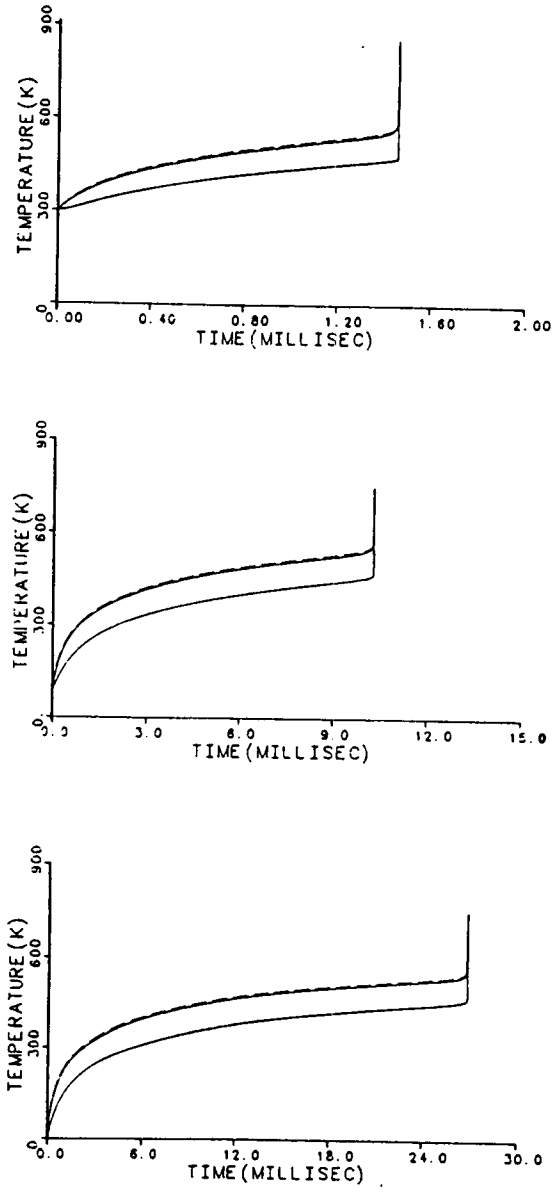


Figure 4 Temperature vs Time for wire (dashed) and first two ZPP nodes. Contact resistivity = 10^{-7} m^2 K/W. Activation energy = 0.13 ΔH_r. Initial Temperature = 300 K (top), 100 K(middle), 10 K(bottom).

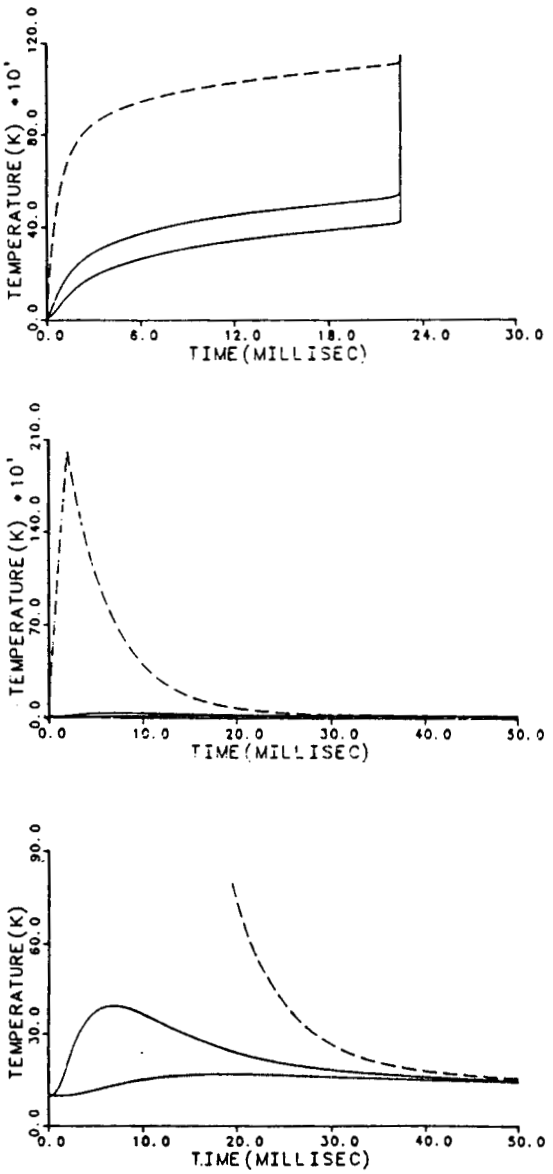


Figure 5 Temperature vs Time for wire (dashed) and first two ZPP nodes. Initial Temperature = 10 K. Activation energy = 0.13 ΔH_f. Contact resistivity = 10^{-5} m^2 K/W (top), 10^{-4} m^2 K/W (middle). Bottom graph same as middle with expanded ordinate.

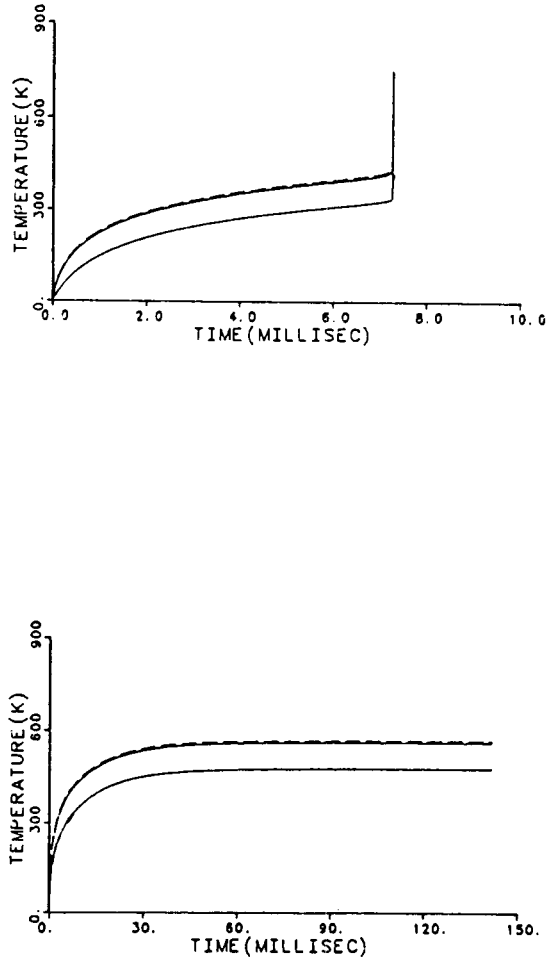


Figure 6 Temperature vs Time for wire (dashed) and first two ZPP nodes. Initial Temperature = 10 K. Contact resistivity = 10^{-7} m^2 K/W. Activation energy = 0.10 ΔH_r (top), 0.20 ΔH_r (bottom).

the contact resistance is increased further to 10^{-4} m^2 K/W then the bridgewire is almost adiabatic and heats up to melting point within a few milliseconds but the charge does not ignite. In this calculation it is assumed that the wire fuses and no longer carries current once the melting point is reached. The wire subsequently cools down by slow heat transfer to the charge. For the contact resistance and activation energy chosen for the calculation this heat transfer is insufficient to initiate chemical reaction in the ZPP.

The sensitivity of the model predictions to the activation energy of the charge is shown in Fig.6 for an initial temperature of 10 K and a contact resistance of 10^{-7} m^2 K/W. When the activation energy is reduced by 30% to $0.1\Delta H_R$ the model predicts that the time to ignition is reduced by a factor of approximately 3 from 27 ms to 7.3 ms. If the activation energy is increased to $0.2\Delta H_R$, the mixture does not ignite at all but approaches steady state. Hence the ignition time is very sensitive to the activation energy of the charge mixture.

The results obtained with this simple model showed that ignition time had a non-monotonic dependence on contact resistance and was sensitive to the chemical reaction rate (through the activation energy). However, the function times at low temperature predicted by the model were too large by about an order of magnitude, and it seemed unlikely that useful conclusions could be drawn from the model without additional refinements. In particular, the property data used to model the ZPP mixture were based on extrapolations from similar compounds. It was necessary to improve the accuracy of the thermophysical property data on zirconium

and potassium perchlorate, and include temperature dependent properties of the other materials if possible. This work is described in the next chapter.

Chapter 5

TWO DIMENSIONAL MODEL

Property Data

To improve the accuracy of the model, an extensive literature search was conducted to obtain thermophysical properties of all NSI materials (SS, Zr, $KClO_4$, Al_2O_3) as a function of temperature. The results of the search are summarized in Appendix B. During the course of this search the thermal conductivity of alumina was found to be a strong function of temperature. At cryogenic temperatures the thermal conductivity of high purity alumina is 10 times higher than that of ZPP. Thus the alumina charge cup acts as a very effective heat sink at low temperature if the wire makes good thermal contact with it. Furthermore, the thermal conductivity of the alumina is very sensitive to purity. As Fig. 7 shows, a difference of several orders of magnitude exists between the conductivity of 98% dense alumina and that of 100% dense alumina. The data were obtained from two different references but their relative consistency is made more credible because the data for sapphire obtained from the two references are in good agreement.

Chemical kinetic data for the reaction between Zr and $KClO_4$ are not available and so initially the activation energy for the chemical reaction was taken from the Zr/O_2 reaction data[9] and set to 193.1 kJ/mol. The pre-exponential factor for the chemical reaction rate was modified to $A = 5.19\times10^{25}/\sqrt{T}\ s^{-1}$. This

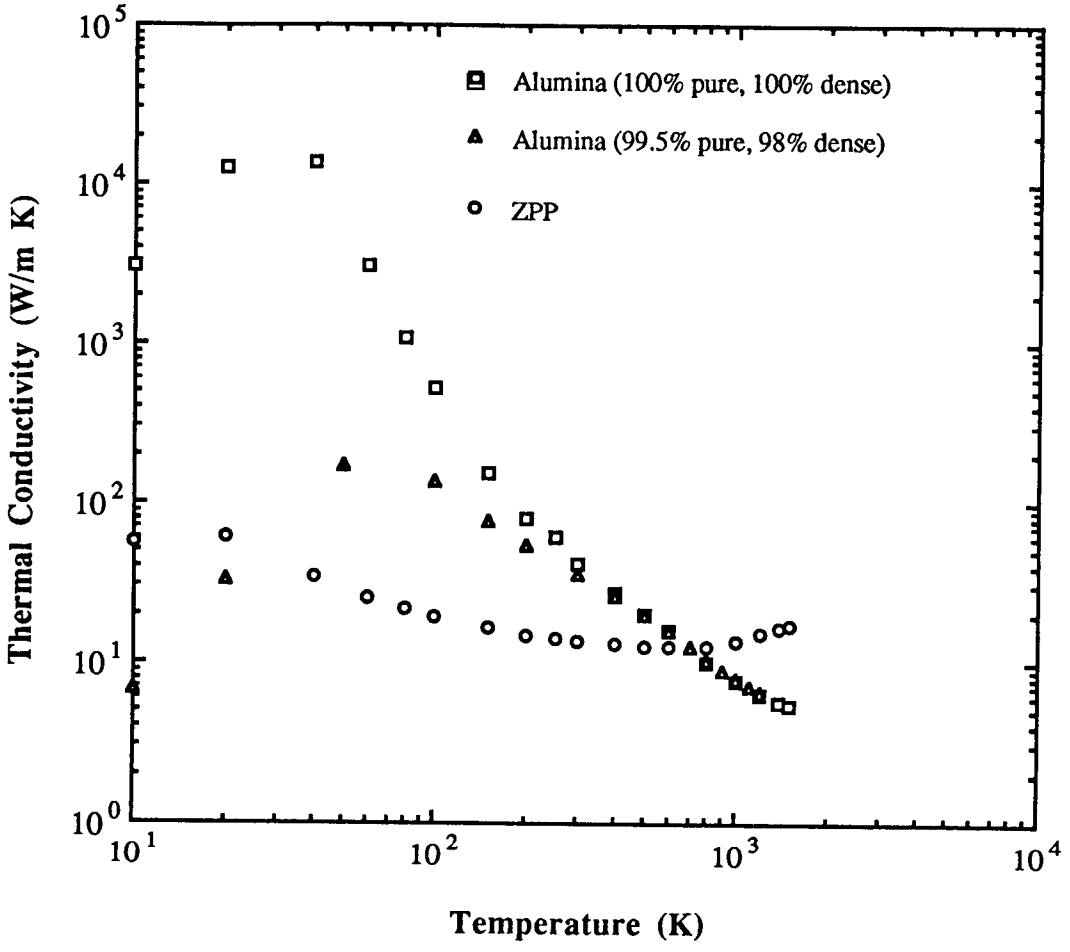


Figure 7 Thermal conductivity of 100% dense alumina, 98% dense alumina, and ZPP vs. Temperature.

combination of kinetic parameters produced a model ignition temperature that was in good agreement with the experimental data (580 K), but failed to reproduce the no-fire case (initial temperature of 422 K, 1 A constant current firing mode). The

sensitivity of the results to the pre-exponential factor and the activation energy were then studied parametrically. A combination of parameters that would successfully model the no-fire case, as well as yield an ignition temperature in agreement with experimental data was desired. This was accomplished by constructing a small model in Cartesian coordinates. One "hot" ZPP node was assumed to be imbedded in surrounding "cold" ZPP nodes. The model was used to determine the minimum temperature of the hot node that would produce ignition in the charge when the cold nodes were at an initial temperature of 100 K. An ignition temperature of 680 K was obtained, and the no fire case was successfully reproduced when the activation energy and pre-exponential factor were set to 208.2 kJ/mol and 3.71×10^{25} s^{-1} respectively. Thus the volumetric energy release rate due to chemical reaction was modelled by

$$\dot{q}_{chem}\ (W/m^3) = \rho_m\ \Delta H_R\ Ar\ e^{(-E_a/R_uT)} = 1.0\times10^{31}\ e^{(-25040/T)}$$

where ρ_m is the mean mass density of the mixture (4600 kg/m^3).

The old and new kinetic rates are plotted vs. temperature in Fig. 8. As can be seen, the values of the reaction rates for both old and new data were almost equivalent to each other at temperatures exceeding 550 K. It is at lower temperature that a larger difference between the old and new values of the reaction rates (at the same temperature) existed. Figure 8 also displays the sensitivity of the model to kinetic rate parameters. A difference in the kinetic rate values of about 10% seemed to make the difference in reproducing the no-fire case.

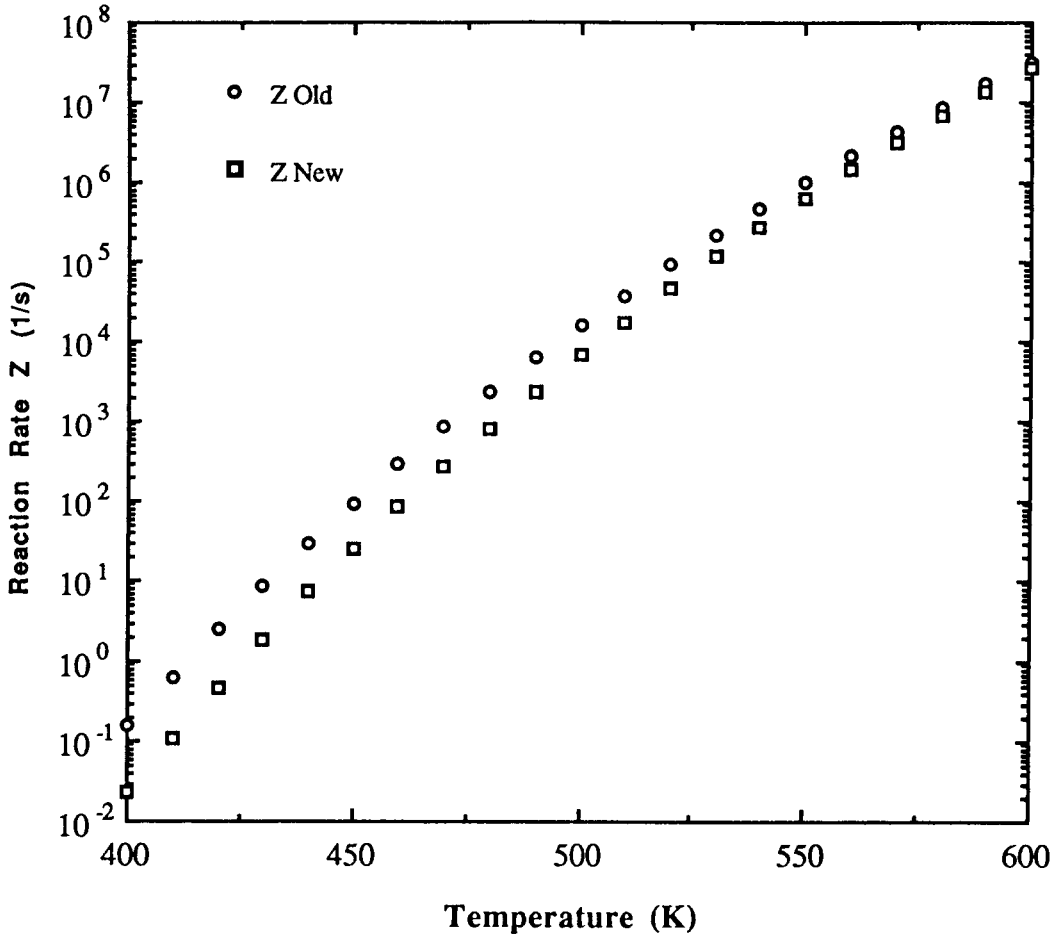


Figure 8 Old and new Reaction Rates vs. Temperature.

Model Description

A two-dimensional model incorporating temperature dependent thermophysical properties for the various materials was developed. The model

allows for the simulation of all three firing modes: the Constant Current (CC), the Standard Firing Unit (SFU), and the Pyrotechnic Initiator Controller (PIC) discussed previously. The value of the current was evaluated at each time step, and was set to zero for the whole wire when the temperature in any portion of the wire exceeded the melting temperature of stainless steel (1700 K). The numerical method used employs the Alternate Direction Implicit (ADI) technique developed by Peaceman and Rachford.[15] (The derivation of the heat diffusion equation in the ADI form is included in Appendix C). This method is unconditionally stable for linear problems. Since the thermophysical properties in this model are temperature dependent, the problem is no longer linear and in some temperature ranges the problem became highly nonlinear because of the strong dependence of the properties on temperature. This caused some instabilities when large time steps were used, but smaller time steps continued to show the needed stability. Small time steps are also needed to maintain adequate accuracy in implicit calculations.

In the two-dimensional model the cylindrical wire is approximated by a square, and a Cartesian coordinate system was used. This was done mainly to facilitate the modeling of the alumina/ZPP interface, and to allow the independent specification of contact resistances between the wire and its surroundings (this is discussed in more detail later in this chapter). Figure 9 shows a portion of the model close to the wire and specifies the x and y axis orientations. The area enclosed by the dashed rectangle is that which is shown by the false temperature plots in the following section. All calculations were performed per unit length (z) of wire and gradients in the z-direction were neglected. Even though the geometry

of the system was distorted, the areas of the various sections (charge, wire, and alumina) were chosen to preserve the correct relative masses. Different spatial grid sizes were used to improve the spatial resolution of temperature gradients in the region where the heating is occurring. In the region close to the wire, the grid size was set to 4 μm in both the x- and y- directions. Far away from the wire the grid

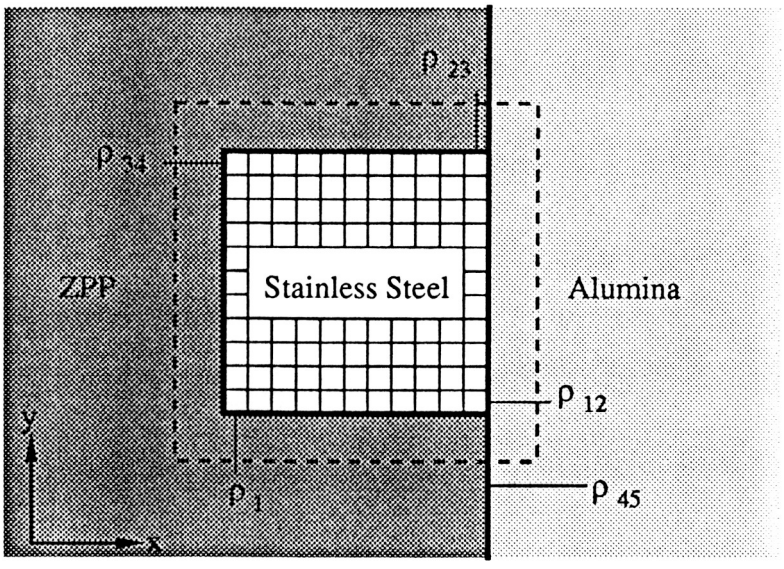


Figure 9 Mesh used in the two-dimensional model of the NSI.

size was set to 193.6 μm in the y-direction in both ZPP and alumina, 195.1 μm in the x-direction in the ZPP, and 241.4 μm in the x-direction in the alumina. The mesh had 71 nodes in the x-direction and 53 nodes in the y-direction, with the wire consisting of 11 nodes in each of the directions.

The computational mesh simulating the model geometry was rather large. To improve computational efficiency the code was run to determine the extent of

thermal diffusion at the onset of ignition with all the contact resistances set to zero and the initial system temperature at 10 K. This combination of parameters was used because it yields the "worst case" results - maximum energy transfer from the wire to its surroundings and the longest heating time before ignition. The spatial temperature distribution at ignition showed that nodes far away from the wire were unaffected because the thermal diffusion was slow relative to the rate of resistive heating. The nodes outside the "thermal front" were eliminated from the grid in subsequent computations to reduce computing time. The resulting grid had 46 nodes in the x-direction and 53 nodes in the y-direction which reduced the number of nodes by about 35%.

Thermal contact resistances were also incorporated in the model. An equivalent thermal conductivity for two adjacent control volumes including contact resistance was determined using thermal resistances (derived in Appendix D). This value was evaluated after every sweep since the thermophysical properties are temperature dependent. Initially, the thermal resistivity, ρ_i, (and hence contact resistance, $R_i = \rho_i/A_i$) for each face was specified independently. In reality the wire may be making intimate contact with only a portion of the charge interface. Additionally, the calculated minimum values of the power transfer coefficient (γ) for the model to predict ignition of the NSI were significantly higher than those measured by NASA using ETR tests. Hence, the model was changed so that the contact resistances could be independently specified along each node on the wire/charge interface. This resulted in 45 contact resistances, 44 along the wire/charge and wire/alumina interface and one along all of the charge/alumina

interface (Fig. 9). This was done to compute the temperature distribution and evaluate the performance of the NSI for various combinations of contact resistances between the bridgewire, the charge mixture, and the charge cup. Even though the new values of (γ) for which successful ignition of the NSI was predicted were reduced, they were not in complete agreement with those measured by NASA. This is due to the two-dimensional nature of the code. The contact resistances were specified for the entire length of the wire (z direction). If, for example, the wire is assumed in good contact ($\rho=10^{-6}$ m^2K/W) with the charge along whole wire/charge interface, and in poor contact ($\rho=10^{-3}$ m^2K/W) with the charge cup, then a γ value of 396 mW/K is obtained when the resistivities are specified over the entire length of the wire. When the good contact between the wire and the charge is specified over only a portion of the wire length ($\approx$1 wire diameter in the z direction), the resulting γ drops to a value of 6.9 mW/K, which falls well in the range of power transfer coefficient values determined experimentally by NASA using ETR (values ranged from 1 mW/K to 11 mW/K). The results produced by the model, however, are meaningful if the model dependence on γ is studied relatively and not absolutely. Even though three dimensional modelling of the wire might reproduce the exact values of γ, the computational time would be very large since the numerical algorithm employed would not be unconditionally stable as in the two-dimensional case.

After the wire melted, the contact resistances between the wire/charge and wire/charge cup were allowed to be changed. Displacement of the molten metal might occur due to a combination of forces acting on it. These forces could include

body forces (induced by gravity), and surface tension forces. The wire might then wet one portion of its interface in some cases, and another portion in other cases. Therefore a possible combination of resistances were studied to determine if the area that the wire "wets" after melting has an effect on the initiator performance.

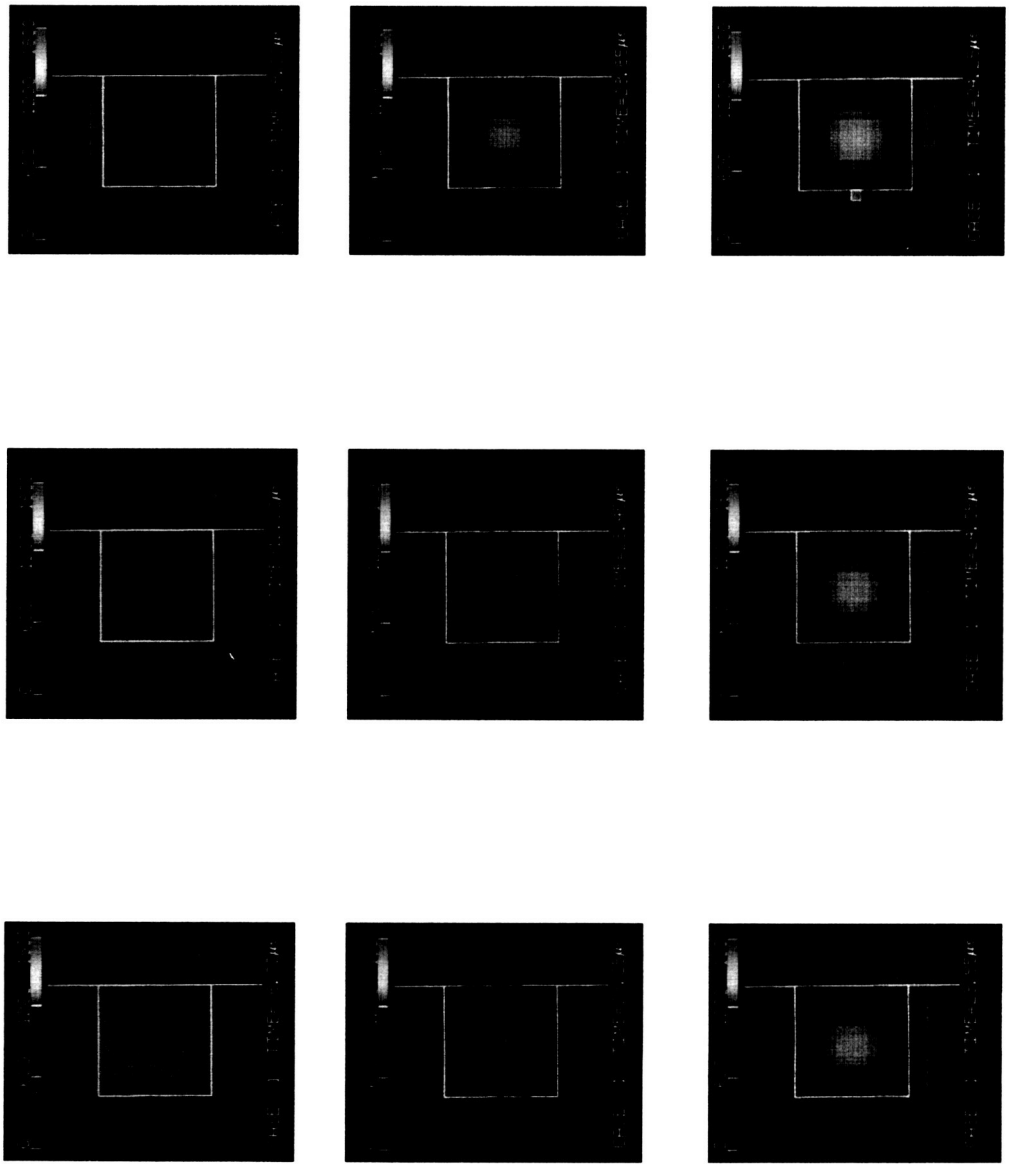
Results

Some results obtained using the code are presented in Figs. 10-15 which show false color plots of the spatial temperature distribution at various times after firing. The time is noted below each plot. Green represents computed temperatures of 1800 K or above, while blue represents a temperature of 0 K. Only an 84 μm $\times$ 84 μm portion of the computational grid is displayed for greater spatial resolution (see Fig. 9), since temperatures outside this region were almost uniform. The computational time step varied during the simulations. It was initially set to a value 2 Δt_1 (effective time step of Δt_1 per sweep) and, when there was a rapid temperature rise (due to chemical reactions), the time step was changed to 2 Δt_2. The wire, charge, and charge cup were assumed to be at the same initial temperature T_{in}. The thermal conductivity used in the simulation corresponds approximately to that of 99.5% pure, 98% dense alumina (see Fig. 7). Table 2 belows lists the parameters used in obtaining the plots for Figs. 10-15. The various contact resistivity values used are also shown in the table. Before and after subscripts refer to values prior to and following the melting of the bridgewire. The contact resistivity values for the nodes that are not mentioned in the table below were set to a high value of 10^{-3} m^2 K/W, making them effectively adiabatic.

Table 2 Parameters used in calculations of cases presented in Figs. 10 - 15

Case	Δt_1, Δt_2 (μs)	Firing Mode	T_{in} (K)	ρ_{before} (m^2 K/W), nodes	ρ_{after} (m^2 K/W), nodes
1	0.025,0.025	PIC	10	10^{-7}, all nodes	10^{-7}, all nodes
2	0.1,0.01	PIC	100	10^{-7}, all nodes	10^{-7}, all nodes
3	0.1,0.01	PIC	100	10^{-3}, all nodes	10^{-6}, 1-6 & 28-44
4	0.1,0.01	PIC	100	10^{-3}, all nodes	10^{-6}, 6-28
5	0.025, 0.025	PIC	10	10^{-7}, 38-40 10^{-9}, 12-22	10^{-3}, all nodes
6	0.025 0.025	PIC	10	10^{-7}, 6,28,39 10^{-9}, 12-22	10^{-3}, all nodes

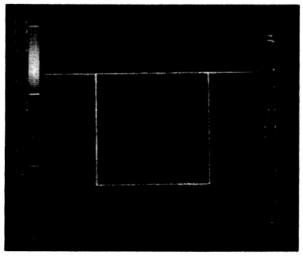
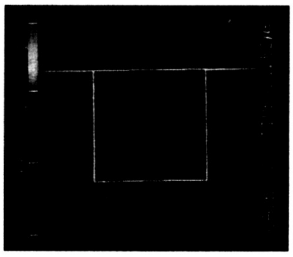
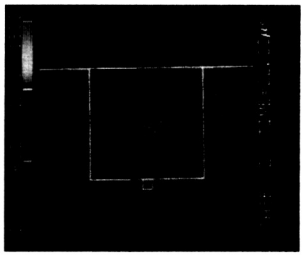
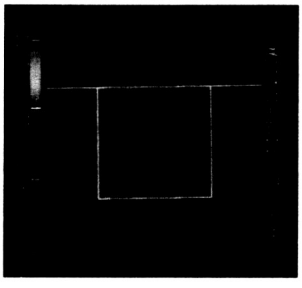
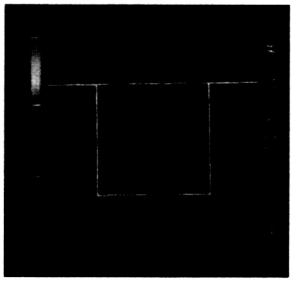
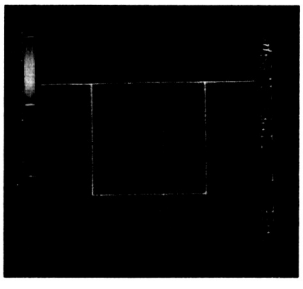
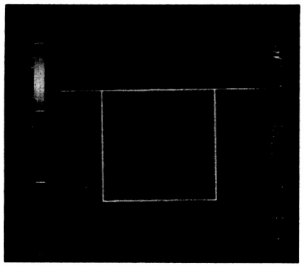
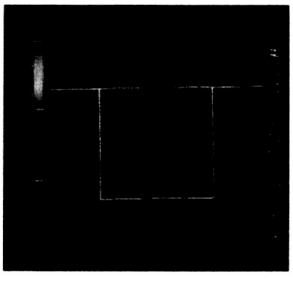
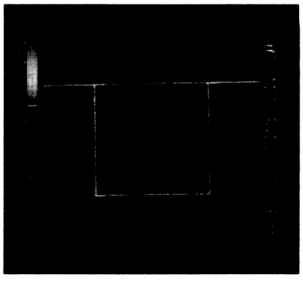
Figure 10 shows the simulation results for perfect thermal contact resistance between all of the materials when the initial temperature was 10 K. Ignition first occurred in the ZPP node furthest away from the charge cup along the center line of the wire (as expected from symmetry). The computed time to ignition was 24.25 μs; note the change of time step of the display after 18 μs. The temperature of the alumina adjacent to the bridgewire did not rise as much as the ZPP because heat is conducted away rapidly from the interface due to the high thermal conductivity of the alumina. In spite of the differences in thermal conductivity of the materials in contact with the wire, its temperature distribution was almost



centro-symmetric, and ignition on the sides of the wire occurred within 20 ns of the first ignition. This indicated that thermal diffusion from the stainless steel bridgewire is the rate limiting process (compare with the results below).

It should be noted that after ignition first occurred, ignition spread rapidly into the rest of the ZPP nodes. The extent of chemical reaction at each grid point was monitored and the energy release was limited to the consumption of the initial quantity of charge at that location. However, the phase changes that accompany the large heat release due to the chemical reactions were not modelled. The reacted zone was modelled as a very hot solid (temperatures reach ~15,000 K) with the specific heat and thermal conductivity of the original ZPP. Changes in the thermal transport properties corresponding to formation of ZrO_2 and KCl were neglected. Thus the calculations of the chain reaction after the first ignition were very qualitative. The product temperature was too high because phase changes were not modelled and the enthalpy of vaporization was not accounted for when computing product temperature. However, the calculations of the rate of propagation of the reaction are likely to be conservative, since the hot gaseous products of the chemical reaction would expand rapidly through the voids in the charge mixture. This would increase the speed of propagation of the reaction compared to thermal diffusion.

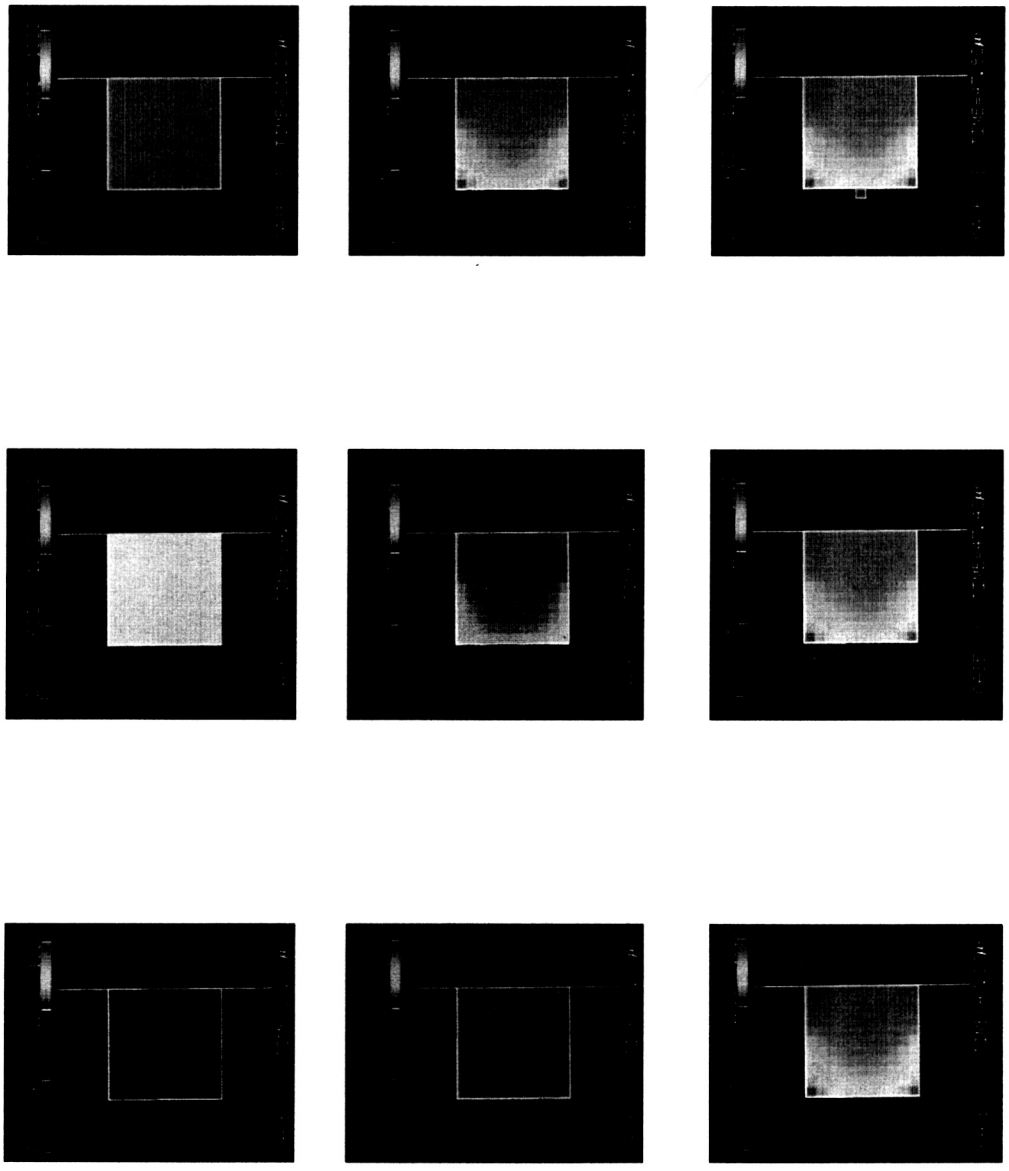
In the temperature plots for Case 2 presented in Fig. 11, contact resistivity values identical to those used in Case 1 were specified. However, the initial temperature of the system was increased to 100 K. This case is presented so that

the NSI behavior at 10 and 100 K initial temperatures can be compared. The model predicted ignition in 22 μs, a decrease of 2.25 μs (or about 9%) from the previous case. Ignition occurred at the same location as the previous case. These results imply that if intimate contact between the wire and the charge is maintained at low initial temperatures (10 K), the NSI should function reliably at that temperature.

The purpose of Cases 3 and 4, presented in Figs. 12 and 13 respectively, was to determine if the initial temperature of the system had an effect on the initiator performance when the wire wet different portions of the interface after it melted. In both cases the wire was assumed adiabatic prior to its melting. As mentioned in the previous section, the forces acting on the molten metal might cause different distributions of contact resistances. In this analysis, only two of the possibilities were studied.

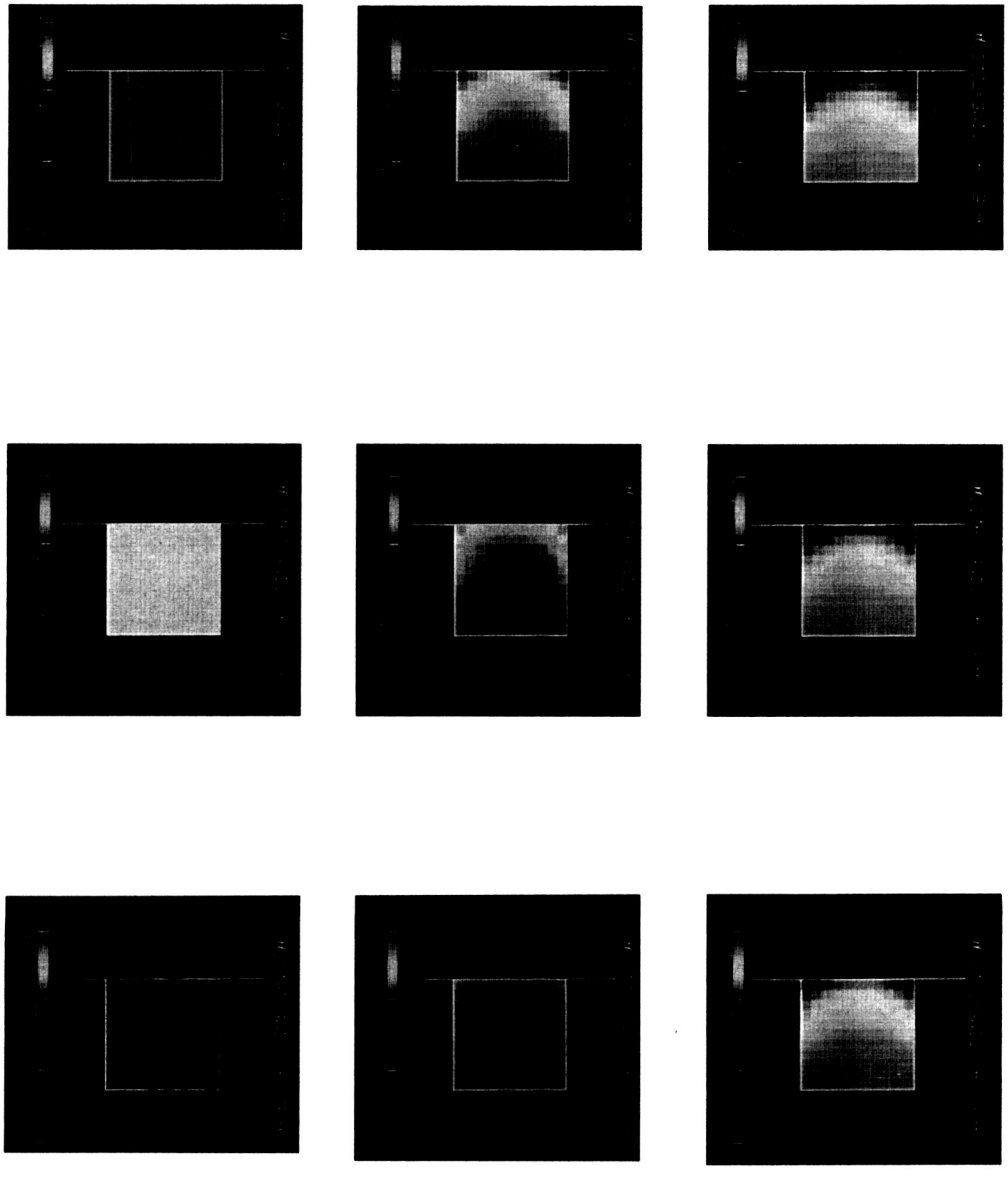
In both cases, the contact resistivity between the charge and the cup (ρ_{45}) was set to a moderate 10^{-7} m^2 K/W. As the false color plots indicate, the temperature distribution of the wire was uniform for both cases until the onset of the melting of the wire at 31 μs. After the wire melted, the molten wire was assumed to wet the charge cup and part of the ZPP in Case 3 (Fig.12), while the ZPP away from the wire was wet in Case 4 (Fig.13). The initiator failed to ignite in the former case, since most of the intimate contact between the hot molten metal was with the alumina. The heat was diffused out of the wire without causing a



temperature rise large enough to initiate the chemical reaction in the ZPP. However in Case 4, intimate contact was assumed with the ZPP only as illustrated in Fig. 13. The chemical decomposition of ZPP started at 46.60 μs, and ignition of the ZPP was achieved at 47.80 μs. Thus for an initial temperature of 100 K, the performance of the initiator is dependent on the portion of the interface that the wire wets after it had melted.

When the initial temperature was reduced to 10 K, results similar to those of Cases 3 and 4 were obtained. Similar computations at an initial temperature of 300 K showed that the performance of the initiator was independent of the distribution of contact resistivities after the wire melted.

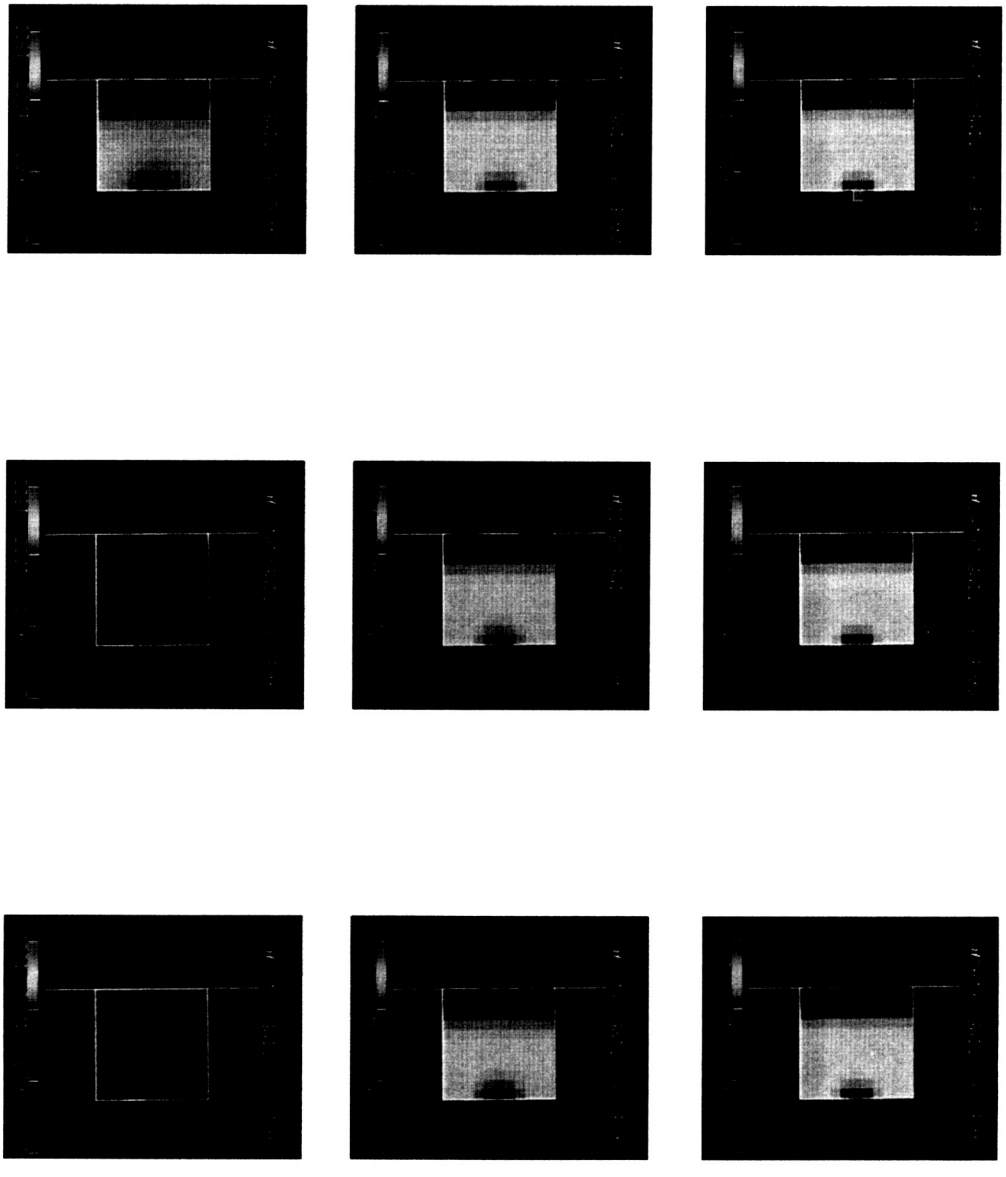
The purpose of Cases 5 and 6 was to illustrate the indirect dependence of the NSI performance on the overall power transfer coefficient γ. Results are shown in Figs. 14 and 15 respectively. In both cases the contact resistivity between the charge and the cup was set to 10^{-7} m^2 K/W. The heat loss from the wire to the cup is evident through the temperature distribution in the wire in the region near the alumina. This confirms our predictions that the heat loss to the cup is very substantial at the low initial system temperature of 10 K. In Case 5 the good contact between the wire and the charge mixture was in three adjacent nodes as shown in Fig. 14. The chemical decomposition reaction was initiated at 25.80 μs, and the initiator ignited at 26.80 μs. However initiator failure was obtained when the nodes of good thermal contact between the wire and the charge were assumed

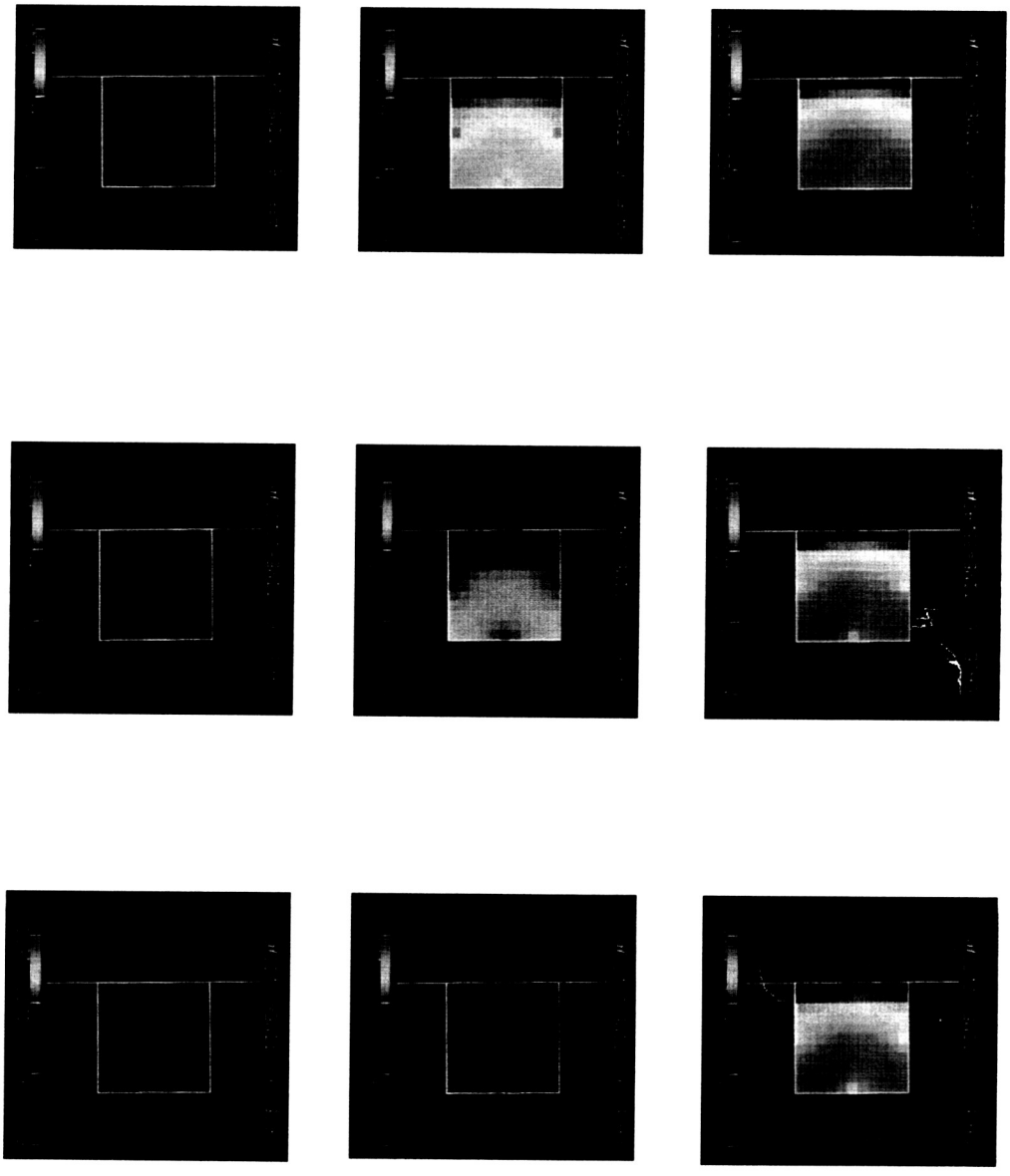


not to be adjacent. This caused the energy transferred to the charge to be diffused out since the hot regions were surrounded by cold ones. The wire melted at 33.40 μs, and the heat stored in the hot ZPP nodes was diffused out. The wire was assumed to adiabatic after it melted and thus no energy was transferred from the wire to the mix after melting. These two cases show that the initiator performance is not only dependent on γ, but also on the distribution of the areas making the good contact with the ZPP.

As the cases discussed above imply, several parameters affect the performance of the NSI. To help understand the role each of the parameters plays in the initiation of the NSI, a parametric analysis was conducted. This was accomplished by varying one of the parameters while holding the rest constant and then analyzing the results obtained. The parameters varied were: initial contact resistance (wire/surroundings) assuming wire makes same contact with whole interface, contact resistance between the wire and the alumina, location of nodes of intimate contact between the wire and the charge, and contact resistance after the wire melts. It was assumed that the wire wets whole interface after it melts.

Figure 16 shows the dependence of the time to ignition on the contact resistivity between the wire and the charge/charge cup for both PIC and SFU firing modes. The contact resistivity was assumed to be uniform across the interface, and heating of the interface was assumed to cease after the wire melted (contact resistivity along whole interface = 10^{-3} m^2 K/W). The initial temperature of the system was 100 K.





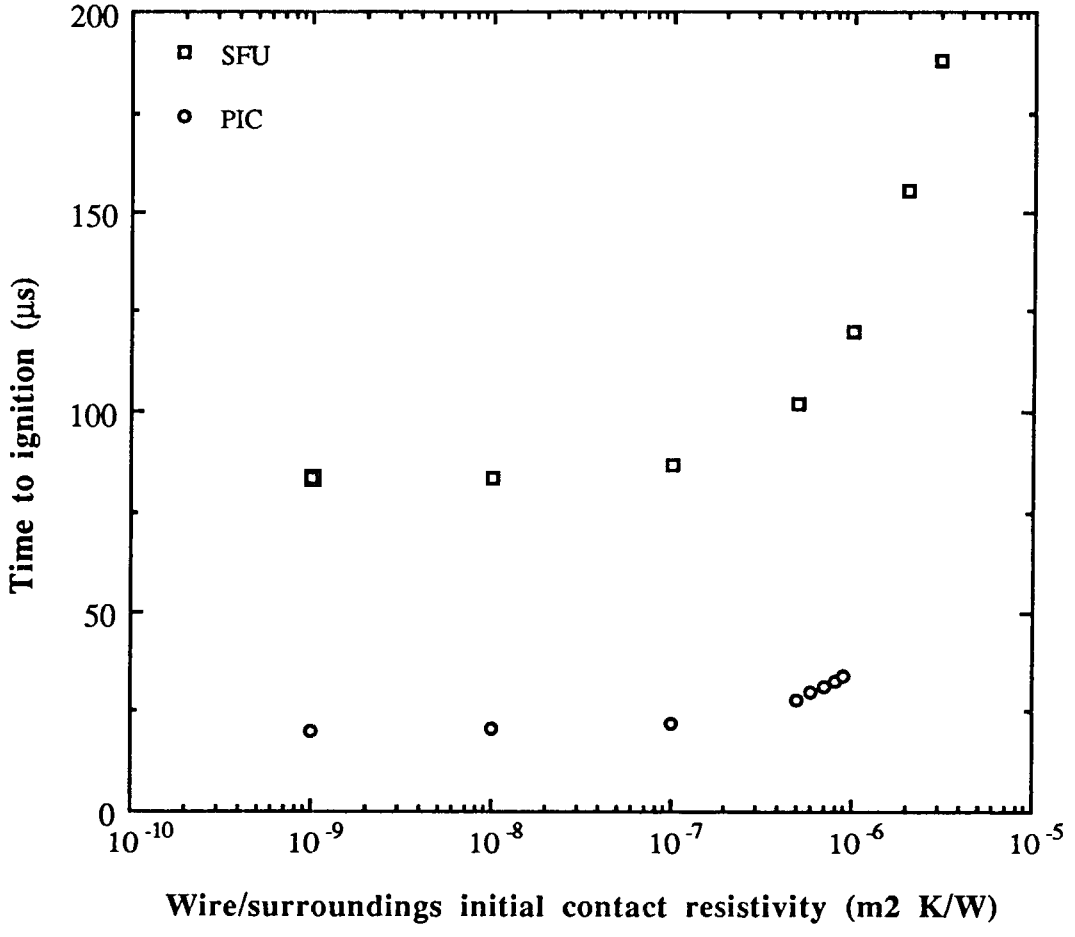


Figure 16 Time to ignition vs. initial contact resistivity between the wire and its surrounding. Wire was adiabatic after melting. T_{in}=100 K. Contact resistivity between ZPP/Al_2O_3 = 10^{-7} m^2 K/W.

First it can be observed that firing the initiator using the SFU mode takes longer than using the PIC. This is expected since the power delivered by the wire

in the PIC mode is significantly higher than in the SFU mode (see Fig. 2). The increase in time to ignition as the contact resistivity increases is also expected since the rate of energy transfer from the wire to the charge decreases; thus the charge takes longer to heat up. Figure 16 also suggests that the initiator will ignite over a wider range of initial contact resistances when the SFU mode is used. This is due again to the power delivered to the wire in each of the firing modes. The wire cannot transfer the energy fast enough when the PIC mode is used, since the power input to the wire is large for that mode. For the SFU mode however, the slower electrical power transfer to the wire allows it to diffuse its energy to the surrounding charge and cup.

Tests performed by NASA revealed a very high failure rate when the initiators were fired using the PIC mode at 22 K (85%). This rate was lower when the SFU mode was used to fire the initiators (42%). This result was very puzzling since the wire dissipated more power when the PIC mode was used. This can be explained by noting that even though more energy is input to the wire in the PIC mode, this energy is not necessarily transferred to the charge. The thermal contact resistance between the wire and the charge is dependent on the size of the gap between the two material. The gap size is a function of the coefficient of linear expansion of each of the material, which in turn is dependent on the temperature of each of the material.. Thus the contact resistance between the wire and the ZPP is a function of the temperature of both wire and ZPP. Since a larger fraction of the power dissipated by the wire is transferred to the charge when the SFU mode is used, the charge surrounding the wire will be hotter than that when the PIC mode is

used. Therefore the wire/charge contact resistance is improved more in the SFU mode than in the PIC mode. This may explain why initiators at the same initial temperatures and with identical initial contact resistance distribution have a performance that is dependent on the firing mode. It should be mentioned that in this model, all contact resistances were assumed independent of temperature.

Figure 17 is a plot of the time to ignition as a function of the initial contact resistivity along the wire/alumina interface for initial temperatures of 10 K, 100 K, and 300 K. The contact resistivity between the charge and cup was set to 10^{-7} m^2 K/W. After the wire melted, it was assumed to behave adiabatically with its surrounding (ρ was set to 10^{-3} m^2 K/W) for whole interface. Good contact between the wire and the ZPP ($\rho = 10^{-9}$ m^2 K/W) was assumed to occur away from the alumina (nodes 9-11). The PIC firing mode was simulated since NASA tests indicated that the initiator failure rate was the highest in this mode.

For an initial temperature of 300 K, the time to ignition was 16.60 μs when high contact resistivity along the wire/alumina interface (10^{-3} m^2 K/W) was assumed. The time increased to about 19.12 μs when a low contact resistivity value along that interface was used (10^{-9} m^2 K/W). When the initial system temperature was reduced to 100 K, the times were 25.44 μs and 29.92 μs for the high and low wire/alumina contact resistivities respectively. The plots for the times to ignition for initial temperatures of 300 K and 100 K shown in Fig. 17 above are almost parallel. This implies that reducing the initial system temperature from 300 K to 100 K simply shifted the time to ignition upward by the same amount for

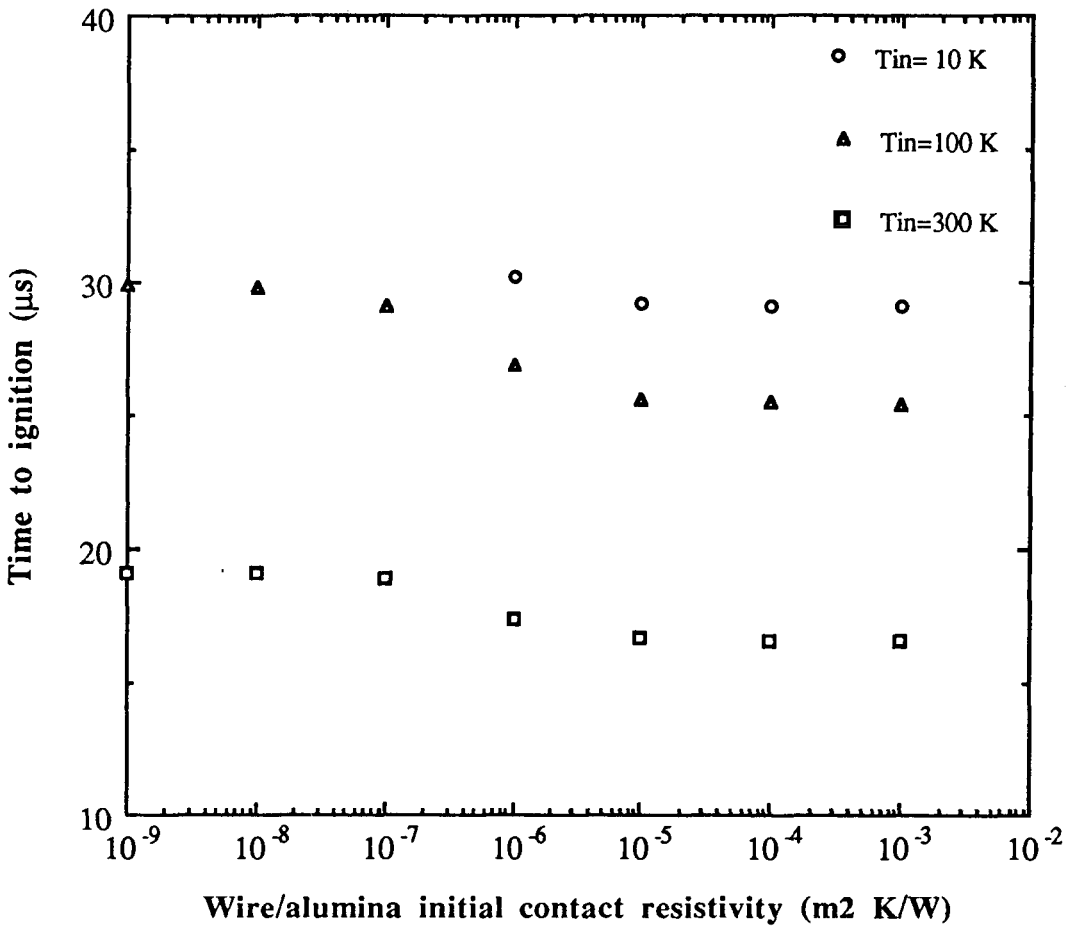


Figure 17 Time to ignition vs. contact resistivity between the wire and the alumina. PIC firing mode was used. Contact resistivity between SS/ZPP along faces 9-11 was 10^{-9} m^2 K/W. Rest of SS/ZPP contact resistivities 10^{-3} m^2 K/W. Wire was adiabatic after melting.

all the wire/alumina contact resistivities. This is not true for an initial temperature of 10 K. The time to ignition increased from 29.10 μs for high contact resistivity

along the interface (10^{-3} m^2 K/W), to about 30.23 μs when moderate resistivity values were used (10^{-6} m^2 K/W). When the resistivity was reduced further, the wire melted without initiating the chemical reaction in the charge. The maximum charge temperature was 628 K when the resistivity was 10^{-7} m^2 K/W. At this temperature the decomposition reaction of the potassium perchlorate was initiated, but it ceased after the wire melted. This value dropped to 615 K when the resistivity was reduced even further (10^{-9} m^2 K/W). These results show the significance of the heat loss to the alumina at very low temperatures. If the alumina were a thermal insulator at low temperatures, as previously assumed, then the plot of the time to ignition for the initial temperature of 10 K would be parallel to those at 100 and 300 K.

Figure 18 is similar to Fig. 17; it shows the time to ignition when the contact resistivity between the wire and the charge was increased from 10^{-9} m^2 K/W (Fig. 17) to 10^{-7} m^2 K/W (Fig. 18). For an initial temperature of 300 K, the time to ignition was 18.44 μs when high contact resistivity along the wire/alumina interface (10^{-3} m^2 K/W) was assumed. The time increased to about 21.64 μs when a low contact resistivity value along that interface was used (10^{-9} m^2 K/W). When the initial system temperature was reduced to 100 K, the time to ignition for the high wire/alumina resistivity (10^{-3} m^2 K/W) was 28.48 μs. The initiator failed to ignite when wire/alumina contact resistivity values smaller than 10^{-7} m^2 K/W were used. No ignition was achieved over the whole range of resistivity values when the initial temperature was further reduced to 10 K. When this set of results

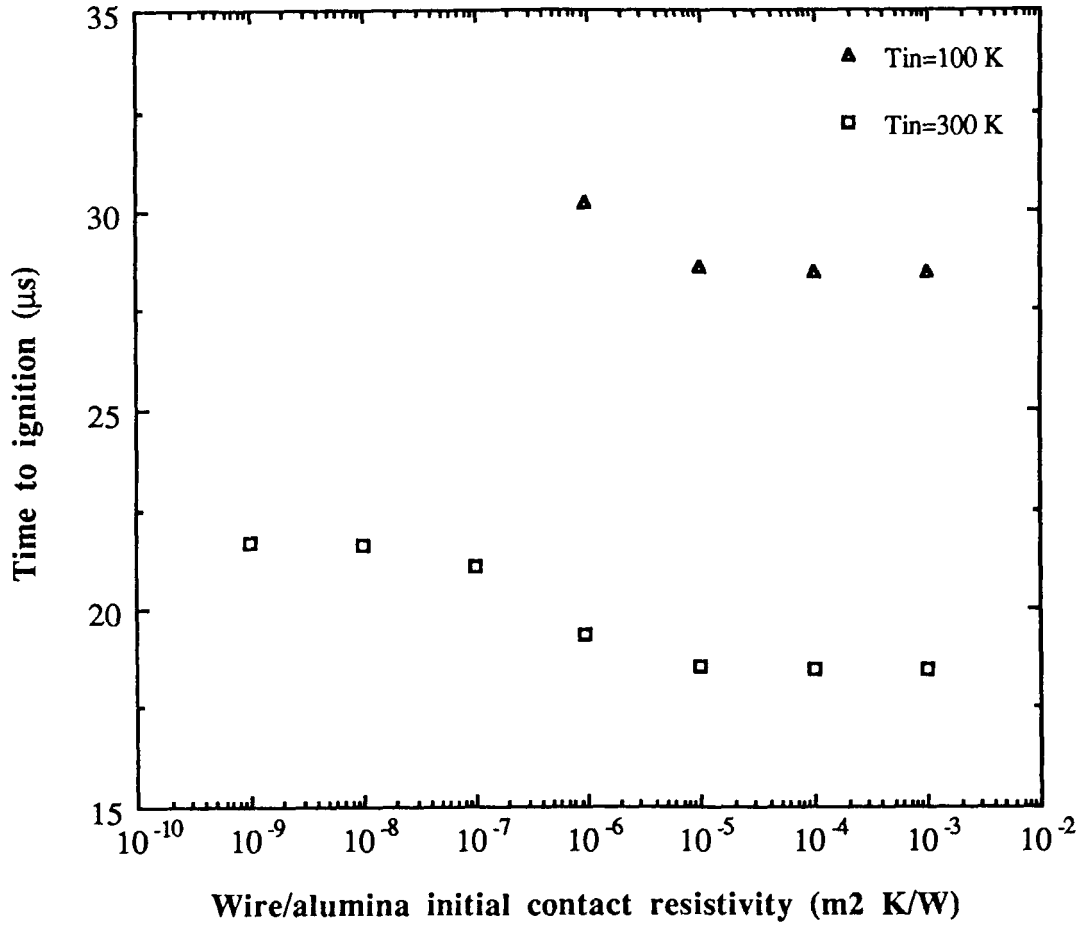


Figure 18 Time to ignition vs. contact resistivity between the wire and the alumina. PIC firing mode was used. Contact resistivity between SS/ZPP along faces 9-11 was 10^{-7} m^2 K/W. Rest of SS/ZPP contact resistivities 10^{-3} m^2 K/W. Wire was adiabatic after melting.

is compared to Fig. 17, it is evident that for the initiator to ignite at low initial temperatures, good contact along a portion of the wire/ZPP interface must be established.

The purpose of this analysis, results of which are presented in Fig. 19, was to determine whether the wire had enough thermal energy to initiate the chemical reaction in the ZPP after it had melted. The molten metal was assumed to make contact with over the whole charge interface. In Fig. 19, the time to ignition of the initiator is plotted against the contact resistivity between the wire and its surrounding charge after the wire has melted. The wire was assumed almost adiabatic initially (ρ was set to 10^{-3} m^2 K/W). Moderate contact resistivity (10^{-7} m^2 K/W) was specified over the charge/cup interface.

At an initial temperature of 10 K, and with the PIC firing mode used, the time to ignition was 33.18 μs for low contact resistivity (10^{-9} m^2 K/W). This time increased to 83.40 μs when the wire/charge contact resistivity was increased to 1.5×10^{-6} m^2 K/W. Contact resistivity values greater than 1.5×10^{-6} m^2 K/W lead to failure of the initiator. When the initial temperature was increased to 100 K, an ignition time of 32.50 μs was obtained when low contact (10^{-9} m^2 K/W) along the interface was specified. The time was 100.26 μs at the maximum resistance that lead to ignition (1.7×10^{-6} m^2 K/W).

When the SFU firing mode was used at an initial temperature of 10 K, the time to ignition was 125.32 μs for good contact (10^{-9} m^2 K/W). The time increased to 166.80 μs when poor contact resistivity was specified along the wire/charge interface (1.4×10^{-6} m^2 K/W). For an initial temperature of 100 K, these values ranged from 125.30 μs to 179.60 μs for low (10^{-9} m^2 K/W) and high

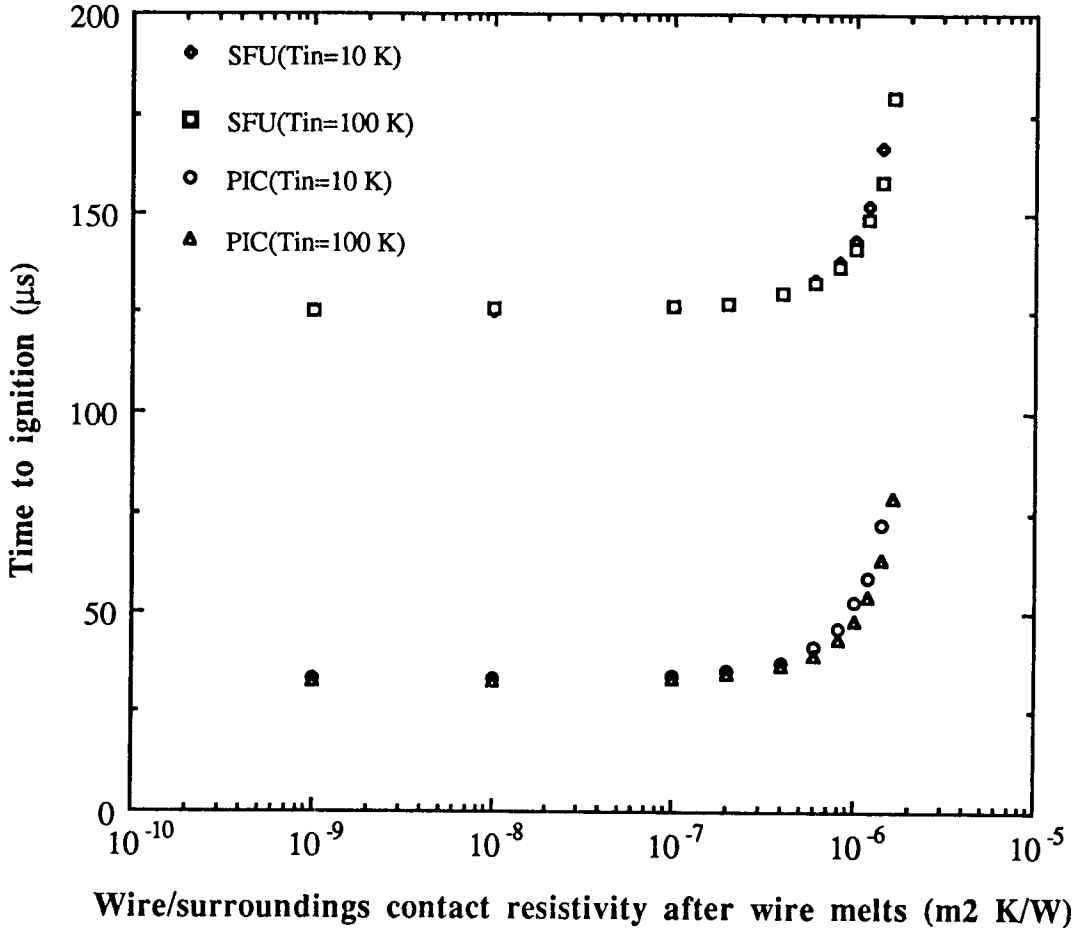


Figure 19 Time to ignition vs. contact resistivity between wire/surrounding after wire melts. Wire was assumed adiabatic initially ($\rho = 10^{-3}$ m^2 K/W) for all interface. ZPP/Al_2O_3 contact resistivity = 10^{-7} m^2 K/W.

(1.6×10^{-6} m^2 K/W) contact resistivities respectively. It is therefore obvious that the performance of the initiator is not significantly affected by firing mode when the initiator is assumed adiabatic initially.

As results of the parametric analysis show, the heat loss to the alumina plays a significant role in the performance of the NSI. The conclusions drawn from this analysis and the recommendations are presented in the following chapter.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

The thermal conductivity of alumina at low temperatures is a crucial factor that has been neglected in previous analyses (alumina was assumed to be a thermal insulator), and can explain many of the seemingly contradictory results of experimental measurements. For example, the lack of correlation between measured contact resistance and failure rate can be explained by noting that the bridgewire might be making good thermal contact with the alumina charge cup and not with the charge mixture. The electrothermal response test could not distinguish between these two cases. Because of the good thermal contact with the alumina, the bridgewire might take as long, or longer, to burn out since energy dissipated by the wire would be transferred to the alumina cup. Thus when the wire is not making good contact with the ZPP, it need not be behaving adiabatically as previously assumed. This explains why the bridgewire burnout times for both ignition and failure cases can be comparable, with the latter not initiating a reaction in the ZPP. Additionally, if the wire were making good contact with the ZPP in the area adjacent to the charge cup, the energy transferred to the ZPP might diffuse into the cup. This reduces the rate of temperature rise in the ZPP and might prevent the initiation of combustion before the wire melts. Further, chemical reactions initiated adjacent to the cup might be quenched due to heat diffusion into the cup.

Circumstantial evidence also supports this hypothesis. Initiators from one

vendor had a ZPP slurry brushed onto the bridgewire after it was welded to the pins in the charge cup. The slurry was then allowed to dry before pressing the pyrotechnic charge into the cup. Initiators with these "buttered" bridgewires were extremely reliable at low temperature, whereas those from another vendor which were not treated in this way were more unreliable. This can be explained by noting the dried slurry would insulate the bridgewire from the alumina charge cup and enhance heat transfer to a reactive mixture.

The results obtained show that changes in the contact resistivity between the wire and the charge of less than an order of magnitude can make the difference between successful firing and failure. Initiators with bridgewires which make excellent thermal contact with the alumina but poor contact with the ZPP will fail even though they have better total heat transfer coefficients. Measurements of overall heat transfer from the wire do not account for the material into which the energy is transferred. Thus they are not reliable predictors of performance of the initiator.

The modelling suggests manufacturing and/or design changes that will improve the reliability of initiators. Improving the thermal contact between the bridgewire and the pyrotechnic charge by "buttering" the bridgewire during manufacture will improve reliability. The bridgewire should not be stretched tightly across the connecting pins before welding, because differential contraction in the radial direction might draw it closer to the charge cup and improve thermal contact with it at low temperature.

A design change to reduce thermal conduction from bridgewire to the charge cup at low temperature would also improve reliability. The simplest means of accomplishing this is to reduce the thermal conductivity of the alumina at cryogenic temperatures. Since the thermal conductivity is very sensitive to packing density and impurities (see Fig. 7), the charge cup could be made from alumina with lower purity. The alumina presently used must meet or exceed certain density and purity specifications and is about 96% pure, 100% dense.[16] If the specifications are changed so that there is a *maximum* density and/or purity the thermal conductivity at low temperatures could be significantly reduced without compromising mechanical properties. This design modification has the advantage of using a cheaper material for the charge cup although this is unlikely to affect the overall cost of the initiator significantly.

If alumina with low enough thermal conductivity at low temperature is used for the charge cup it may permit discrimination of failure-prone initiators using the ETR test described in Chapter 3. The test will identify those initiators which do not make good thermal contact with the pyrotechnic mixture provided that the thermal conductivity of the charge cup is much lower than the pyrotechnic charge.

Appendix A

ESTIMATION OF RADIATIVE HEAT TRANSFER IN THE NSI

The purpose of the calculation in this appendix is to verify that radiative energy transfer between the wire and the charge is negligible.

If the wire contracts away from the charge at low temperatures then a gap develops between the wire and the charge. Radiation would then be the only means of energy transfer between the wire and the charge. Assuming that both wire and charge are black bodies, and assuming that the gap between the wire and the charge is small (so that the charge does not see itself), then the rate of energy transfer between the wire and the charge in the radiative mode is governed by the following equation:

$$\dot{Q}_{rad} = \varepsilon \sigma A (T_w^4 - T_c^4)$$

where ε is the emissivity, σ is the Stefan-Boltzmann constant (5.67×10^{-8} W/m^2K^4), and A is the surface area of the wire (4.71×10^{-7} m^2). The maximum value of the radiated power, $\dot{Q}_{rad,max}$, is obtained when the wire is at the highest possible temperature (melting temperature of stainless steel, 1700 K), the charge is at the lowest possible temperature (initial temperature of 10 K), and the emissivity is unity. When these values are substituted into the equation above, a value of 0.22 W is obtained for $\dot{Q}_{rad,max}$. As shown in Fig. 2, the power dissipated by the

wire in the 5 A CC mode is 26.25 W. Thus the actual power transfer from the wire to the charge in the purely radiative mode is always less than 1% of that dissipated by the wire when the 5 A CC firing mode is used. Since the 1% difference is well within the uncertainties of the calculations, the assumption that radiative transfer between the wire and the charge is negligible is justified.

Appendix B

THERMOPHYSICAL PROPERTIES OF NSI MATERIALS

This Appendix provides a listing of the curvefit equations for the thermophysical properties of Zr, $KClO_4$, AL_2O_3, and SS 304. The properties of the propellant were obtained by multiplying the properties by the corresponding stoichiometric coefficients (e.g., $K_{zpp} = 0.568 * K_{Zr} + 0.432* K_{KClO_4}$).

SPECIFIC HEAT c_p (J/kg K)

Zirconium[17]

$10 < T < 300$ K $\quad c_p = -1.093*10^{-9}T^5 + 8.767*10^{-7}T^4 - 2.360*10^{-4}T^3 + 0.0187T^2 + 1.7661T - 0.88$

$300 < T < 1100$ K $\quad c_p = 3.091*10^{-7}T^3 - 6.207*10^{-4}T^2 + 0.489T + 186.51$

$T > 1100$ K $\quad c_p = 395$

Potassium Perchlorate[18]

$10 < T < 260$ K $\quad c_p = -5.066*10^{-8}T^4 + 9.424*10^{-5}T^3 - 0.0435T^2 + 8.9371T - 77.16$

$260 < T < 1500$ K $\quad c_p = 1.335*10^{-12}T^5 - 6.802*10^{-9}T^4 + 1.362*10^{-5}T^3 - 0.0136T^2 + 7.3602T - 480.58$

$T > 1500$ K $\quad c_p = 1600$

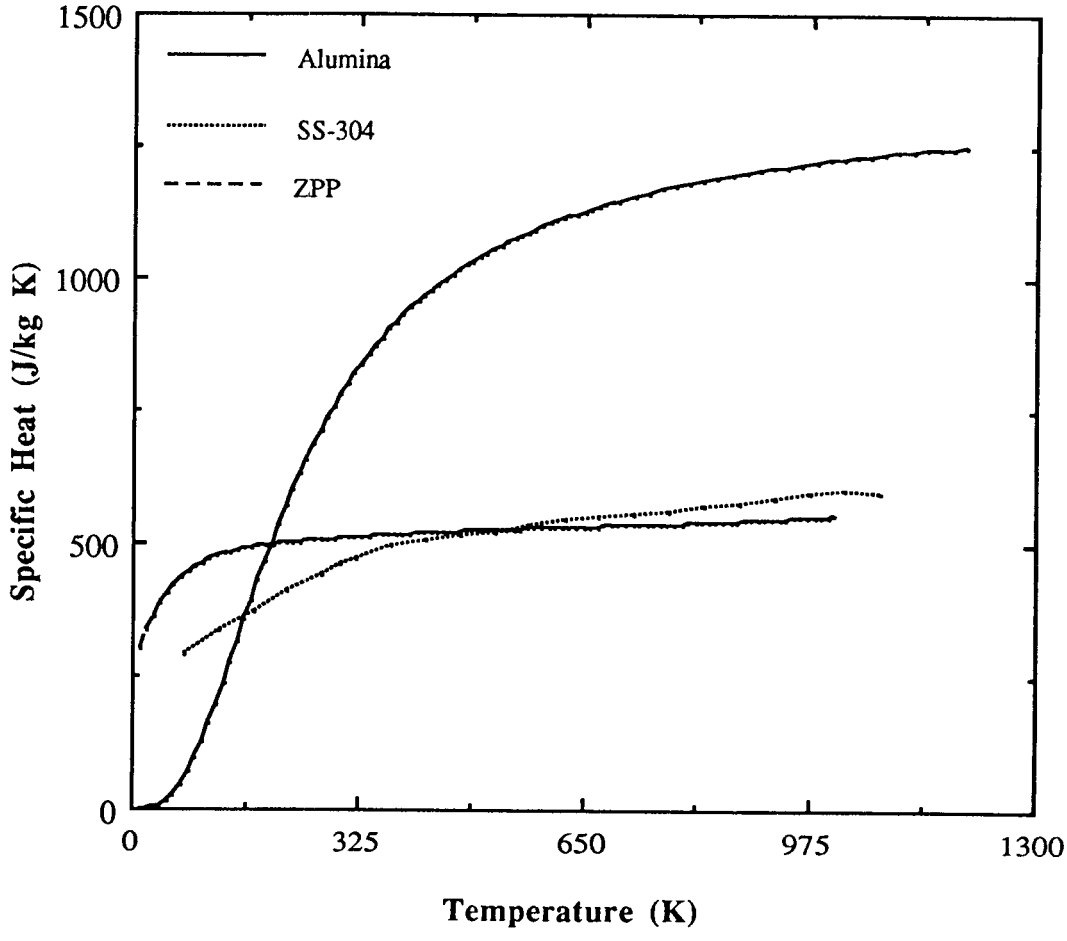


Figure 20 Specific Heats of ZPP, SS-304, and Alumina vs. temperature.

Alumina[19]

$10 < T < 200$ K $\qquad c_p = 4.643*10^{-9}T^5 - 2.727*10^{-6}T^4 + 5.047*10^{-4}T^3 - 0.0177T^2 + 0.249T - 0.89$

$200 < T < 1200$ K $\quad c_p = 2.927*10^{-12} T^5 - 1.259*10^{-8} T^4 + 2.168*10^{-5} T^3 - 0.019T^2 + 8.9272 T - 676.76$

$T > 1250$ K $\quad c_p = 1250$

Stainless Steel[20]

$10 < T < 1070$ K $\quad c_p = -2.064*10^{-12} T^5 + 5.319*10^{-9} T^4 - 4.2*10^{-6} T^3 + 3.962*10^{-4} T^2 + 0.9225 T + 220.22$

$T > 1070$ K $\quad c_p = 595$

THERMAL CONDUCTIVITY k (W/m K)

Zirconium[21]

$10 < T < 40$ K $\quad k = -0.1117T^2 + 4.25 T+67.67$

$40 < T < 300$ K $\quad k = 286.8429T^{-0.4552}$

$300 < T < 1500$ K $\quad k = -1.219*10^{-8}T^3 + 4.389*10^{-5}T^2 - 0.0401T+ 31.61$

$T > 1500$ K $\quad k = 25$

Potassium Perchlorate[22]

No data on the thermal conductivity of $KCLO_4$ were found. It was then assumed to be approximately that of KNO_3.

$k = 1.0$ for all T

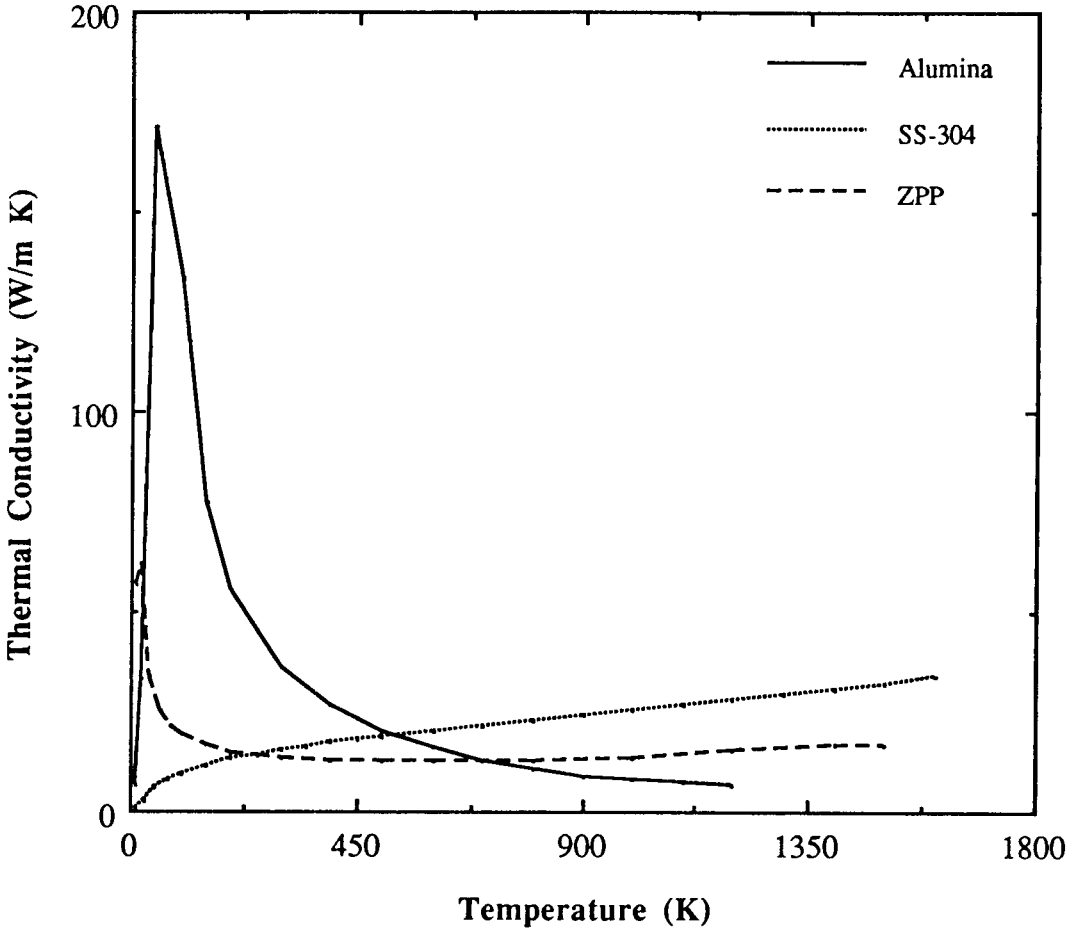


Figure 21 Thermal Conductivity of ZPP, SS-304, and Alumina vs. temperature.

Alumina[23]

$10 < T < 50$ K $k = 0.049\ T^2 + 1.176\ T - 9.94$

$40 < T < 300$ K $k = 1.870*10^{4}\ T^{-1.1134}$

$T > 1500$ K $k = 6$

Stainless Steel[24]

$10 < T < 1660$ K $k = 3.952*10^{-14}\ T^{5} - 1.816*10^{-10}\ T^{4} + 3.108*10^{-7}\ T^{3} - 2.432*10^{-4}\ T^{2} + 0.0991\ T + 0.88$

$T > 1500$ K $k = 34.7$

Appendix C

HEAT DIFFUSION EQUATION FOR TWO DIMENSIONAL MODEL

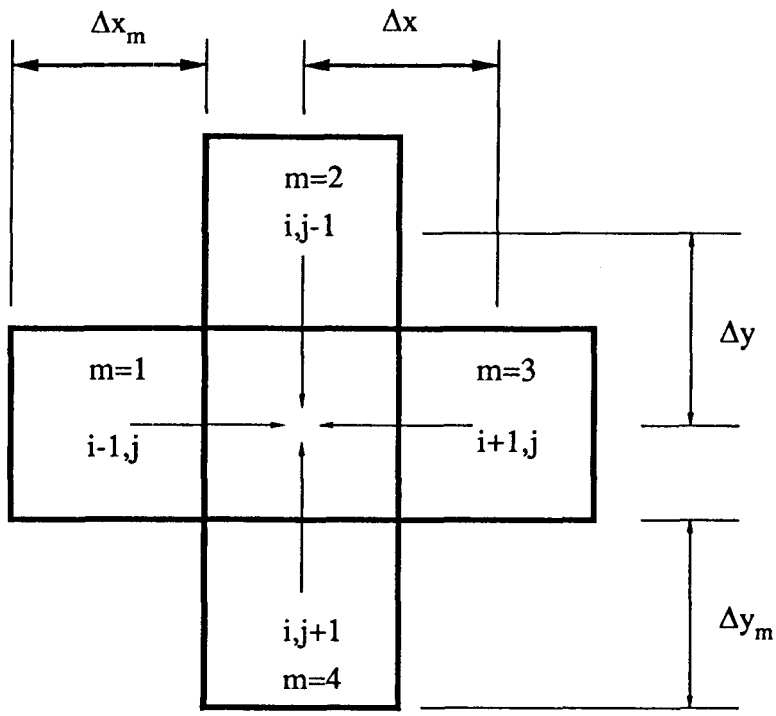


Figure 22 Control volumes and heat flow directions for two-dimensional model.

Applying the conservation of energy equations on the control volumes shown in Fig. 22 above we get:

$$\dot{Q}_{in} - \dot{Q}_{out} + \dot{Q}_{gen} = \dot{Q}_{stored}$$

According to our heat flow direction above $\dot{Q}_{out}=0$.

$\dot{Q}_{gen} = \dot{Q}_{el} + \dot{Q}_{chem}$; where $\dot{Q}_{el} = RI^2$, and $\dot{Q}_{chem} = \Delta H\ V_{cv}\ Ar\ e^{(-E_a/R_uT)} =$ Gen V_{cv}.

$$\dot{Q}_{stored} = \rho_m\, c_p\, V \frac{\Delta T}{\Delta t}$$

$$\dot{Q}_{in} = \sum_{m=1}^{4} k_{m-(i,j)} A_m \frac{T_m - T_{i,j}}{\Delta l}$$

The heat transfer areas are defined by: $A_1 = A_3 = \Delta y_m\, L$, and $A_2 = A_4 = \Delta x_m\, L$.

Expanding the above equation, we get for implicit in x, explicit in y:

$$-\frac{\Delta y_m}{\Delta x} k_{1\text{-}i,j} T^{n+1}_{i-1,j} + \left(\frac{1}{\lambda} + \frac{\Delta y_m}{\Delta x} k_{1\text{-}i,j} + \frac{\Delta y_m}{\Delta x} k_{3\text{-}i,j}\right) T^{n+1}_{i,j} - \frac{\Delta y_m}{\Delta x} k_{3\text{-}i,j} T^{n+1}_{i+1,j} =$$

$$\frac{\Delta x_m}{\Delta y} k_{2\text{-}i,j} T^{n}_{i,j-1} + \left(\frac{1}{\lambda} - \frac{\Delta x_m}{\Delta y} k_{2\text{-}i,j} - \frac{\Delta x_m}{\Delta y} k_{4\text{-}i,j}\right) T^{n}_{i,j} + \frac{\Delta x_m}{\Delta y} k_{4\text{-}i,j} T^{n}_{i,j+1} + \frac{RI^2}{L} + \text{Gen}\ A_{cv}$$

and for implicit in y, explicit in x:

$$-\frac{\Delta x_m}{\Delta y} k_{2\text{-}i,j} T^{n+2}_{i,j-1} + \left(\frac{1}{\lambda} + \frac{\Delta x_m}{\Delta y} k_{2\text{-}i,j} + \frac{\Delta x_m}{\Delta y} k_{4\text{-}i,j}\right) T^{n+2}_{i,j} - \frac{\Delta x_m}{\Delta y} k_{4\text{-}i,j} T^{n+2}_{i,j+1} =$$

$$\frac{\Delta y_m}{\Delta x} k_{1\text{-}i,j} T^{n+1}_{i-1,j} + \left(\frac{1}{\lambda} - \frac{\Delta y_m}{\Delta x} k_{1\text{-}i,j} - \frac{\Delta y_m}{\Delta x} k_{3\text{-}i,j}\right) T^{n+1}_{i,j} + \frac{\Delta y_m}{\Delta x} k_{3\text{-}i,j} T^{n+1}_{i+1,j} + \frac{RI^2}{L} + \text{Gen}\ A_{cv}$$

where $\lambda = \dfrac{\Delta t}{\rho_m\, c_p\, A_{cv}}$

Appendix D

EQUIVALENT THERMAL RESISTANCES FOR TWO DIMENSIONAL MODEL

Consider the two adjacent control volumes shown in the figure below.

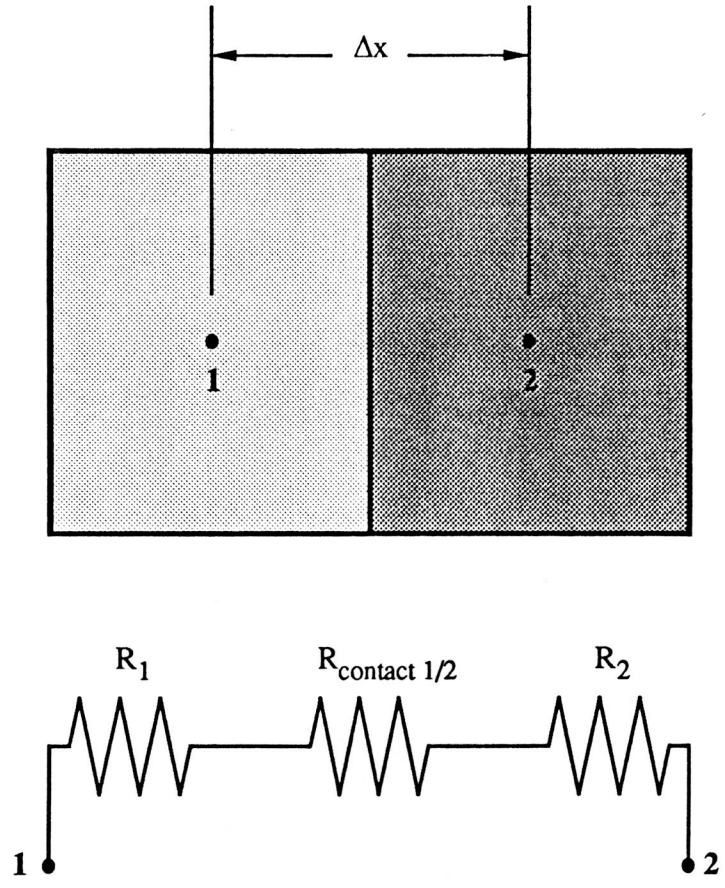


Figure 23 Thermal resistances including conduction and contact for two adjacent nodes.

The equivalent thermal resistance for the adjacent control volumes **1** and **2** shown in Fig. 23 above needs to be determined.

The various resistances are defined by

$$R_1=\frac{\Delta x/2}{k_1 A} \qquad R_{1/2}=\frac{\rho_{1/2}}{A} \qquad R_2=\frac{\Delta x/2}{k_2 A} \qquad R_{eq}=\frac{\Delta x/2}{k_{eq} A}$$

where

Δx is the distance between the centers of the control volumes.

k_1 is the thermal conductivity of the material in control volume one.

k_2 is the thermal conductivity of the material in control volume two.

k_{eq} is the equivalent thermal conductivity of the region between the centers of the adjacent control volumes.

R_{eq} is the equivalent thermal resistance of the region between the centers of the adjacent control volumes.

$\rho_{1/2}$ is the thermal contact resistivity of the interface.

A is the area of the interface.

The equivalent resistance of resistances in series is equal to the sum of the resistances. thus,

$$R_{eq}=\sum_{i=1}^{n} R_i \Rightarrow R_{eq}= R_1+ R_{1/2}+ R_2= \frac{1}{A}\left[\frac{\Delta x}{2 k_1} + \rho_{1/2} + \frac{\Delta x}{2 k_2}\right]$$

then

$$\frac{\Delta x}{k_{eq} A} = \frac{1}{A}\left[\frac{\Delta x \left\langle k_1 + k_2 \right\rangle + 2 k_1 k_2 \rho_{1/2}}{2 k_1 k_2}\right] \Rightarrow k_{eq}= \frac{2 k_1 k_2 \Delta x}{\Delta x \left\langle k_1 + k_2 \right\rangle + 2 k_1 k_2 \rho_{1/2}}$$

Appendix E

TWO DIMENSIONAL CODE

```
      PROGRAM APPROX(DBGDAT7, OUTDAT, TTY, SCRAP2, CONIN,
     2    TAPE5=DBGDAT7, TAPE6=OUTDAT, TAPE1=TTY,
     3    TAPE8=SCRAP2, TAPE9=CONIN,OUTPUT)
C
      DIMENSION TSTAR(81,71),T(81,71),QAVAIL(81,71),
     2    QREL(81,71),Q(81,71),QEL(81,71),A(81),B(81),CC(81),D(81),
     3    TT(81),C(5),RES(45),CONRES(81,71,4),CON(81,71,4),
     4    CVCP(81,71),CVRHO(81,71)
C
C     ARRAY TSTAR CONTAINS THE TEMPERATURE VALUES OF THE
C        NODES IN THE MODEL AT THE END OF THE FIRST SWEEP OF
C        THE ADI ITERATION.
C
C     ARRAY T CONTAINS THE TEMPERATURE VALUES OF THE
C        NODES IN THE MODEL AT THE END OF THE SECOND SWEEP
C        OF THE ADI ITERATION.
C
C     ARRAY QAVAIL CONTAINS UPDATED VALUES OF THE
C        THERMOCHEMICAL ENERGY AVAILABLE FOR EACH NODE.
C
C     ARRAY Q IS A WORKING ARRAY THAT CONTAINS THE
C        INTERMEDIATE VALUES OF THE RELEASED
C        THERMOCHEMICAL ENERGY.
C
C     ARRAY QREL CONTAINS THE UPDATED VALUES OF THE
C        RELEASED ENERGY FOR EACH NODE.
C
C     ARRAY QEL CONTAINS THE ELECTRICAL ENERGY GENERATED
C        BY EACH NODE.
C
C     VECTORS A,B,CC,AND D ARE WORKING VECTORS THAT STORE
C        THE VALUES OF THE COEFFICIENTS OF THE TRIDIAGONAL
C        MATRIX GENERATED.  TT IS THE SOLUTION VECTOR.
C
C     VECTOR C IS A WORKING VECTOR THAT STORES THERMAL
C        CONDUCTIVITY VALUES OF A CENTER NODE AND ITS FOUR
C        ADJACENT NEIGHBORS.
C
```

```
C     VECTOR RES CONTAINS CONTACT RESISTIVITY VALUES FOR
C
C        THE WIRE/CUP, WIRE/CHARGE, AND CHARGE/CUP
C        INTERFACES.
C
C     ARRAY CONRES CONTAINS THE CONTACT RESISTIVITY
C        VALUES OF EACH NODE WITH ALL FOUR OF ADJACENT
C        NODES.
C
C     ARRAY CON CONTAINS THE VALUES OF THE EQUIVALENT
C        THERMAL CONDUCTIVITY (INCLUDING CONTACT
C        RESISTANCE) OF ALL NODES.
C
C     ARRAYS CVCP AND CVRHO CONTAIN SPECIFIC HEAT AND
C        MASS DENSITY VALUES (RESPECTIVELY) FOR ALL NODES.
C
      COMMON X(81),Y(71),AREA(81,71),XM(81),YM(71)
C
C     ARRAYS X AND Y CONTAIN THE ABSCISSAS AND ORDINATES
C        (RESPECTIVELY) OF THE CENTERS OF THE CONTROL
C        VOLUMES.
C
C     ARRAYS XM AND YM CONTAIN THE ABSCISSAS AND
C        ORDINATES (RESPECTIVELY) OF THE BOUNDARIES OF THE
C        CONTROL VOLUMES.
C
C     ARRAY AREA CONTAINS THE AREA  OF EACH NODE.
C
C
C
C*************************************************************
C                                                            *
C     READ THE INPUT PARAMETERS                              *
C                                                            *
C*************************************************************
C     DT1 IS DELTA TIME USED (S).
      READ(5,*)DT1
C
C     DT2 IS SECOND DELTA TIME USED.
      READ(5,*)DT2
C
C     EPV IS THE AVAILABLE THERMOCHEMICAL ENERGY PER UNIT
C     VOLUME (J/M3).
      READ(5,*)EPV
C
C     TMPSS IS THE MELTING TEMPERATURE OF SS-304(K).
      READ(5,*)TMPSS
C
```

```
C       TIG IS THE CRITICAL TEMPERATURE ABOVE WHICH REACTION
C          STARTS.
        READ(5,*)TIG
C
C       SS RHO IS THE DENSITY OF STAINLESS STEEL (KG/M3).
        READ(5,*)SSRHO
C
C       ALRHO IS THE DENSITY OF ALUMINA (KG/M3).
        READ(5,*)ALRHO
C
C       ZRRHO IS THE DENSITY OF ZIRCONIUM (KG/M3).
        READ(5,*)ZRRHO
C
C       RLEN IS THE LENGTH OF THE WIRE (M).
        READ(5,*)RLEN
C
C       NTCUR REPRESENTS THE FIRING MODE CHOSEN (1=CC, 2=PIC,
C       3=SFU).
        READ(5,*)NTCUR
C
C       TIN IS THE INITIAL TEMPERATURE OF THE SYSTEM (K).
        READ(5,*)TIN
C
C       AA IS THE PREEXPONENTIAL FACTOR IN THE HEAT RELEASE
C       TERM.
        READ(5,*)AA
C
C       EA IS THE ACTIVATION ENERGY FOR THE REACTION DIVIDED
C       BY R (K).
        READ(5,*)EA
C
C       KAL IS THE KIND OF ALUMINA TO BE USED.(1=98%,2=100%
PURE).
        READ(5,*)KAL
C
C       MITER IS A FLAG USED FOR RESULT PRINTOUT.
        READ(5,*)MITER
C
        FLAG=1.0
        TFLAG=0.0
        TPMAX=0.0
C
C       RE IS THE RESISTANCE OF THE WIRE (OHMS).
        RE=1.05
C
C*************************************************************
C                                                            *
C       ECHO PRINT BLOCK                                     *
```

```
C                                                                      *
C**********************************************************************
C
      WRITE(6,9970)DT,EPV,TMPSS,SSRHO,ALRHO,ZRRHO,RLEN
 9970 FORMAT(2X,'DT USED=',E15.8,/,2X,'THEMOCHEMICAL ENERGY
     2    CONTENT =',E15.8,' J/M3',/,2X,'MELTING TEMPERATURE OF
     3    THE WIRE=',F10.3,' K',/,2X,'DENSITY OF THE WIRE=',F10.3,'
     4    KG/M3',/,2X,'DENSITY OF THE ALUMINA=',F10.3,'
     5    KG/M3',/,2X,'DENSITY OF THE ZPP MIX=',F10.3,'
     6    KG/M3',/,2X,'LENGTH OF THE WIRE =',F10.7,' M')
      IF(NTCUR.EQ.1) THEN
         WRITE(6,9971)
 9971    FORMAT(2X,'FIRING MODE USED WAS CONSTANT
C        CURRENT')
      ELSEIF(NTCUR.EQ.2) THEN
         WRITE(6,9972)
 9972    FORMAT(2X,'FIRING MODE USED WAS PIC')
      ELSE
         WRITE(6,9973)
 9973    FORMAT(2X,'FIRING MODE USED WAS SFU')
      ENDIF
      IF(KAL.EQ.1) THEN
         WRITE(6,9974)
 9974    FORMAT(2X,'98% PURE ALUMINA WAS USED')
      ELSE
         WRITE(6,9975)
 9975    FORMAT(2X,'100% PURE ALUMINA WAS USED')
      ENDIF
C
C
C**********************************************************************
C                                                                      *
C     DETERMINE THE MESH COORDINATES                                   *
C                                                                      *
C**********************************************************************
C
      CALL MESH(IXCOUNT,IYCOUNT,LXWIRE,LYWIRE,MM)
C
C     NX IS THE NUMBER OF NODES IN THE X DIRECTION.
      NX=IXCOUNT-1
C
C     NY IS THE NUMBER OF NODES IN THE Y DIRECTION.
      NY=IYCOUNT-1
C
C     MX IS THE NODE NUMBER OF THE CENTER OF THE WIRE N THE
C     X DIRECTION.
      MX=LXWIRE
C
```

```
C       MY IS THE NODE NUMBER OF THE ENTER OF THE WIRE IN THE
C       Y DIRECTION.
C
        MY=LYWIRE
        WRITE(1,887)NX,NY,MX,MY
        WRITE(6,887)NX,NY,MX,MY
 887    FORMAT(2X,'NUMBER OF NODES IN X DIRECTION=',I3,/,
     2      2X,'NUMBER OF NODES IN Y DIRECTION =',I3,/,2X,'NODE
     3      NUMBER OF CENTER OF WIRE(X)=',I3,/,2X,'NODE NUMBER
     4      OF CENTER OF WIRE(Y)=',I3)
C
        ML=MX-MM
        MR=MX+MM
        MT=MY-MM
        MB=MY+MM
        WRITE(6,898)ML,MR,MT,MB
 898    FORMAT(1X,'ML=',I3,1X,'MR=',I3,1X,'MT=',I3,1X,'MB=',I3)
C
C*************************************************************
C                                                            *
C       SELECT THE NODES TO BE PLOTTED                       *
C                                                            *
C*************************************************************
C
        IX1=MX-10
        IX2=MX+10
        IY1=MY-10
        IY2=MY+10
C
C*************************************************************
C                                                            *
C       INITIALIZE THE MATRICES                              *
C                                                            *
C*************************************************************
C
        DO 10 I=1,NX
            A(I)=0.
            B(I)=0.
            CC(I)=0.
            D(I)=0.
            DO 20 J=1,NY
                QAVAIL(I,J)=0.0
                QREL(I,J)=0.0
                T(I,J)=TIN
                TSTAR(I,J)=TIN
                QEL(I,J)=0.0
                DO 20 K=1,4
                    CONRES(I,J,K)=0.0
```

```
 20        CONTINUE
 10     CONTINUE
C
C*************GET THE CONTACT RESISTANCE MATRIX*************
C
C       RESMAIN IS THE MAIN CONTACT RESISTANCE.
        READ(5,*)RESMAIN
C
C       IRFLAG IS THE FLAG TO INDICATE THE LOCATION OF THE
C       GOOD CONTACT RESISTANCE AFTER THE WIRE MELTS (1=TOP,
C       2=BOTTOM, 3=ALL).
        READ(5,*)IRFLAG
C
C       RESAFT IS THE CONTACT RESISTANCE OF NODES SPECIFIED BY
C       IRFLAG AFTER THE WIRE MELTS.
        READ(5,*)RESAFT
C
C       NREC IS THE NUMBER OF CONTACT RESISTANCE RECORDS TO
C       FOLLOW.
        READ(5,*)NREC
C
        DO 414 I=1,45
           RES(I)=RESMAIN
 414    CONTINUE
C
        IF(NREC.NE.0) THEN
           DO 415 I=1,NREC
              READ(5,*)KKK,RES(KKK)
 415       CONTINUE
        ENDIF
C
        CALL CONTACT(NX,NY,ML,MR,MT,MB,RES,CONRES,RESFLAG)
C
C****************GET THE MASS DENSITY MATRIX.****************
C
        DO 350 I=1,NX
           DO 351 J=1,NY
              IF(I.LT.ML) THEN
                 CVRHO(I,J)=ZRRHO
              ELSEIF((I.GE.ML).AND.(I.LE.MR)) THEN
                 IF((J.GE.MT).AND.(J.LE.MB)) THEN
                    CVRHO(I,J)=SSRHO
                 ELSE
                    CVRHO(I,J)=ZRRHO
                 ENDIF
              ELSEIF(I.GT.MR) THEN
                 CVRHO(I,J)=ALRHO
              ENDIF
```

```
 351        CONTINUE
 350    CONTINUE
C
C********FIND THE AVAILABLE THERMOCHEMICAL ENERGY********
C
        DO 38 I=1,MR
            DO 44 J=1,NY
                IF((J.GE.MT).AND.(J.LE.MB).AND.(I.GE.ML)) THEN
                    QAVAIL(I,J)=0.0
                ELSE
                    QAVAIL(I,J)=EPV*AREA(I,J)
                ENDIF
 44         CONTINUE
 38     CONTINUE
C
        RESFLAG=0.0
        TIME=0.0
C
        ITER=0
C
C**********************START THE ITERATION**********************
C
 22     ITER=ITER+1
C
C******DETERMINE TIME STEP TO BE USED IN THIS ITERATION******
C
            IF(TFLAG.NE.1.0) THEN
                CALL DETDT(DT1,DT2,DT,TPMAX,TIG,TFLAG)
            ENDIF
C
            WRITE(1,*)'ITERATION NUMBER',ITER
C
C*****************************************************************
C                                                                *
C       FIND THE THERMOCHEMICAL ENERGY RELEASED                  *
C                                                                *
C*****************************************************************
C
            DO 46 I=1,MR
                DO 47 J=1,NY
                    IF((J.GE.MT).AND.(J.LE.MB).AND.(I.GE.ML)) THEN
                        Q(I,J)=0.0
                    ELSE
                        Q(I,J)=AREA(I,J)*GEN(T(I,J),AA,EA)
                    ENDIF
 47             CONTINUE
 46         CONTINUE
C
```

```
C*************************************************************
C                                                            *
C      UPDATE THE AVAILABLE ENERGY                           *
C                                                            *
C*************************************************************
C
          DO 48 I=1,MR
             DO 49 J=1,NY
                QQ=Q(I,J)*DT
                IF(QQ.GE.QAVAIL(I,J)) THEN
                   QREL(I,J)=QAVAIL(I,J)/DT
                   QAVAIL(I,J)=0.0
                ELSE
                   QREL(I,J)=Q(I,J)
                   QAVAIL(I,J)=QAVAIL(I,J)-QQ
                ENDIF
 49          CONTINUE
 48       CONTINUE
C
C*************************************************************
C                                                            *
C      FIND THE MAXIMUM TEMPERATURE IN THE WIRE              *
C                                                            *
C*************************************************************
C
          TWMAX=0.0
          DO 55 I=ML,MR
             DO 65 J=MT,MB
                IF (T(I,J).GT.TWMAX) THEN
                   TWMAX=T(I,J)
                ELSE
                   GOTO 65
                ENDIF
 65          CONTINUE
 55       CONTINUE
          WRITE(1,*)'TWMAX=',TWMAX
C
C*************************************************************
C                                                            *
C      FIND THE VALUE OF THE CURRENT                         *
C                                                            *
C*************************************************************
C
          IF((TWMAX.GE.TMPSS).OR.(FLAG.EQ.0.0)) THEN
             AMP=0.0
             FLAG=0.0
             WRITE(1,*)'THE WIRE HAS MELTED'
C
```

```
C*************************************************************
C                                                            *
C      IF DESIRED, CHANGE ANY OF THE WIRE /SURROUNDING       *
C      CONTACT RESISTIVITIES  AFTER THE WIRE MELTS.          *
C                                                            *
C*************************************************************
C
              IF(RESFLAG.EQ.0.0) THEN
                 IF(IRFLAG.EQ.1) THEN
                    DO 431 I=1,6
                       RES(I)=RESAFT
 431                CONTINUE
                    DO 432 I=28,44
                       RES(I)=RESAFT
 432                CONTINUE
                    ELSEIF(IRFLAG.EQ.2) THEN
                       DO 433 I=6,28
                          RES(I)=RESAFT
 433                   CONTINUE
                    ELSEIF(IRFLG.EQ.3) THEN
                       DO 434 I=1,44
                          RES(I)=RESAFT
 434                   CONTINUE
                    ENDIF
                    CALL CONTACT(NX,NY,ML,MR,MT,MB,RES,
     2              CONRES,RESFLAG)
                 ENDIF
                 IF(TPMAX.LT.50.) THEN
                    CALL PRINT(TIME,IX1,IX2,IY1,IY2,T)
                    STOP
                 ENDIF
              ELSE
                 AMP=CUR(NTCUR,TIME,RE)
              ENDIF
C
C             ELEN IS THE ELECTRICAL ENERGY RELEASED PER UNIT
C             AREA.
C
              ELEN=(RE*AMP**2)/(121.*RLEN)
C
              DO 160 I=ML,MR
                 DO 161 J=MT,MB
                    QEL(I,J)=ELEN
 161             CONTINUE
 160          CONTINUE
C
C*************************************************************
C                                                            *
```

```
C      FIRST SWEEP, IMPLICIT IN Y AND EXPLICIT IN X                    *
C                                                                      *
C      GET THE SPECIFIC HEAT AND THERMAL CONDUCTIVITY                  *
C      MATRICES.                                                       *
C                                                                      *
C**********************************************************************
           CALL PROP(TIN,NX,NY,ML,MR,MT,MB,CONRES,CON,
     2          CVCP,T)
C
           DO  30 I=1,NX
              DO 31 J=1,NY
                 CALL ASMB(CON,T,I,J,DT,CVCP,CVRHO,A,B,
     2                CC,D,QREL,QEL,1,NX,NY)
  31          CONTINUE
              CALL TRIDAG(1,NY,A,B,CC,D,TT)
              DO 30 J=1,NY
                 TSTAR(I,J)=TT(J)
  30       CONTINUE
C
C**********************************************************************
C                                                                      *
C      SECOND SWEEP, IMPLICIT IN X AND EXPLICIT IN Y                   *
C                                                                      *
C      GET THE SPECIFIC HEAT AND THE THERMAL                           *
C      CONDUCTIVITY MATRICES.                                          *
C                                                                      *
C**********************************************************************
C
           CALL PROP(TIN,NX,NY,ML,MR,MT,MB,CONRES,CON,
     2          CVCP,TSTAR)
C
           DO 32 J=1,NY
              DO 33 I=1,NX
                 CALL ASMB(CON,TSTAR,I,J,DT,CVCP,CVRHO,A,B,
     2                CC,D,QREL,QEL,2,NX,NY)
  33          CONTINUE
              CALL TRIDAG(1,NX,A,B,CC,D,TT)
              DO 32 I=1,NX
                 T(I,J)=TT(I)
  32       CONTINUE
C
C**********************************************************************
C                                                                      *
C          CALCULATE THE TIME                                          *
C                                                                      *
C**********************************************************************
C
           TIME=TIME+2.*DT
```

```
          WRITE(1,*)'TIME=',TIME
C
C
          TPMAX=0.0
          DO 124 I=1,MR
             DO 122 J=1,NY
                IF((I.GE.ML).AND.(J.GE.MT).AND.(J.LE.MB)) THEN
                   GOTO 122
                ELSEIF(T(I,J).GE.TPMAX) THEN
                   TPMAX=T(I,J)
                ELSE
                   GOTO 122
                ENDIF
 122         CONTINUE
 124      CONTINUE
          WRITE(1,*)'TPMAX=',TPMAX
C
C*************************************************************
C                                                            *
C         WRITE OUT THE RESULTS EVERY SET OF ITERATIONS.     *
C                                                            *
C*************************************************************
C
          IF(TPMAX.GE.TIG) THEN
             CALL PRINT(TIME,IX1,IX2,IY1,IY2,T)
             GOTO 67
          ELSE
             GOTO 75
          ENDIF
C
 75       YY=FLOAT(ITER)/FLOAT(MITER)-ITER/MITER
          IF(YY.EQ.0.0) THEN
             CALL PRINT(TIME,IX1,IX2,IY1,IY2,T)
          ELSE
             GOTO 67
          ENDIF
C
C*************************************************************
C                                                            *
C      GET OUT OF LOOP ALL AVAILABLE THERMOCHEMICAL          *
C      ENERGY FOR A CERTAIN ZPP NODE HAS REACTED.            *
C                                                            *
C*************************************************************
C
 67       IF(TPMAX.GT.25000) THEN
             WRITE(1,82)
             WRITE(6,82)
 82          FORMAT(2X,'ALL AVAILABLE CHEMICAL ENERGY HAS
```

C-2

```
     2          BEEN CONSUMED')
                GOTO 88
            ELSE
                GOTO 22
            ENDIF
C
C
 88     STOP
        END
C
C
C
C
C
C**************************************************************
C                                                             *
       FUNCTION GEN(T,AA,EA)                                  *
C                                                             *
C      THIS FUNCTION RETURNS THE VALUE OF THE ENERGY          *
C      RELEASED DUE TO THE CHEMICAL REACTION. (W/M3).         *
C      PARAMETERS ARE ACTIVATION ENERGY (EA), PRE-            *
C      EXPONENTIAL FACTOR OF THE HEAT RELEASE TERM (AA),      *
C      AND ABSOLUTE TEMPERATURE T.                            *

C                                                             *
       GEN=AA*EXP(-EA/T)                                      *
       RETURN                                                 *
       END                                                    *
C                                                             *
C**************************************************************
C
C
C
C
C
C**************************************************************
C                                                             *
       FUNCTION CUR(N,T,R)                                    *
C                                                             *
C      THIS FUNCTION RETURNS THE VALUE OF THE CURRENT AS      *
C      A FUNCTION OF TIME.  THE THREE FIRING MODES ARE        *
C      CONSTANT CURRENT FOR N=1, PIC FOR N=2, SFU FOR N=3.    *
C       R IS THE RESISTANCE OF THE WIRE, AND T IS THE TIME.   *
C                                                             *
       GOTO (10,20,30),N                                      *
C                                                             *
C      THE CONSTANT CURRENT IS EQUAL TO 5 AMPS AT ALL         *
C      TIMES.                                                 *
```

```
C                                                                      *
 10     CUR=5.0                                                        *
        RETURN                                                         *
C                                                                      *
C       THE PIC FIRING MODE IS A CAPACITOR DISCHARGE.                  *
C       C=680 MICROFARADS, V=38 VOLTS.                                 *
C                                                                      *
 20     V2=38.0                                                        *
        C2=680.0E-06                                                   *
C                                                                      *
        CUR=(V2/R)*EXP(-T/(R*C2))                                      *
C                                                                      *
        RETURN                                                         *
C                                                                      *
C       THE SFU FIRING MODE IS A CAPACITOR DISCHARGE.                  *
C       C=1000 MICROFARAD, V=20 VOLTS.                                 *
C                                                                      *
 30     V3=20.0                                                        *
        C3=1000.0E-06                                                  *
C                                                                      *
        CUR=(V3/R)*EXP(-T/(R*C3))                                      *
C                                                                      *
        RETURN                                                         *
        END                                                            *
C                                                                      *
C**********************************************************************
C
C
C
C
C
C**********************************************************************
C                                                                      *
        FUNCTION SSK(T)                                                *
C                                                                      *
C       THIS FUNCTION RETURNS THE THERMAL CONDUCTIVITY OF              *
C       STAINLESS STEEL AS A FUNCTION OF TEMPERATURE.                  *
C                                                                      *
        IF((T.GT.9.9).AND.(T.LT.1660.0)) THEN                          *
        SSK=3.952E-14*T**5-1.816E-10*T**4+3.108E-07*T**3-              *
     2      2.432E-04*T**2+0.0991*T+0.8785                             *
        ELSEIF(T.GE.1660.0) THEN                                       *
        SSK=34.7                                                       *
        ENDIF                                                          *
        RETURN                                                         *
        END                                                            *
C                                                                      *
C**********************************************************************
```

```
C
C
C
C
C
C**************************************************************
C                                                             *
      FUNCTION SSCP(T)                                        *
C                                                             *
C     THIS FUNCTION RETURNS THE SPECIFIC HEAT OF SS-304       *
C     AS A FUNCTION OF TEMPERATURE.                           *
C                                                             *
      IF(T.LT.9.0) THEN                                       *
         WRITE(1,10)T                                         *
10       FORMAT(2X,'STABILITY ERROR IN THE SS REGION.        *
     2        T=',F10.3)                                      *
         STOP                                                 *
      ELSEIF((T.GT.9.9).AND.(T.LE.1070.0)) THEN               *
         SSCP=-2.064E-12*T**5+5.319E-09*T**4-4.2E-06*T**3     *
     2        +3.962E-04*T**2+0.9225*T+220.2225               *
      ELSEIF(T.GT.1070.0) THEN                                *
         SSCP=595.0                                           *
      ENDIF                                                   *
      RETURN                                                  *
      END                                                     *
C                                                             *
C**************************************************************
C
C
C
C
C
C
C**************************************************************
C                                                             *
      FUNCTION ZPPK(T)                                        *
C                                                             *
C     THIS FUNCTION RETURNS THE THERMAL CONDUCTIVITY OF       *
C     ZPP AS A FUNCTION OF TEMPERATURE.  Z IS THE THERMAL     *
C     CONDUCTIVITY OF ZIRCONIUM, AND PPK IS THAT OF           *
C     POTASSIUM PERCHLORATE.  THE STOICHIOMETRIC              *
C     COMPOSITION IS USED TO DETERMINE THE CONDUCTIVITY       *
C     OF THE MIXTURE.                                         *
C                                                             *
      IF((T.GE.9.9).AND.(T.LE.40.0)) THEN                     *
         Z=-0.1117*T**2+4.25*T+67.6667                        *
      ELSEIF((T.GT.40.0).AND.(T.LE.300.0)) THEN               *
         Z=286.8429*T**(-0.4552)                              *
      ELSEIF((T.GT.300.0).AND.(T.LE.1500.0)) THEN             *
```

```
          Z=-1.219E-08*T**3+4.389E-05*T**2-0.0401*T+31.6051          *
        ELSEIF(T.GT.1500.0) THEN                                     *
          Z=25.0                                                     *
        ENDIF                                                        *
C                                                                    *
        PPK=1.0                                                      *
        ZPPK=0.568*Z+0.432*PPK                                       *
        RETURN                                                       *
        END                                                          *
C                                                                    *
C*********************************************************************
C
C
C
C
C
C*********************************************************************
C                                                                    *
        FUNCTION ZPPCP(T)                                            *
C                                                                    *
C       THIS FUNCTION RETURNS THE SPECIFIC HEAT OF ZPP AS A          *
C       FUNCTION OF TEMPERATURE.  ZCP IS THE SPECIFIC HEAT           *
C        OF THE ZIRCONIUM, WHILE PPCP IS THAT OF THE                 *
C       POTASSIUM PERCHLORATE.  AGAIN THE STOICHIOMETRIC             *
C       COMPOSITION IS USED TO DETERMINE THE SPECIFIC HEAT           *
C       OF THE MIX.                                                  *
C                                                                    *
        IF(T.LT.9.0) THEN                                            *
          WRITE(1,88)T                                               *
 88       FORMAT(2X,'STABILITY ERROR IN ZPP REGION. T=',F10.3)*
          STOP                                                       *
        ELSEIF((T.GE.9.9).AND.(T.LT.300.0)) THEN                     *
          ZCP=-1.093E-09*T**5+8.767E-07*T**4-2.360E-04*T**3+         *
     2       0.0187*T**2+1.7661*T-0.8842                             *
        ELSEIF((T.GE.300.0).AND.(T.LE.1140.0)) THEN                  *
          ZCP=3.091E-07*T**3-6.207E-04*T**2+0.489*T+186.5073         *
        ELSEIF(T.GT.1140.) THEN                                      *
          ZCP=395.0                                                  *
        ENDIF                                                        *
C                                                                    *
        IF((T.GE.9.9).AND.(T.LE.260.0)) THEN                         *
          PPCP=-5.066E-08*T**4+9.424E-05*T**3-0.0435*T**2            *
     2       +8.9371*T-77.1619                                       *
        ELSEIF((T.GT.260.0).AND.(T.LE.1500.0)) THEN                  *
          PPCP=1.335E-12*T**5-6.802E-09*T**4+1.362E-05*T**3          *
     2       -0.0136*T**2+7.3602*T-480.5821                          *
        ELSEIF(T.GT.1500.0) THEN                                     *
          PPCP=1600.0                                                *
```

```
      ENDIF                                                             *
C                                                                       *
      ZPPCP=0.568*ZCP+0.432*PPCP                                        *
C                                                                       *
      RETURN                                                            *
      END                                                               *
C                                                                       *
C***********************************************************************
C
C
C
C***********************************************************************
C                                                                       *
      FUNCTION ALK(T,KAL)                                               *
C                                                                       *
C     THIS FUNCTION RETURNS THE THERMAL CONDUCTIVITY OF                 *
C     ALUMINA AS A FUNCTION OF TEMPERATURE.  KAL IS THE                 *
C     INDEX FOR THE KIND OF ALUMINA.  KAL=1 IS FOR 98%                  *
C     DENSE ALUMINA, WHILE KAL=2 IS FOR 100% DENSE                      *
C     ALUMINA.                                                          *
C                                                                       *
      GOTO (10,20),KAL                                                  *
C                                                                       *
C     THE FOLLOWING ARE FOR 98 % DENSE ALUMINA.                         *
C                                                                       *
 10   IF((T.GE.9.9).AND.(T.LT.50.0)) THEN                               *
        ALK=0.049*T**2+1.176*T-9.94                                     *
      ELSEIF((T.GE.50.0).AND.(T.LT.150.0)) THEN                         *
        ALK=-0.0038*T**2-0.1932*T+190.4                                 *
      ELSEIF((T.GE.150.0).AND.(T.LE.1200.0)) THEN                       *
        ALK=3.476E+04*T**(-1.2098)                                      *
      ELSEIF(T.GT.1200.0) THEN                                          *
        ALK=6.50                                                        *
      ENDIF                                                             *
      RETURN                                                            *
C                                                                       *
C                                                                       *
C     THE FOLLOWING ARE FOR 100% DENSE ALUMINA.                         *
C                                                                       *
 20   IF((T.GE.9.9).AND.(T.LE.60.0)) THEN                               *
        ALK=4.875E-05*T**5+6.064E-04*T**4-0.3147*T**3-                  *
     2      17.647*T**2+1675.5212*T-1.169E+04                           *
      ELSEIF((T.GT.60.0).AND.(T.LE.150.0)) THEN                         *
        ALK=1.838E+09*T**(-3.2605)                                      *
      ELSEIF((T.GT.150.0).AND.(T.LE.1500.0)) THEN                       *
        ALK=1.594E+05*T**(-1.4306)                                      *
      ELSEIF(T.GT.1500.0) THEN                                          *
        ALK=5.0                                                         *
```

```
      ENDIF                                                             *
      RETURN                                                            *
      END                                                               *
C                                                                       *
C***********************************************************************
C
C
C
C
C
C***********************************************************************
C                                                                       *
      FUNCTION ALCP(T,I,J,TIN)                                          *
C                                                                       *
C     THIS FUNCTION RETURNS THE SPECIFIC HEAT OF ALUMINA                *
C     AS A FUNCTION OF TEMPERATURE.                                     *
C                                                                       *
      IF(T.LT.TIN-1.0) THEN                                             *
         WRITE(1,88)T                                                   *
 88      FORMAT(2X,'STABILITY ERROR IN THE ALUMINA                      *
     2   REGION. T=',F010.5)                                            *
            WRITE(1,89)I,J                                              *
 89         FORMAT(2X,'ERROR OCCURED IN I=',I3,' AND J=',I3)            *
         STOP                                                           *
      ELSEIF((T.GE.9.9).AND.(T.LT.50.0)) THEN                           *
         ALCP=15.64                                                     *
      ELSEIF((T.GE.50.0).AND.(T.LT.200.0)) THEN                         *
         ALCP=4.643E-09*T**5-2.727E-06*T**4+5.047E-04*T**3              *
     2      -0.0177*T**2+0.249*T                                        *
      ELSEIF((T.GE.200.0).AND.(T.LE.1200)) THEN                         *
         ALCP=2.927E-12*T**5-1.259E-08*T**4+2.168E-05*T**3              *
     2      -0.019*T**2+8.9272*T-676.7606                               *
      ELSEIF(T.GT.1200.0) THEN                                          *
         ALCP=1250.0                                                    *
      ENDIF                                                             *
      RETURN                                                            *
      END                                                               *
C                                                                       *
C***********************************************************************
C
C
C
C
C
C***********************************************************************
C                                                                       *
      SUBROUTINE TCON(NX,NY,C,CON,CONRES,I,J)                           *
      DIMENSION C(5),CON(81,71,4),CONRES(81,71,4)                       *
```

```
      COMMON X(81),Y(71),AREA(81,71),XM(81),YM(71)                    *
C                                                                     *
C     THIS SUBROUTINE EVALUATES THE EQUIVALENT THERMAL                *
C     CONDICTIVITY OF TWO ADJACENT NODES INCLUDING THE                *
C     THERMALCONTACT RESISTANCE BETWEEN THOSE NODES.                  *
C     THE VALUES ARE STORED IN ARRAY CON.                             *
C                                                                     *
      DX1=X(I)-X(I-1)                                                 *
      DX2=X(I+1)-X(I)                                                 *
      DY1=Y(J)-Y(J-1)                                                 *
      DY2=Y(J+1)-Y(J)                                                 *
C                                                                     *
      IF(I.EQ.1) THEN                                                 *
         CON(I,J,1)=0.0                                               *
      ELSE                                                            *
         CON(I,J,1)=(2.*C(1)*C(5)*DX1)/(2.*C(1)*C(5)*CONRES(I,J,1)    *
     2        +DX1*(C(1)+C(5)))                                        *
      ENDIF                                                           *
C                                                                     *
      IF(J.EQ.1) THEN                                                 *
         CON(I,J,2)=0.0                                               *
      ELSE                                                            *
         CON(I,J,2)=(2.*C(2)*C(5)*DY1)/(2.*C(2)*C(5)*CONRES(I,J,2)    *
     2        +DY1*(C(2)+C(5)))                                        *
     ENDIF                                                            *
C                                                                     *
      IF(I.EQ.NX) THEN                                                *
         CON(I,J,3)=0.0                                               *
      ELSE                                                            *
         CON(I,J,3)=(2.*C(3)*C(5)*DX2)/(2.*C(3)*C(5)*CONRES(I,J,3)    *
     2        +DX2*(C(3)+C(5)))                                        *
      ENDIF                                                           *
C                                                                     *
      IF(J.EQ.NY) THEN                                                *
         CON(I,J,4)=0.0                                               *
      ELSE                                                            *
         CON(I,J,4)=(2.*C(4)*C(5)*DY2)/(2.*C(4)*C(5)*CONRES(I,J,4)    *
     2        +DY2*(C(4)+C(5)))                                        *
      ENDIF                                                           *
      RETURN                                                          *
      END                                                             *
C                                                                     *
C*********************************************************************
C
C
C
C
C
```

```
C*************************************************************
C                                                             *
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)                       *
C                                                             *
C     THIS SUBROUTINE SOLVES A SYSTEM OF LINEAR               *
C     SIMULTANEOUS EQUATIONS HAVING A TRIDIAGONAL             *
C     COEFFICIENT MATRIX.                                     *
C                                                             *
      DIMENSION A(1),B(1),C(1),D(1),V(1),BETA(81),GAMMA(81)   *
C                                                             *
C***** COMPUTE INTERMEDIATE ARRAYS BETA AND GAMMA ***** *
C                                                             *
      BETA(IF)=B(IF)                                          *
      GAMMA(IF)=D(IF)/BETA(IF)                                *
      IFP1=IF+1                                               *
      DO 1 I=IFP1,L                                           *
         BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)                   *
         GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)              *
 1    CONTINUE                                                *
C                                                             *
C****** COMPUTE THE FINAL SOLUTION VECTOR ******             *
C                                                             *
      V(L)=GAMMA(L)                                           *
      LAST=L-IF                                               *
      DO 2 K=1,LAST                                           *
         I=L-K                                                *
         V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)                    *
 2    CONTINUE                                                *
      RETURN                                                  *
      END                                                     *
C                                                             *
C*************************************************************
C
C
C
C
C
C
C*************************************************************
C                                                             *
      SUBROUTINE ASMB(CON,T,I,J,DT,CP,RHO,A,B,CC,D,           *
         QREL,QEL,L,NX,NY)                                    *
      DIMENSION A(1),B(1),CC(1),D(1),T(81,71),CON(81,71,4),   *
     2    CP(81,71),RHO(81,71),QREL(81,71),QEL(81,71)         *
      COMMON X(81),Y(71),AREA(81,71),XM(81),YM(71)            *
C                                                             *
C     THIS SUBROUTINE ASSEMBLES THE TRIDIAGONAL MATRIX        *
C     THAT IS USED TO FIND THE SOLUTION VECTOR                *
C                                                             *
```

```
      RLAM=(AREA(I,J)*RHO(I,J)*CP(I,J))/DT                              *
      RLEN=3.0E-03                                                      *
      IF(I.EQ.1) THEN                                                   *
         RJAC1=0.0                                                      *
      ELSE                                                              *
         RJAC1=(YM(J+1)-YM(J))/(X(I)-X(I-1))                            *
      ENDIF                                                             *
C                                                                       *
      IF(J.EQ.1) THEN                                                   *
         RJAC2=0.0                                                      *
      ELSE                                                              *
         RJAC2=(XM(I+1)-XM(I))/(Y(J)-Y(J-1))                            *
      ENDIF                                                             *
C                                                                       *
      IF(I.EQ.NX) THEN                                                  *
         RJAC3=0.0                                                      *
      ELSE                                                              *
         RJAC3=(YM(J+1)-YM(J))/(X(I+1)-X(I))                            *
      ENDIF                                                             *
C                                                                       *
      IF(J.EQ.NY) THEN                                                  *
         RJAC4=0.0                                                      *
      ELSE                                                              *
         RJAC4=(XM(I+1)-XM(I))/(Y(J+1)-Y(J))                            *
      ENDIF                                                             *
C                                                                       *
      GOTO (20,10),L                                                    *
C                                                                       *
C     IMPLICIT IN X                                                     *
C                                                                       *
 10   IF(J.EQ.1) THEN                                                   *
         D(I)=RJAC4*CON(I,J,4)*T(I,J+1) + (RLAM-RJAC4*                  *
     2        CON(I,J,4)) *T(I,J)+QREL(I,J)+QEL(I,J)                    *
         IF(I.EQ.1) THEN                                                *
            A(I)=0.0                                                    *
            B(I)=RLAM+RJAC3*CON(I,J,3)                                  *
            CC(I)=-RJAC3*CON(I,J,3)                                     *
         ELSEIF(I.EQ.NX) THEN                                           *
            A(I)=-RJAC1*CON(I,J,1)                                      *
            B(I)=RLAM+RJAC1*CON(I,J,1)                                  *
            CC(I)=0.0                                                   *
         ELSE                                                           *
            A(I)=-RJAC1*CON(I,J,1)                                      *
            B(I)=RLAM+RJAC1*CON(I,J,1)+RJAC3*CON(I,J,3)                 *
            CC(I)=-RJAC3*CON(I,J,3)                                     *
         ENDIF                                                          *
C                                                                       *
      ELSEIF(J.EQ.NY) THE                                               *
```

```
          D(I)=RJAC2*CON(I,J,2)*T(I,J-1)+(RLAM-RJAC2*                    *
     2        CON(I,J,2))*T(I,J)+QREL(I,J)+QEL(I,J)                      *
          IF(I.EQ.1) THE                                                 *
             A(I)=0.0                                                    *
             B(I)=RLAM+RJAC3*CON(I,J,3)                                  *
             CC(I)=-RJAC3*CON(I,J,3)                                     *
          ELSEIF(I.EQ.NX) THEN                                           *
             A(I)=-RJAC1*CON(I,J,1)                                      *
             B(I)=RLAM+RJAC1*CON(I,J,1)                                  *
             CC(I)=0.0                                                   *
          ELSE                                                           *
             A(I)=-RJAC1*CON(I,J,1)                                      *
             B(I)=RLAM+RJAC1*CON(I,J,1)+RJAC3*CON(I,J,3)                 *
             CC(I)=-RJAC3*CON(I,J,3)                                     *
          ENDIF                                                          *
       ELSE                                                              *
          D(I)=RJAC2*CON(I,J,2)*T(I,J-1)+(RLAM-RJAC2*CON(I,J,2)-         *
     2        RJAC4*CON(I,J,4))*T(I,J)+RJAC4*CON(I,J,4)*T(I,J+1)         *
     3        +QREL(I,J)+QEL(I,J)                                        *
          IF(I.EQ.1) THEN                                                *
             A(I)=0.0                                                    *
             B(I)=RLAM+RJAC3*CON(I,J,3)                                  *
             CC(I)=-RJAC3*CON(I,J,3)                                     *
          ELSEIF(I.EQ.NX) THEN                                           *
             A(I)=-RJAC1*CON(I,J,1)                                      *
             B(I)=RLAM+RJAC1*CON(I,J,1)                                  *
             CC(I)=0.0                                                   *
          ELSE                                                           *
             A(I)=-RJAC1*CON(I,J,1)                                      *
             B(I)=RLAM+RJAC1*CON(I,J,1)+RJAC3*CON(I,J,3)                 *
             CC(I)=-RJAC3*CON(I,J,3)                                     *
          ENDIF                                                          *
       ENDIF                                                             *
       RETURN                                                            *
C                                                                        *
C                                                                        *
C      IMPLICIT IN Y.                                                    *
C                                                                        *
 20    IF(I.EQ.1) THEN                                                   *
          D(J)=RJAC3*CON(I,J,3)*T(I+1,J)+(RLAM-RJAC3*                    *
     2        CON(I,J,3))*T(I,J)+QREL(I,J)+QEL(I,J)                      *
          IF(J.EQ.1) THEN                                                *
             A(J)=0.0                                                    *
             B(J)=RLAM+RJAC4*CON(I,J,4)                                  *
             CC(J)=-RJAC4*CON(I,J,4)                                     *
          ELSEIF(J.EQ.NY) THEN                                           *
             A(J)=-RJAC2*CON(I,J,2)                                      *
             B(J)=RLAM+RJAC2*CON(I,J,2)                                  *
```

```
            CC(J)=0.0                                                    *
         ELSE                                                            *
            A(J)=-RJAC2*CON(I,J,2)                                       *
            B(J)=RLAM+RJAC2*CON(I,J,2)+RJAC4*CON(I,J,4)                  *
            CC(J)=-RJAC4*CON(I,J,4)                                      *
         ENDIF                                                           *
      ELSEIF(I.EQ.NX) THEN                                               *
         D(J)=RJAC1*CON(I,J,1)*T(I-1,J)+(RLAM-RJAC1*                     *
     2      CON(I,J,1))*T(I,J)+QREL(I,J)+QEL(I,J)                       *
         IF(J.EQ.1) THEN                                                 *
            A(J)=0.0                                                     *
            B(J)=RLAM+RJAC4*CON(I,J,4)                                   *
            CC(J)=-RJAC4*CON(I,J,4)                                      *
         ELSEIF(J.EQ.NY) THEN                                            *
            A(J)=-RJAC2*CON(I,J,2)                                       *
            B(J)=RLAM+RJAC2*CON(I,J,2)                                   *
            CC(J)=0.0                                                    *
         ELSE                                                            *
            A(J)=-RJAC2*CON(I,J,2)                                       *
            B(J)=RLAM+RJAC2*CON(I,J,2)+RJAC4*CON(I,J,4)                  *
            CC(J)=-RJAC4*CON(I,J,4)                                      *
         ENDIF                                                           *
      ELSE                                                               *
         D(J)=RJAC1*CON(I,J,1)*T(I-1,J)+(RLAM-RJAC1*CON(I,J,1)-          *
     2      RJAC3*CON(I,J,3))*T(I,J)+RJAC3*CON(I,J,3)*T(I+1,J)          *
     3      +QREL(I,J)+QEL(I,J)                                         *
         IF(J.EQ.1) THEN                                                 *
            A(J)=0.0                                                     *
            B(J)=RLAM+RJAC4*CON(I,J,4)                                   *
            CC(J)=-RJAC4*CON(I,J,4)                                      *
         ELSEIF(J.EQ.NY) THEN                                            *
            A(J)=-RJAC2*CON(I,J,2)                                       *
            B(J)=RLAM+RJAC2*CON(I,J,2)                                   *
            CC(J)=0.0                                                    *
         ELSE                                                            *
            A(J)=-RJAC2*CON(I,J,2)                                       *
            B(J)=RLAM+RJAC2*CON(I,J,2)+RJAC4*CON(I,J,4)                  *
            CC(J)=-RJAC4*CON(I,J,4)                                      *
         ENDIF                                                           *
      ENDIF                                                              *
C                                                                        *
      RETURN                                                             *
      END                                                                *
C                                                                        *
C************************************************************************
C
C
C
```

```
C
C
C*****************************************************************
C                                                                *
      SUBROUTINE MESH(IXCOUNT,IYCOUNT,LXWIRE,                    *
     2     LYWIRE,MM)                                            *
      COMMON X(81),Y(71),AREA(81,71),XM(81),YM(71)               *
C                                                                *
C     THIS SUBROUTINE GENERATES THE COORDINATES OF THE           *
C     MESH THAT IS USED IN THE CODE.                             *
C                                                                *
      XCUR=0.0                                                   *
      YCUR=0.0                                                   *
      IXCOUNT=0                                                  *
      IYCOUNT=0                                                  *
C                                                                *
      XC1=0.0007804                                              *
      XC2=0.0009004                                              *
      XL=0.0035558                                               *
C                                                                *
      YC1=0.001936                                               *
      YC2=0.002064                                               *
      YL=0.0040                                                  *
C                                                                *
      DX1=1.951E-04                                              *
      DX2=4.000E-06                                              *
      DX3=2.414E-04                                              *
C                                                                *
      DY1=1.936E-04                                              *
      DY2=4.000E-06                                              *
      DY3=1.936E-04                                              *
C                                                                *
      DO 10 I=1,81                                               *
         X(I)=0.0                                                *
         Y(I)=0.0                                                *
         DO 10 J=1,71                                            *
            AREA(I,J)=0.0                                        *
10    CONTINUE                                                   *
C                                                                *
20    IXCOUNT=IXCOUNT+1                                          *
      IF(IXCOUNT.EQ.1) THEN                                      *
         DX=0.0                                                  *
      ELSEIF((XCUR.GE.0.0).AND.(XCUR.LT.XC1)) THEN               *
         DX=DX1                                                  *
      ELSEIF((XCUR.GE.XC1).AND.(XCUR.LE.XC2)) THEN               *
         DX=DX2                                                  *
      ELSEIF((XCUR.GT.XC2).AND.(XCUR.LT.XL)) THEN                *
         DX=DX3                                                  *
```

```
      ELSE                                                              *
         GOTO 30                                                        *
      ENDIF                                                             *
      XCUR=XCUR+DX                                                      *
      X(IXCOUNT)=XCUR                                                   *
      IF((XCUR.GT.0.008563E-04).AND.(XCUR.LT.8.565E-04)) THEN           *
         LXWIRE=IXCOUNT                                                 *
         GOTO 20                                                        *
      ELSE                                                              *
         GOTO 20                                                        *
      ENDIF                                                             *
C                                                                       *
C                                                                       *
 30   IYCOUNT=IYCOUNT+1                                                 *
      IF(IYCOUNT.EQ.1) THEN                                             *
         DY=0.0                                                         *
      ELSEIF((YCUR.GE.0.0).AND.(YCUR.LT.YC1)) THEN                      *
         DY=DY1                                                         *
      ELSEIF((YCUR.GE.YC1).AND.(YCUR.LT.YC2)) THEN                      *
         DY=DY2                                                         *
      ELSEIF ((YCUR.GT.YC2).AND.(YCUR.LT.YL)) THEN                      *
         DY=DY3                                                         *
      ELSE                                                              *
         GOTO 31                                                        *
      ENDIF                                                             *
      YCUR=YCUR+DY                                                      *
      Y(IYCOUNT)=YCUR                                                   *
      IF((YCUR.GT.1.999E-03).AND.(YCUR.LT.2.001E-03)) THEN              *
         LYWIRE=IYCOUNT                                                 *
         GOTO 30                                                        *
      ELSE                                                              *
         GOTO 30                                                        *
      ENDIF                                                             *
C                                                                       *
C                                                                       *
 31   DO 40 I=1,IXCOUNT                                                 *
         IF(I.EQ.1) THEN                                                *
            XM(I)=0.0                                                   *
         ELSEIF(I.EQ.IXCOUNT) THEN                                      *
            XM(I)=XL                                                    *
         ELSE                                                           *
            XM(I)=(X(I)+X(I-1))/2.0                                     *
         ENDIF                                                          *
 40   CONTINUE                                                          *
C                                                                       *
      DO 41 J=1,IYCOUNT                                                 *
         IF(J.EQ.1) THEN                                                *
            YM(J)=0.0                                                   *
```

```
          ELSEIF(J.EQ.IYCOUNT) THEN                                    *
              YM(J)=YL                                                 *
          ELSE                                                         *
              YM(J)=(Y(J)+Y(J-1))/2.0                                  *
          ENDIF                                                        *
 41     CONTINUE                                                       *
C                                                                       *
C                                                                       *
       DO 50 I=1,IXCOUNT-1                                             *
          DO 60 J=1,IYCOUNT-1                                          *
              AREA(I,J)=(XM(I+1)-XM(I))*(YM(J+1)-YM(J))                *
 60       CONTINUE                                                     *
 50    CONTINUE                                                       *
C                                                                       *
       MM=5                                                            *
       WRITE(6,*)'IXCOUNT=',IXCOUNT                                    *
       WRITE(6,*)'IYCOUNT=',IYCOUNT                                    *
       RETURN                                                          *
       END                                                             *
C                                                                       *
C***********************************************************************
C
C
C
C
C
C***********************************************************************
C                                                                       *
       SUBROUTINE DETDT(DT1,DT2,DT,T,TIG,TFLAG)                        *
C                                                                       *
C      THIS SUBROUTINE DETERMINES THE TIME STEP (DT) TO BE              *
C      USED IN THE CALCULATIONS FOR EACH ITERATION.                     *
C                                                                       *
       IF(T.GT.TIG) THEN                                               *
          DT=DT2                                                       *
          TFLAG=1.0                                                    *
       ELSE                                                            *
          DT=DT1                                                       *
          TFLAG=0.0                                                    *
       ENDIF                                                           *
       RETURN                                                          *
       END                                                             *
C                                                                       *
C***********************************************************************
C
C
C
C
```

```
C
C*****************************************************************
C                                                                *
      SUBROUTINE CONTACT(NX,NY,ML,MR,MT,MB,RES,CONRES,*
         RESFLAG)                                               *
C                                                               *
C     THIS SUBROUTINE EVALUATES THE FOUR CONTACT                *
C     RESISTANCES FOR ALLTHE CONTACT RESISTANCES AND            *
C     STORES THE VALUES IN ARRAY RES.                           *
C                                                               *
      DIMENSION RES(50),CONRES(81,71,4)                         *
C                                                               *
C     INITIALIZE THE CONTACT RESISTANCE MATRIX.                 *
C                                                               *
      KOUNT=0                                                   *
C                                                               *
      DO 40 I=ML,MR                                             *
         KOUNT=KOUNT+1                                          *
         CONRES(I,MT,2)=RES(KOUNT)                              *
         CONRES(I,MT-1,4)=RES(KOUNT)                            *
 40   CONTINUE                                                  *
C                                                               *
      DO 41 J=MT,MB                                             *
         KOUNT=KOUNT+1                                          *
         CONRES(MR,J,3)=RES(KOUNT)                              *
         CONRES(MR+1,J,1)=RES(KOUNT)                            *
 41   CONTINUE                                                  *
C                                                               *
      DO 42 I=ML,MR                                             *
         II=MR+ML-I                                             *
         KOUNT=KOUNT+1                                          *
         CONRES(II,MB,4)=RES(KOUNT)                             *
         CONRES(II,MB+1,2)=RES(KOUNT)                           *
 42   CONTINUE                                                  *
C                                                               *
      DO 43 J=MT,MB                                             *
         JJ=MB+MT-J                                             *
         KOUNT=KOUNT+1                                          *
         CONRES(ML,JJ,1)=RES(KOUNT)                             *
         CONRES(ML-1,JJ,3)=RES(KOUNT)                           *
 43   CONTINUE                                                  *
C                                                               *
      DO 44 J=1,MT-1                                            *
         CONRES(MR,J,3)=RES(45)                                 *
         CONRES(MR+1,J,1)=RES(45)                               *
 44   CONTINUE                                                  *
C                                                               *
      DO 45 J=MB+1,NY                                           *
```

```
          CONRES(MR,J,3)=RES(45)                                        *
          CONRES(MR+1,J,1)=RES(45)                                      *
 45     CONTINUE                                                        *
C                                                                       *
       RESFLAG=1.0                                                      *
C                                                                       *
       RETURN                                                           *
       END                                                              *
C                                                                       *
C***********************************************************************
C
C
C
C
C
C
C***********************************************************************
C                                                                       *
       SUBROUTINE PRINT(TIME,IX1,IX2,IY1,IY2,T)                         *
C                                                                       *
C      THIS SUBROUTINE PRINTS OUT THE TEMPERATURE VALUES                *
C      FOR THE NODES IN CONSIDERATION.                                  *
C                                                                       *
       DIMENSION T(81,71)                                               *
C                                                                       *
       WRITE(9,*)TIME                                                   *
       DO 10 J=IY1,IY2                                                  *
          DO 20 I=IX1,IX2                                               *
             WRITE(9,*)T(I,J)                                           *
 20       CONTINUE                                                      *
 10    CONTINUE                                                        *
       RETURN                                                           *
       END                                                              *
C                                                                       *
C***********************************************************************
C
C
C
C
C
C
C***********************************************************************
C                                                                       *
       SUBROUTINE PROP(TIN,NX,NY,ML,MR,MT,MB,CONRES,CON,*
          CVCP,T)                                                       *
       DIMENSION CON(81,71,4),CVCP(81,71),C(5),T(81,71),                *
     2    CONRES(81,71,4)                                              *
C                                                                       *
C      THIS SUBROUTINE DETERMINES THE THERMAL                           *
C      CONDUCTIVITY AND SPECIFIC HEAT MATRICES FOR THE                  *
```

```
C     MODEL.                                                          *
C                                                                     *
      DO 10 I=1,NX                                                    *
        DO 11 J=1,NY                                                  *
          IF(I.EQ.1) THEN                                             *
            C(1)=0.0                                                  *
          ELSEIF((I.GT.1).AND.(I.LE.ML)) THEN                         *
            C(1)=ZPPK(T(I-1,J))                                       *
          ELSEIF((I.GT.ML).AND.(I.LE.MR+1)) THEN                      *
            IF((J.GE.MT).AND.(J.LE.MB)) THEN                          *
              C(1)=SSK(T(I-1,J))                                      *
            ELSE                                                      *
              C(1)=ZPPK(T(I-1,J))                                     *
            ENDIF                                                     *
          ELSEIF(I.GT.MR+1) THEN                                      *
            C(1)=ALK(T(I-1,J),KAL)                                    *
          ENDIF                                                       *
C                                                                     *
C                                                                     *
          IF(J.EQ.1) THEN                                             *
            C(2)=0.0                                                  *
          ELSEIF((J.GT.1).AND.(J.LE.MT)) THEN                         *
            IF(I.LE.MR) C(2)=ZPPK(T(I,J-1))                           *
            IF(I.GT.MR) C(2)=ALK(T(I,J-1),KAL)                        *
          ELSEIF((J.GT.MT).AND.(J.LE.MB+1)) THEN                      *
            IF(I.LT.ML) C(2)=ZPPK(T(I,J-1))                           *
            IF((I.GE.ML).AND.(I.LE.MR)) C(2)=SSK(T(I,J-1))            *
            IF(I.GT.MR) C(2)=ALK(T(I,J-1),KAL)                        *
          ELSEIF(J.GT.MB+1) THEN                                      *
            IF(I.LE.MR) C(2)=ZPPK(T(I,J-1))                           *
            IF(I.GT.MR) C(2)=ALK(T(I,J-1),KAL)                        *
          ENDIF                                                       *
C                                                                     *
C                                                                     *
          IF((I.GE.1).AND.(I.LE.ML-2)) THEN                           *
            C(3)=ZPPK(T(I+1,J))                                       *
          ELSEIF((I.GT.ML-2).AND.(I.LE.MR-1)) THEN                    *
            IF((J.GE.MT).AND.(J.LE.MB)) THEN                          *
              C(3)=SSK(T(I+1,J))                                      *
            ELSE                                                      *
              C(3)=ZPPK(T(I+1,J))                                     *
            ENDIF                                                     *
          ELSEIF((I.GE.MR).AND.(I.LT.NX)) THEN                        *
            C(3)=ALK(T(I+1,J),KAL)                                    *
          ELSEIF(I.EQ.NX)THEN                                         *
            C(3)=0.0                                                  *
          ENDIF                                                       *
C                                                                     *
```

```
C                                                                       *
            IF((J.GE.1).AND.(J.LE.MT-2)) THEN                           *
               IF(I.LE.MR) C(4)=ZPPK(T(I,J+1))                          *
               IF(I.GT.MR) C(4)=ALK(T(I,J+1),KAL)                       *
            ELSEIF((J.GE.MT-1).AND.(J.LE.MB-1)) THEN                    *
               IF(I.LT.ML) C(4)=ZPPK(T(I,J+1))                          *
               IF((I.GE.ML).AND.(I.LE.MR)) C(4)=SSK(T(I,J+1))           *
               IF(I.GT.MR) C(4)=ALK(T(I,J+1),KAL)                       *
            ELSEIF((J.GE.MB).AND.(J.LT.NY)) THEN                        *
               IF(I.LE.MR) C(4)=ZPPK(T(I,J+1))                          *
               IF(I.GT.MR) C(4)=ALK(T(I,J+1),KAL)                       *
            ELSEIF(J.EQ.NY) THEN                                        *
               C(4)=0.0                                                 *
            ENDIF                                                       *
C                                                                       *
C                                                                       *
            IF(I.LT.ML) THEN                                            *
               C(5)=ZPPK(T(I,J))                                        *
               CVCP(I,J)=ZPPCP(T(I,J))                                  *
            ELSEIF((I.GE.ML).AND.(I.LE.MR)) THEN                        *
               IF((J.GE.MT).AND.(J.LE.MB)) THEN                         *
                  C(5)=SSK(T(I,J))                                      *
                  CVCP(I,J)=SSCP(T(I,J))                                *
               ELSE                                                     *
                  C(5)=ZPPK(T(I,J))                                     *
                  CVCP(I,J)=ZPPCP(T(I,J))                               *
               ENDIF                                                    *
            ELSEIF(I.GT.MR) THEN                                        *
               C(5)=ALK(T(I,J),KAL)                                     *
               CVCP(I,J)=ALCP(T(I,J),I,J,TIN)                           *
            ENDIF                                                       *
C                                                                       *
            WRITE(8,88)I,J,CVCP(I,J),(C(K),K=1,5)                       *
 88         FORMAT(1X,'I=',I2,1X,'J=',I2,1X,'CP=',F7.2,                 *
     2         1X,5(F6.2,1X))                                           *
            CALL TCON(NX,NY,C,CON,CONRES,I,J)                           *
C                                                                       *
 11       CONTINUE                                                     *
 10    CONTINUE                                                        *
       RETURN                                                          *
       END                                                             *
C                                                                       *
C***********************************************************************
```

References

1) "NASA Standard Initiator Failure Analysis Report", *Preliminary*, T. J. Graves (ed), National Aeronautics and Space Administration, Lyndon B. Johnson Center, Houston, Texas, February 1987.

2) J. A. Conkling, *Chemistry of Pyrotechnics*, pp. 49-81, Marcel Dekker, New York, 1985.

3) K. O. Brauer, *Handbook of Pyrotechnics*, pp. 41-51, Chemical Publishing Co., New York, 1974.

4) Ref. 1.

5) W. B. Leslie and R. W. Dietzel, "Study of Raised Bridgewires in a Pyrotechnic Device", Sandia Laboratories, Albuquerque, New Mexico, SAND76-0725, 1976.

6) M. L. Lieberman and F. J. Villa, "Analysis of a Potential Hot Wire Ignition Problem", Sandia Laboratories, Albuquerque, New Mexico, SAND83-1894, 1983.

7) A. B. Donaldson, "Hot Wire Ignition of Pyrotechnic Materials", Sandia Laboratories, Albuquerque, New Mexico, CONF-760711-2, 1976.

8) H. L. Selberg and C. H. Johansson, "Numerical Calculations of the Temperature Rise in Ignition by Hot Wires", Arkiv for Fysik, **13**, pp. 423-427, 1958.

9) J. Belle and M. W. Mallett, "Kinetics of the High Temperature Oxidation of Zirconium", J. Electrochem. Soc., **101**, pp. 339-341, 1954.

10) Ref. 1, Appendix A.

11) S. Kakaç and Y. Yener, *Heat Conduction*, p. 318, Hemisphere, New York, 1985.

12) Ref.1, Appendix N.

13) Ref.1, Appendix Q.

14) Ref.1, Appendix C.

15) B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*, pp. 452-453, Wiley, New York, 1973.

16) K. Van Tassel, NASA, Johnson Space Center, private communication.

17) S. S. Todd, "Heat Capacities of Zirconium, Zirconium Nitride, and Zirconium Tetrachloride", American Chemical Society Journal, **72**, p. 2914, 1950.

18) M. W. Chase, C. A. Davies, J. R. Downey, D. J. Frurip, R. A. McDonald, A. N. Syverud, Journal of Physical and Chemical Reference Data, **14**, Supplement No.1, JANAF Thermochemical Tables, Third Edition, p. 757, 1985.

19) G. T. Furukawa, T. B. Douglas, R. E. McCoskey, D. C. Ginnings, "Thermal Properties of Aluminum Oxide from 0-1200 K", Journal of Research of the National Bureau of Standards, **57**, p. 67-82, 1956.

20) Y. S. Touloukian, "Thermophysical Properties of Matter", **4**, p. 699-708, Plenum Publishing Corporation, 1970.

21) "Heat Exchanger Design Handbook", p. 5.5.6-2, Hemisphere Publishing Corporation, 1983.

22) Y. S. Touloukian, "Thermophysical Properties of Matter", **2**, p. 647, Plenum Publishing Corporation, 1970.

23) K. Wefers, and G. M. Bell, "Oxides and Hydroxides of Aluminum", Technical paper no. 19, Alcoa Research Laboratories, 1972.

24) Y. S. Touloukian, "Thermophysical Properties of Matter", **1**, p. 1175, Plenum Publishing Corporation, 1970.