NASA-CR-190664

Adiabatic Demagnetization Refrigerator for Use in Zero Gravity

SBIR-08.07-8629
release date 2/27/90

p. 143

Michael L. Dirgus

Alabama Cryogenic Engineering, Inc.
P.O. Box 2451
Huntsville, AL 35804

June 16, 1988

Final Report - February 28, 1986 through April 11, 1988

Contract Number NAS5-29418

Prepared for

Goddard Space Flight Center
Greenbelt, MD

(NASA-CR-190664) ADIABATIC DEMAGNETIZATION REFRIGERATOR FOR USE IN ZERO GRAVITY Final Report, 28 Feb. 1986 - 11 Apr. 1988 (Alabama Cryogenic Engineering) 143 p

Unclas

G3/29

ABSTRACT

In this effort, a new design concept for an Adiabatic Demagnetization Refrigerator that is capable of operation in zero gravity has been developed. The design uses a vortex precooler to lower the initial temperature of magnetic salt from the initial space superfluid helium dewar of 1.8 K to 1.1 K. This reduces the required maximum magnetic field from 4 Tesla to 2 Tesla.

The laboratory prototype vortex precooler reached a minimum temperature of 0.78 K, and had a cooling power of 1 mW at 1.1 K. A study was conducted to determine the dependence of vortex cooler performance on system element configuration. A superfluid filled capillary heat switch was used in the design. The laboratory prototype ADR reached a minimum temperature of 0.107 K, and maintained temperatures below 0.125 K for 90 minutes. Demagnetization was carried out from a maximum field of 2T. A soft iron shield was developed that reduced the radial central field to 1 gauss at 0.25 meters.

TABLE OF CONTENTS

I. INTRODUCTION

Cryogenics has always played an important part in the advancement of space science. Applications range from the storage of cryogenic propellants to the development of cryocoolers for space use. Development of sensor technology in recent years has increased the importance of cryogenics in space science considerably. In the temperature range below 0.5 K, the dominant area of interest has been providing cooling for bolometers and other sensors.

A bolometer is essentially a radiation sensitive variable resistor. Response can be achieved over a wide variety of wavelengths, but the main focus of bolometer development work has been to develop detectors in the long wavelength infrared band (IR). The performance of bolometers has been shown to improve as temperature is decreased. The noise equivalent power (NEP) of bolometers should decrease as T^n where T is the temperature and $3/2 < n < 5/2$ (Castles, 1980). Conventional cooling systems, such as evaporative cooling with stored liquid cryogens, is only useful down to approximately 1.5 K. Therefore, there is a great practical need for cryocooler systems that can cool below this temperature and that can operate successfully in zero gravity conditions.

An Adiabatic Demagnetization Refrigerator is an attractive means to achieving cooling in zero gravity down to 0.1 K. Earth based ADR technology is very mature, and achieving such temperatures can be accomplished with a variety of magnetic refrigerant materials. The ADR cycle itself is not fundamentally dependent on gravity, and has a relatively high thermodynamic efficiency.

Problems do exist, however. A superconducting magnet will almost certainly be used to drive the cycle. Weight and spatial dimensions must be minimized as much as possible. Efficiency of the cycle should be maximized, to avoid depleting the necessary stored superfluid helium. Also, the effects of fringing magnetic fields aboard the spacecraft could be quite serious.

This work has been conducted in an attempt to find solutions to some of the aforementioned problems. In the ACE, Inc. ADR, a device called a vortex cooler is used to precool the magnetic salt from 1.8 K below 1.1 K before demagnetization takes place. This precooling results in a substantial reduction in the required magnetic field, which reduces system size, weight, and lowers fringing magnetic field effects. Figure 1 shows that precooling the salt from 2.0 K to 1.1 K reduces the required field for ferric amonium alum by a factor of two. Also, a passive soft iron shield that surrounds a superconducting magnet has been developed that substantially reduces stray magnetic fields. Computer software has been written to serve as a tool for estimating magnetic shielding effects for a given configuration. All of these components have been successfully tested in the laboratory.

In the development of the vortex precooler, a great deal was learned about the dependence of device performance on construction parameters. The vortex cooler, a gravity independent device which has no moving parts and uses superfluid ^{4}He to cool to temperatures as low as 0.7 K, should prove to be a very useful apparatus in other space cryocooler applications.

The research and development program for the shielded, vortex precooled ADR development program will be discussed in detail in the following text.

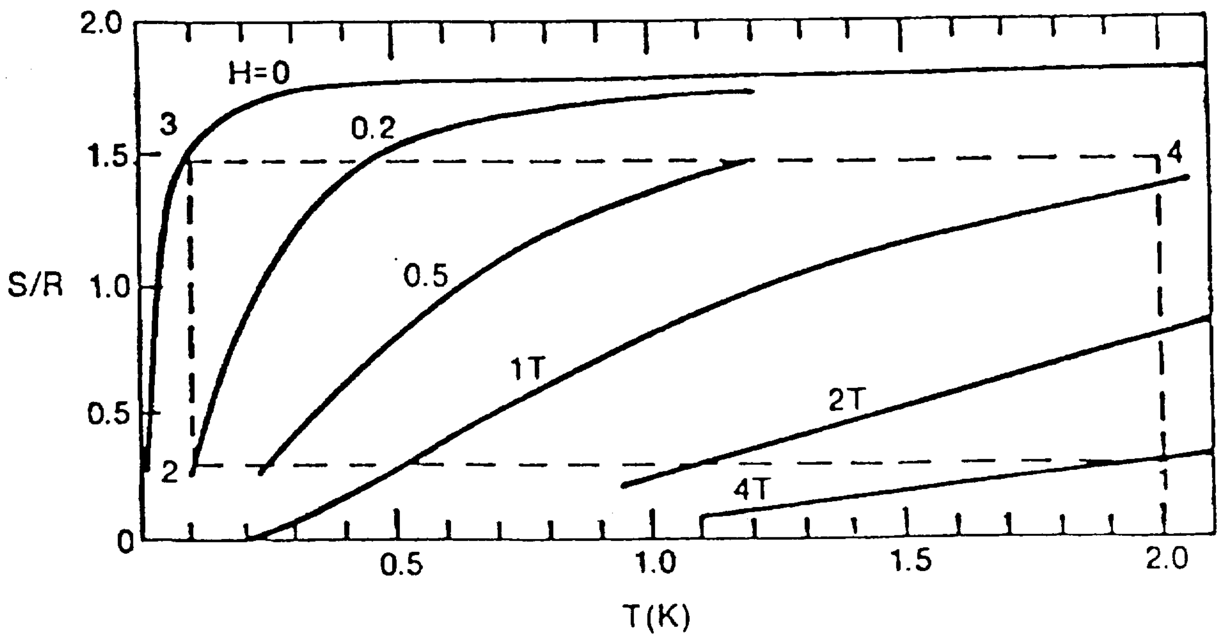

Figure 1 Entropy versus Temperature Chart for Ferric Ammonium Alum. From Castles (1980)

II. BASELINE DESIGN

The purpose of this Phase II research effort has been to build and test a laboratory prototype version of a 0.1 K Adiabatic Demagnetization Refrigerator (ADR) suitable for spaceborne operation. The cryocooler was designed to make use of a vortex cooler precooling device driven by a superfluid fountain pump to lower the temperature of magnetization from the space dewar temperature of 1.8 K down to 1.1 K. The vortex precooler system has no moving parts, is gravity independent, and uses superfluid ^{4}He as a cooling fluid. This precooling reduces the required magnetic field for ADR operation. A superfluid filled capillary is used for the heat switch in the ADR cycle by a factor of two. A soft iron magnetic shield has been developed to reduce fringing fields. The vortex precooler assumes the availability of a 1.8 K heat sink, such as will be available on some helium space dewar configurations.

III. TEST FACILITY AND INSTRUMENTATION

In this section, the test facilities used to develop both the vortex cooler precooling system and the ADR/heat switch assembly will be discussed in detail. A description will be given for the basic cryostat and dewar configuration along with a listing of the electronics and support equipment used in the tests.

ADR Dewar Support System

The support system for the dewar and cryostat will now be described. A unistrut frame is used to support a three inch thick hexagonal aluminum piece, known as the top block. A sketch of the top block is shown in Figure 2. The top block supports the cryostat and the dewar. The dewar is attached via screws to the top block from below; an o-ring seals the dewar and top block together. The cryostat slides into the top block from above, coming to rest on the cryostat top plate, which seals to the top block with a rubber o-ring. This arrangement allows easy removal of the cryostat from the test facility when repairs or changes are needed. Use of o-ring seals makes it possible that the ^{4}He bath space can be pumped on or evacuated and backfilled with gas as required.

The dewar itself is shown in Figure 3. It was manufactured by Cryofab, Inc. and is a model CSM8.5N-5 standard dewar. This dewar has double insulated walls, with space provided for a bath of liquid nitrogen that surrounds the inner dewar wall containing the bath of liquid helium. A system of counterweights was employed to raise and lower the dewar easily.

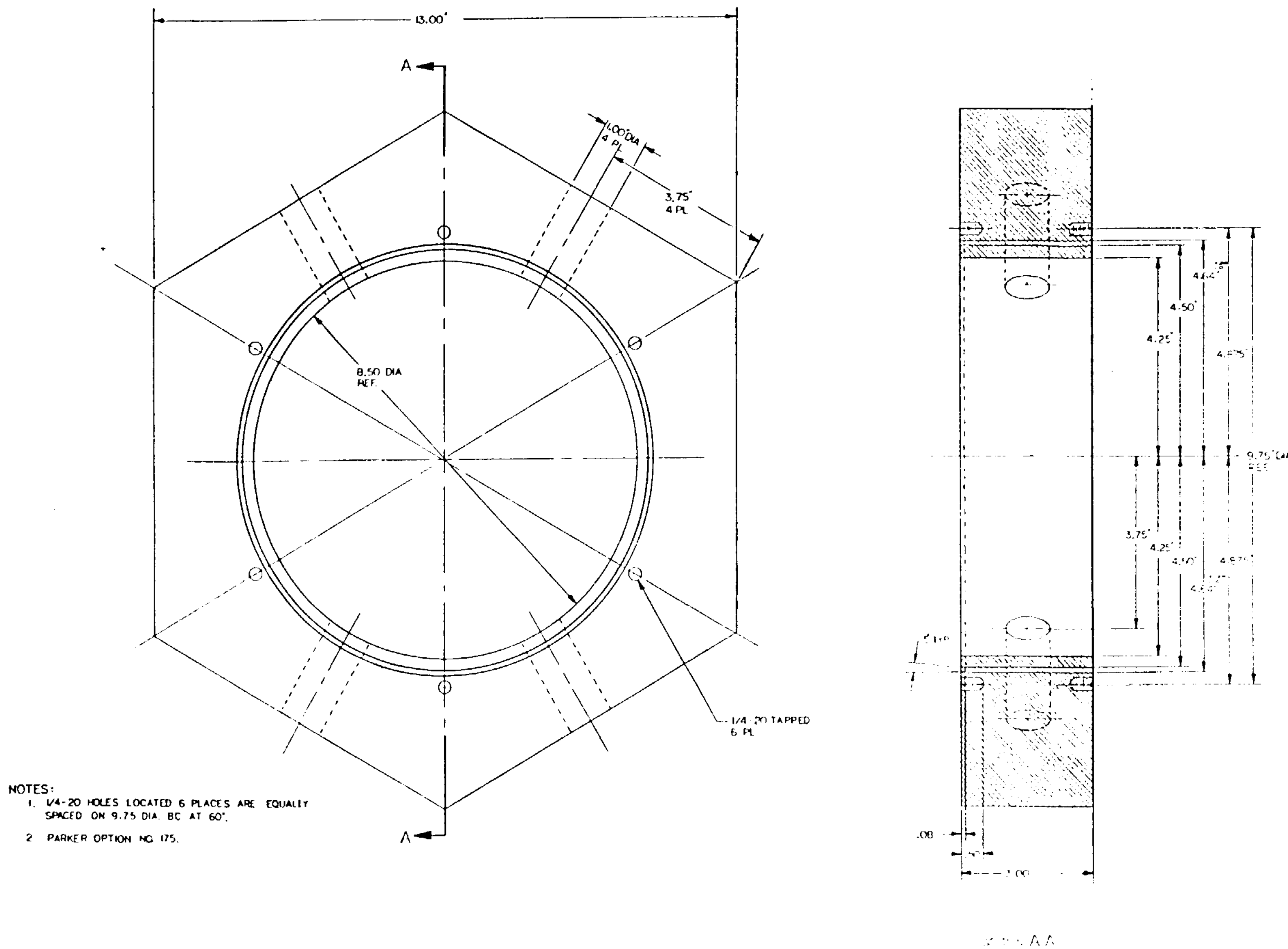

Figure 2 Cryostat Top Block

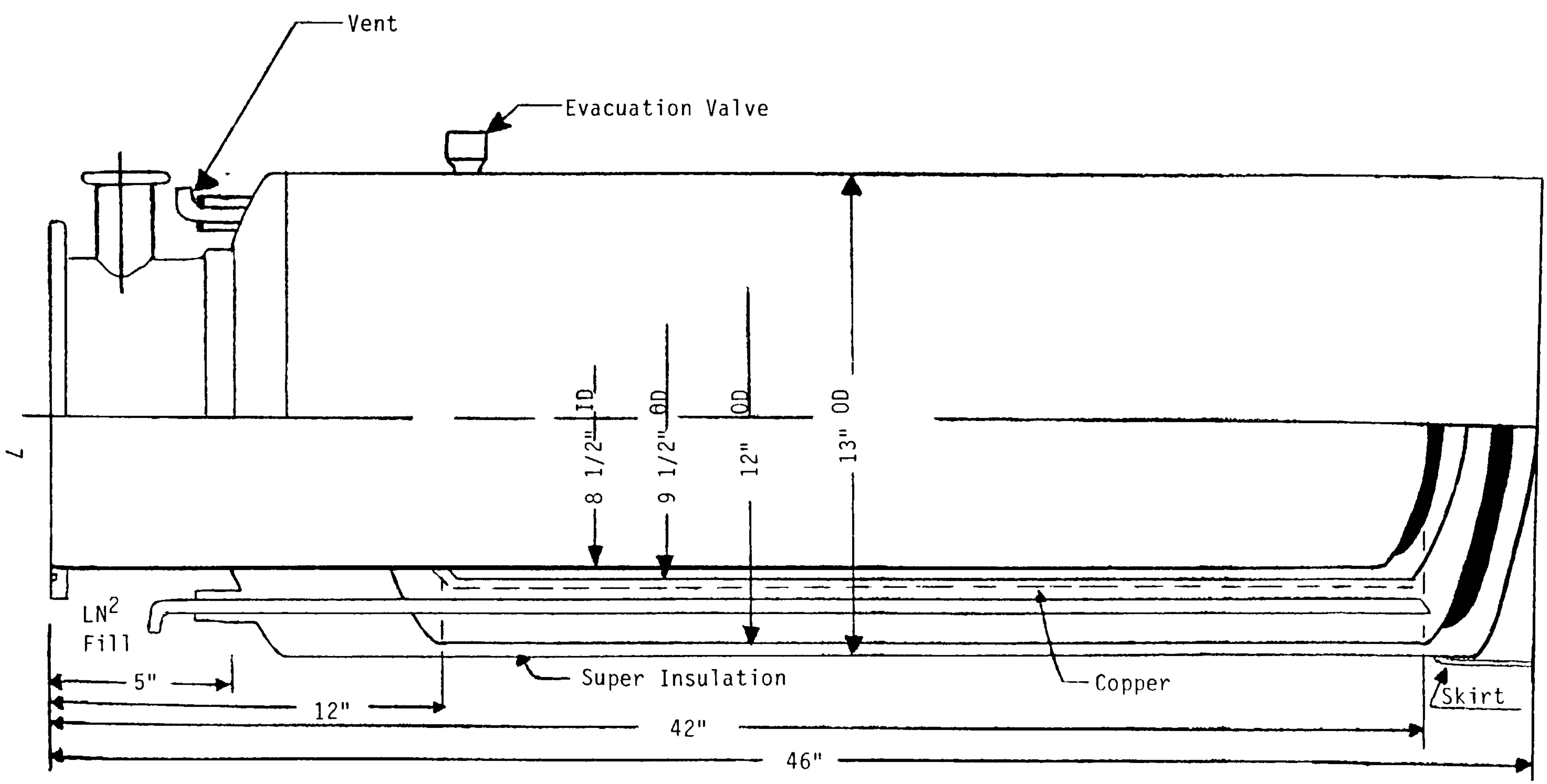

Figure 3 Test Dewar

ADR Cryostat

A scale drawing of the ADR Cryostat is shown in Figure 4. At the room temperature end of the cryostat, which is shown as the top in Figure 4, a pin feed-through is shown that allows leak tight electrical connections to pass from the laboratory into the evacuated vacuum can pumping line. This pumping line runs down to the large vacuum can at the low temperature end of the cryostat. This can is immersed in liquid helium when the experiment is being tested. The vacuum can houses the heart of the experiment. When the can is evacuated using a room temperature vacuum pump via the vacuum can pumping line, thermal isolation of the contents of the can is achieved. Running parallel to the vacuum can pumping line is the ^{4}He pumping line. This line is used to pump on the closed vessel of liquid ^{4}He known as the "^{4}He pot." This pot is located inside the vacuum can and is a source of evaporative cooling for the experiment. The ^{4}He pot typically runs at 1.4 K when the pumping is at maximum. Liquid helium is admitted from the bath into the ^{4}He pot in two ways. The first method uses a stainless steel needle valve controlled by a room temperature actuator. This valve is known as the "one shot fill valve." When the actuator is turned, liquid helium at 4.2 K quickly rushes in to fill the pot. After closing the valve, pumping can be resumed. The second way liquid can enter the pot is through the continuous fill capillary. This capillary is of small inside diameter (.016") and is 11" long with a .015" diameter 11" long wire fitted tightly inside the capillary. The resulting high impedance passageway allows a small trickle of helium to pass from the bath to the ^{4}He pot. This keeps the pot full of liquid at all times, and does not interfere with the evaporative cooling process significantly.

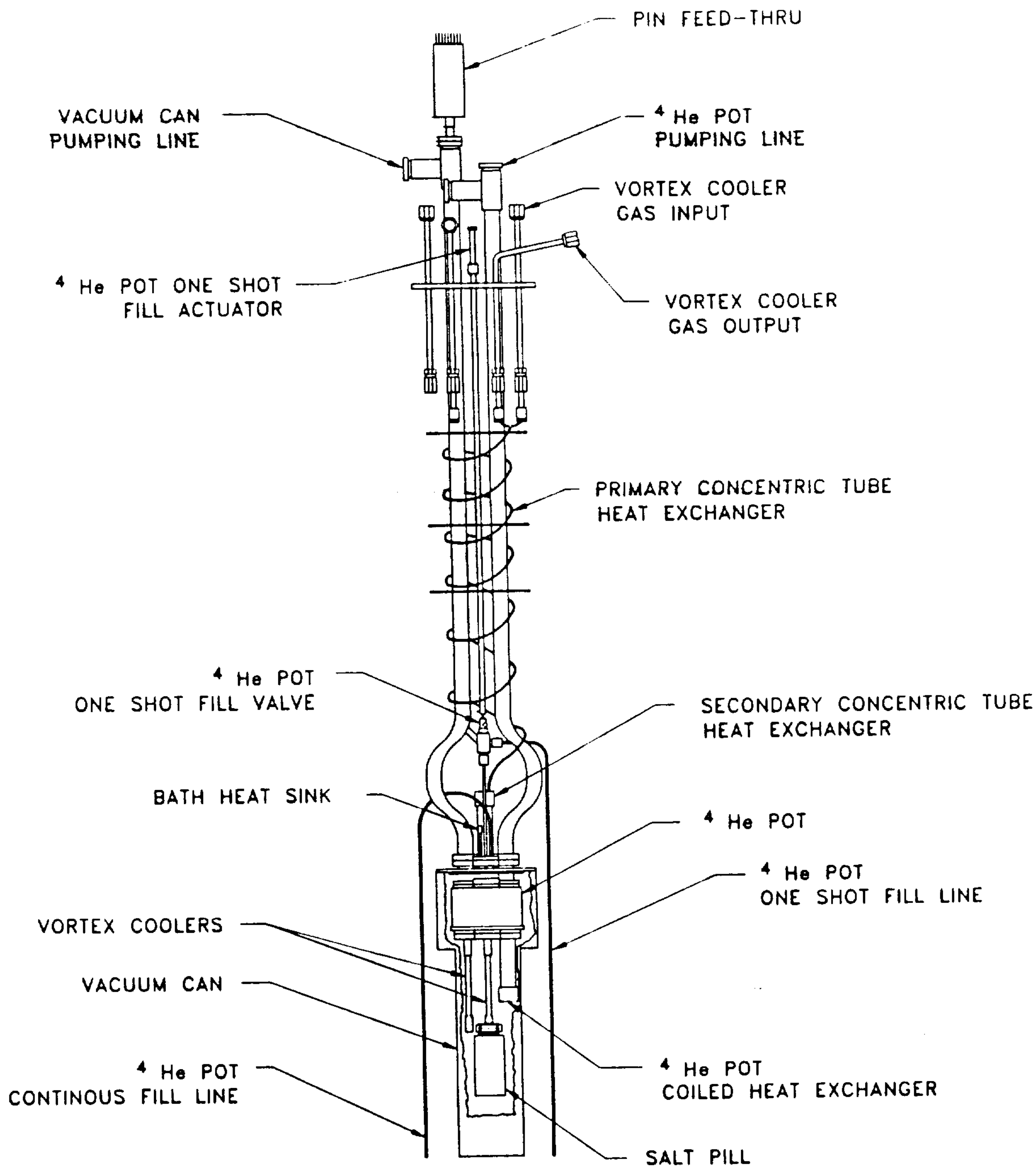

Figure 4 ADR Cryostat Configuration

It should be noted that the vacuum can and ^4He pot pumping lines are made of thermal low conductivity stainless steel to limit the amount of heat conducted from room temperature down to the ^4He bath. Radiation baffle plates lie along the middle section of these pumping lines. These plates, which are perpendicular to the tubes, block radiation that would travel from room temperature to the helium bath.

Inside the vacuum can, the vortex precoolers and salt pill assembly are suspended. The vortex coolers are attached on their upper end to the 1.4 K ^4He pot. The salt pill is attached to the lower ends of the vortex coolers via various arrangements of fluid filled capillaries and static supports that are too detailed to be described in this drawing. Further configuration details will be supplied in the Results section of this work.

Two methods were used to provide the flow of superfluid needed to make the vortex coolers function. In the first method, the gas from room temperature storage cylinders was used to supply the vortex coolers. To keep the warm gas being fed into the device from causing excessive bath boiloff, a primary concentric tube heat exchanger was constructed. This exchanged consists of 10 foot lengths of two capillaries (.050" O.D., .038" I.D. and .028" O.D., .016" I.D.) located one inside the other. Cold gas returning up the inside capillary acts to cool the warm gas entering the outside tube, and liquifaction takes place inside the heat exchanger near the bath level. After being heat sunk securely at 4.2 K, the flow enters a secondary concentric tube heat exchanger between the 4.2 K heat sink and the 1.4 K helium pot. The secondary heat exchanger is shown in Figure 5. This heat exchanger acts to prevent excessive heat leaks into the ^4He pot from the 4.2 K level. This heat

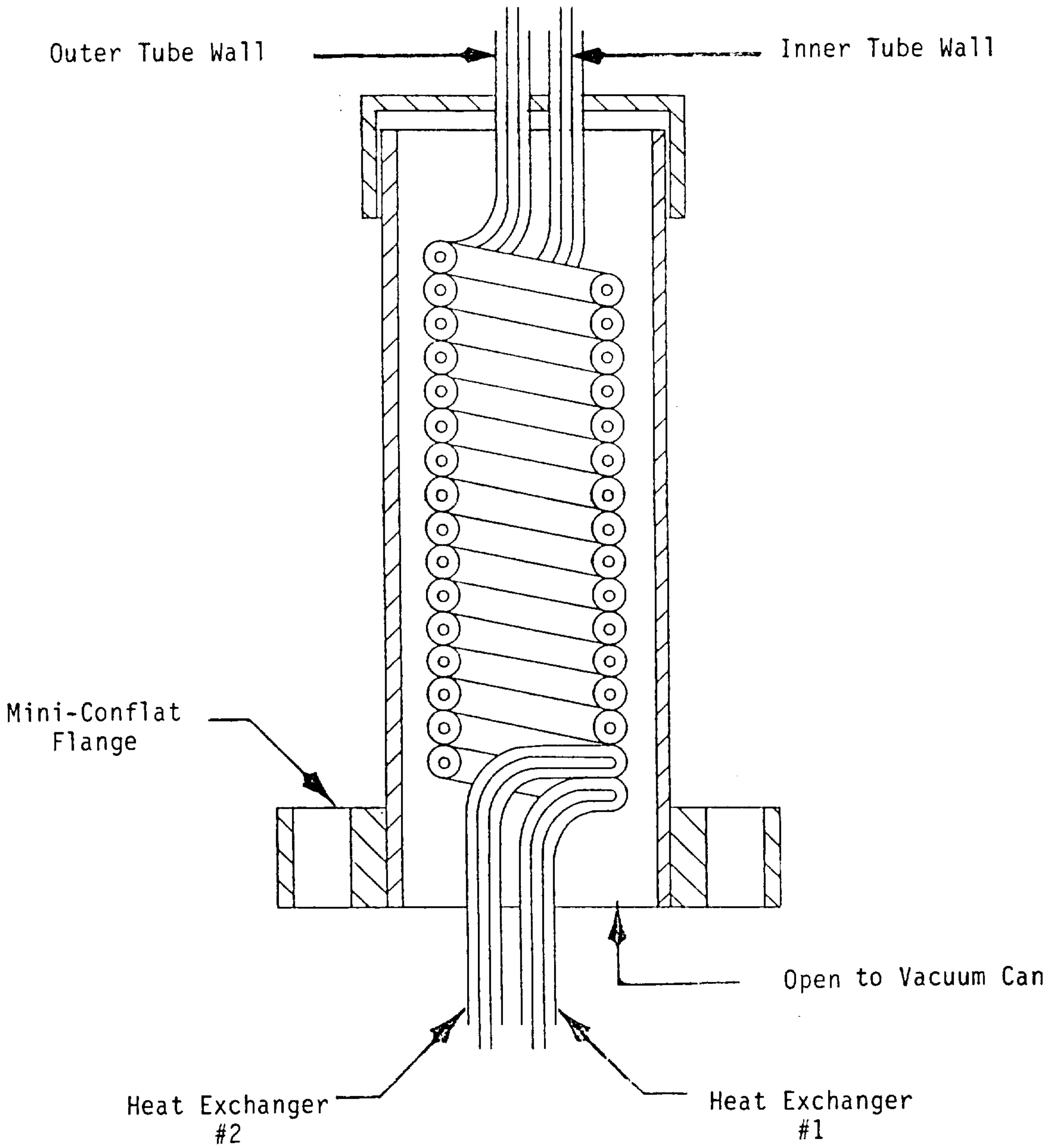

Figure 5 Secondary Concentric Tube Heat Exchanger

exchanger is then used to supply the vortex coolers with a source of flowing superfluid helium.

The second method of vortex cooler supply is accomplished using a fountain pump to drive the helium from the pot into the vortex cooler. After leaving the fountain pump, the liquid must be cooled back down to the ^{4}He pot temperature. This is accomplished with the ^{4}He pot coiled heat exchanger. Detailed descriptions of the vortex cooler, fountain pump, coiled heat exchanger and salt pill will be given in later sections. Also, the interconnections between these devices will be shown in detail for various experimental configurations.

Gas Handling System

In Figure 6, the gas handling system for the ADR cryostat is shown. This system was used to send room temperature helium gas through the heat exchangers where liquifaction took place, to the vortex cooler, and back up to room temperature. The simple panel design allowed for monitoring and controlling the flow rate through each of two vortex coolers. A bypass valve was available to connect the input and output side of the vortexes together if desired. A check valve allowed gas to escape in the event of rapid pressurization due to an unexpected system warm-up.

Magnet Support System

Two different configurations were used to support the superconducting magnet in the cryostat. In both cases, the magnet was suspended from rods

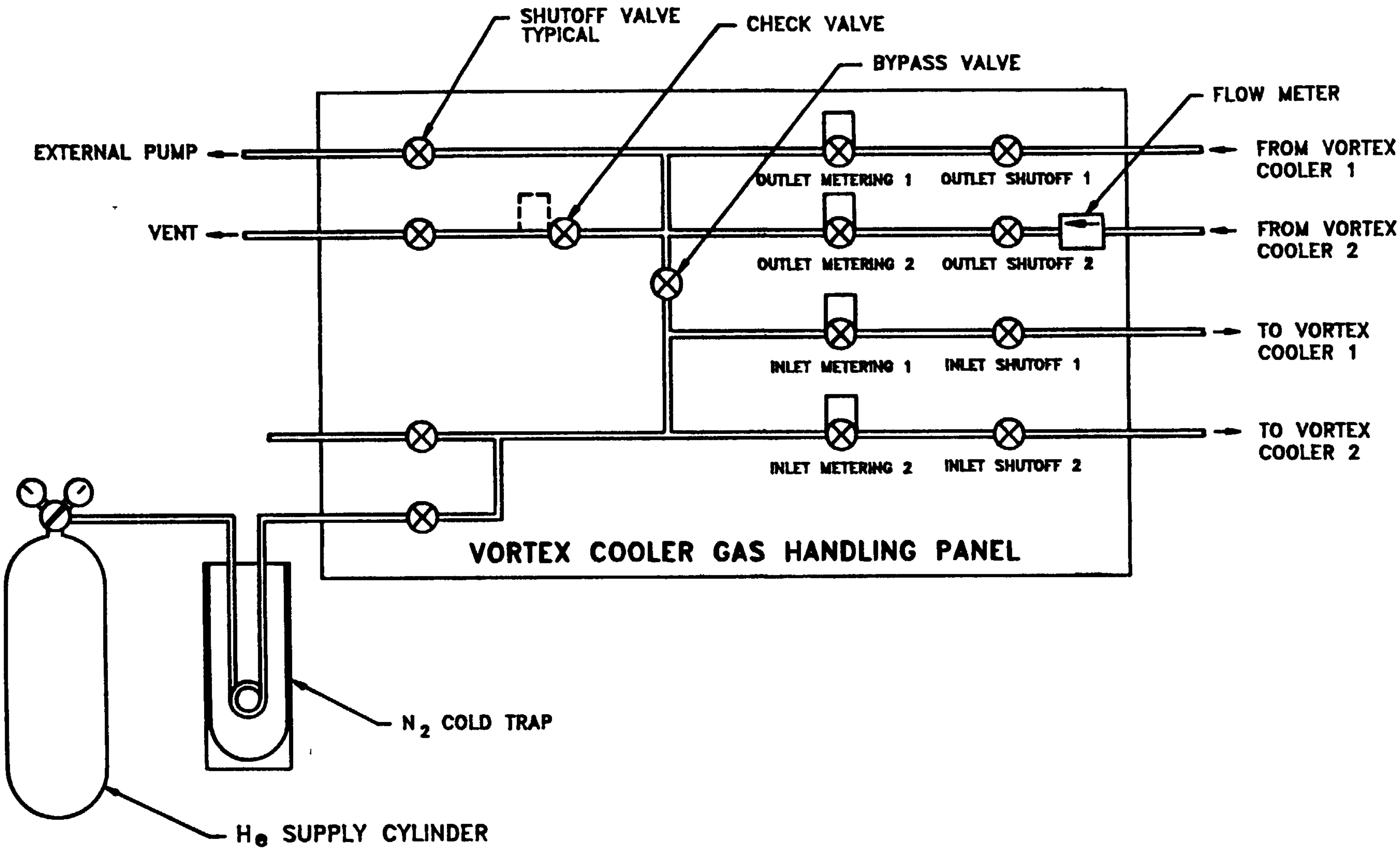

Figure 6 Vortex Cooler Gas Handling Panel

attached to the magnet support top plate. This plate is securely fastened to the cryostat top block with a rubber o-ring. When the cryostat is brought into the top plate from above, the narrow part of the vacuum can fits inside the bore of the magnet so that the salt pill is positioned at the magnet's center.

In the non-shielded configuration, shown in Figure 7, rods extend from the magnet support top plate down to the magnet support brackets, which attach directly to the magnet. The support rods used are made of epoxy fiberglass with short threaded sections of 1/4" stainless steel rods in each end.

In the shielded configuration, shown in Figure 8, three solid 1/4" stainless steel rods run from the magnet support top plate to the magnetic shield support brackets. These brackets attach directly to the soft iron magnetic shield. The magnet is then held in place relative to the shield by stainless steel magnet support braces. These braces are very rigid and prevent shifting of the position of the magnet with respect to the shield. This is important since substantial forces may exist between the shield and the magnet.

In both the shielded and unshielded cases, vapor cooled copper leads extend from the magnet support top plate down to the low temperature environment. A liquid helium level detector sensor probe is also part of the assembly.

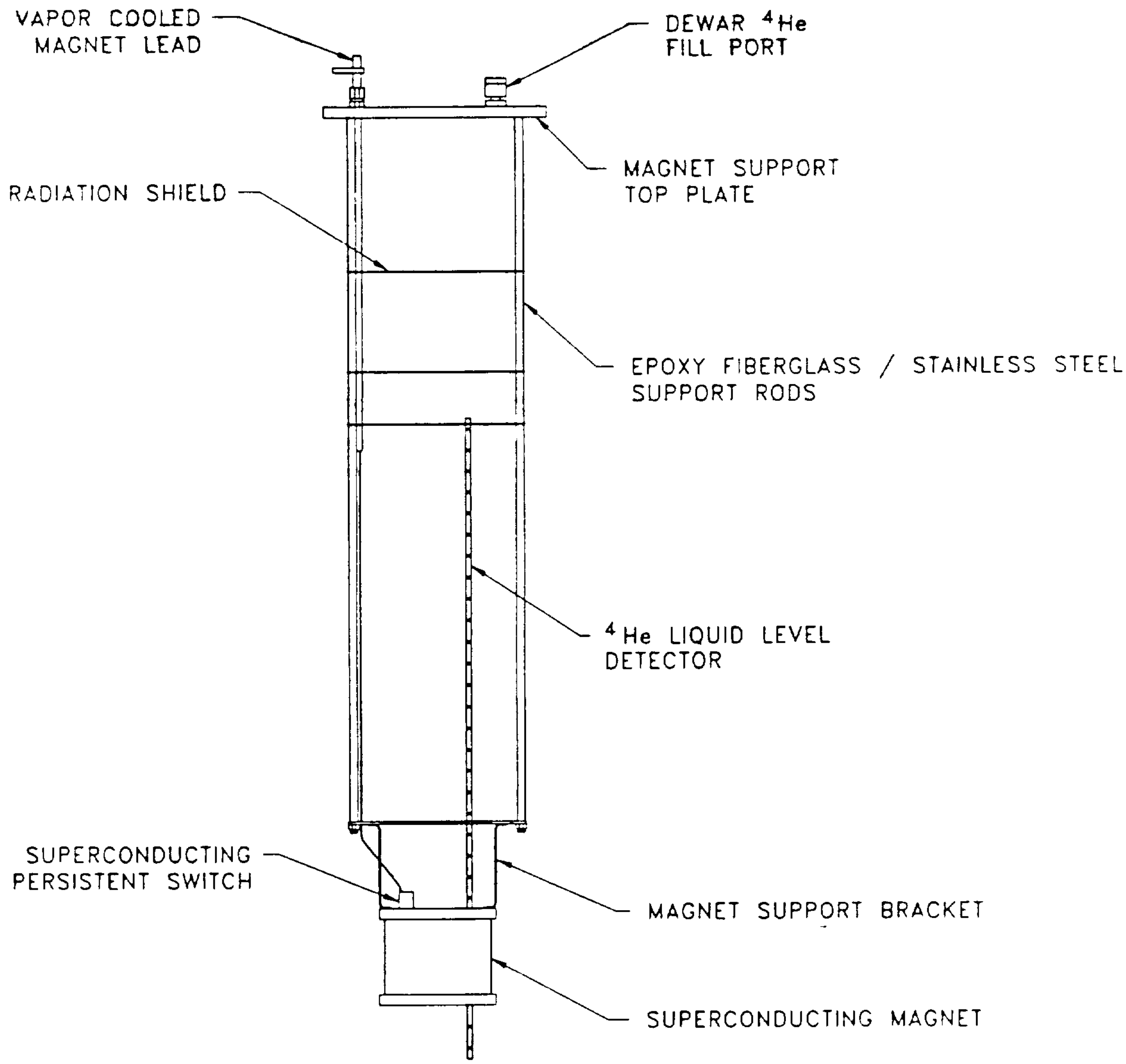

Figure 7 Superconducting Magnet Support

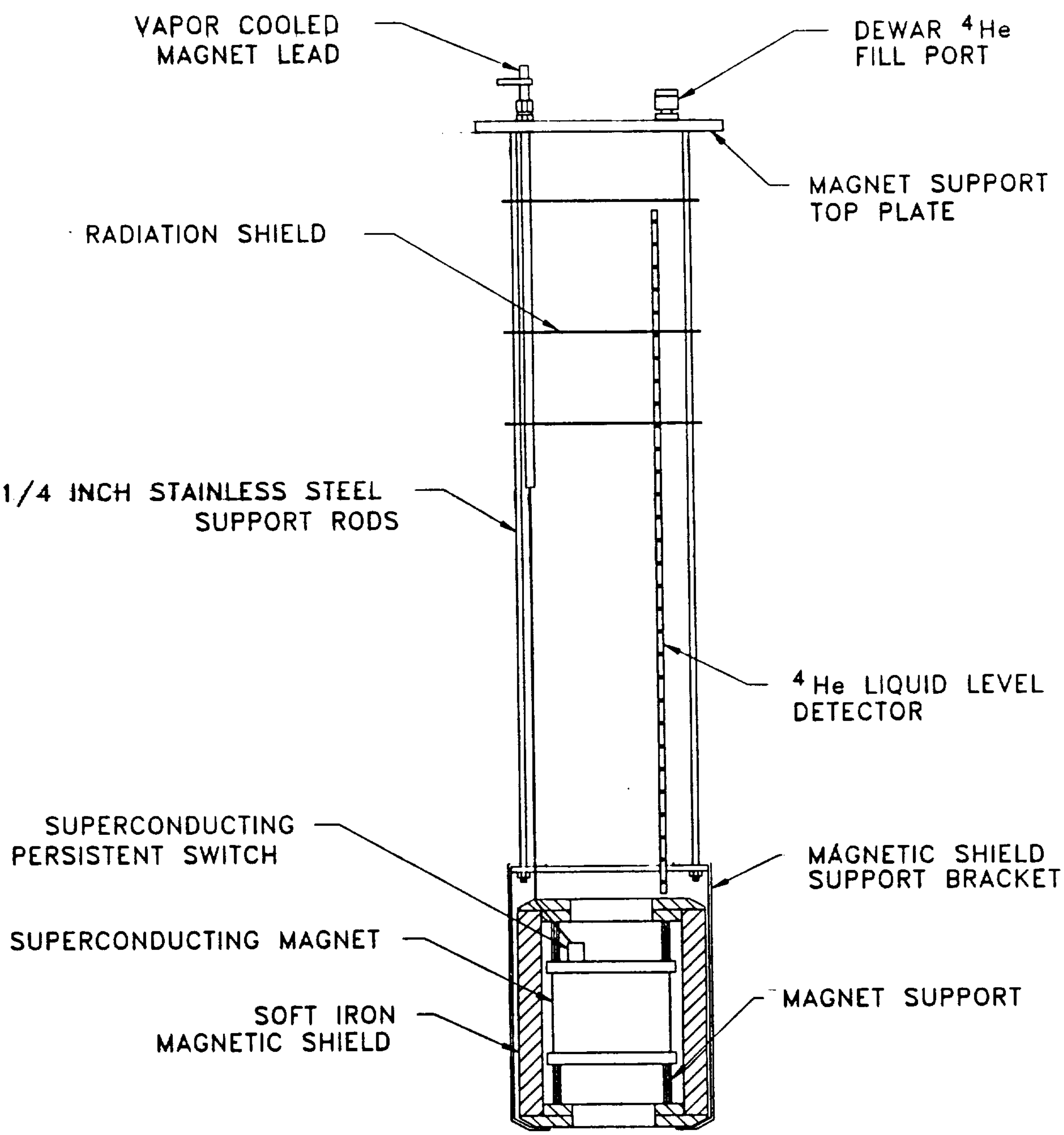

Figure 8 Superconducting Magnet Support with Soft Iron Shield in place.

ELECTRONIC AND SUPPORT EQUIPMENT

In this section, a detailed description will be given of the support equipment used in conjunction with the previously described experimental systems.

Thermometry

The heart of the temperature measurement system used in this experiment is the model 1000 potentiometric conductance bridge manufactured by Biomagnetic Technologies, Inc. This device sends a picowatt AC signal to the resistive thermometers, then measures the voltage across the sample. The measurement is frequency and phase selective, which gives excellent resolution for very small input voltages. This feature is important in low temperature experiments since it prevents dumping in large current signals that could cause local heating of the thermometers.

Model Number GR-200A-50 germanium thermometers manufactured by Lake Shore Cryotronics, Inc., were used in conjunction with the conductance bridge. These thermometers were delivered with certified calibration data and an appropriate Chebychev polynomial fit of temperature as a function of resistance.

Manganin leads (.005" and .0035" diameter) were used for wiring within the cryostat. Manganin was chosen because of its strength and low thermal conductivity, which prevents large heat leaks between stations of different temperatures. The lead wires were securely heat sunk at each temperature station by wrapping the wires and gluing them with down General Electric No. 7031 varnish.

Superconducting Magnet System

The superconducting magnet system used in this experiment was constructed for Alabama Cryogenic Engineering, Inc. by American Magnetics Corporation of Oak Ridge, TN. The specifications of the magnet are summarized in Table 1. A Hewlett Packard Model 6259B 50 amp DC power supply was used in conjunction with an American Magnetics Model 400A magnet controller to ramp the magnet. A system of vapor cooled leads designed by American Magnetics was also used. The magnet was capable of sustaining central fields of 4.5 Tesla, which was more than adequate for the test program. An American Magnetics Inc., Model 130A liquid helium level detector was used to insure that the liquid helium level stayed above the magnet at all times during operation.

Vacuum Pump Systems

For pumping ^4He from the ^4He pot, an Edwards Model E1M40 pump was used. This pump is a 40 cfm one-stage model; it was also used to rough pump the helium bath space when needed. For high vacuum pumping, a four inch oil diffusion pumping station was used. This station was a converted Veeco System used originally for Bell jar evacuation. Both of these pumps are owned by Alabama Cryogenic Engineering, Inc., and are permanently installed in the Huntsville, AL, test facility.

TABLE 1 MAGNET SPECIFICATIONS

(American Magnetics Incorporated, No. 2362)

Rated Central Field	4.5 T at 4.2 K
Rated Current	44 amps
Maximum Test Field	6.6 KG at 4.2 K
Field to Current Ratio	1023.1 Gauss/amp
Inductance	8.9 Henries
Charging Voltage (Used in Test)	1 volt
Clear Bore	3-3/8 inches (3.375)
Overall Length (Outside Flange)	4.5 inches
Maximum Outside Diameter	5.5 inches
Weight	9 pounds
Persistent Switch Heater Current	35 mA
Persistent Switch Heater Resistance	76.7 ohm
Total Magnet & Switch Resistance	203 ohm

Computer System

A Hewlett Packard computer system was used for data acquisition, processing, reduction, and display. The system consisted of:

(1) HP 9000 Series 300 Computer

(2) HP 35731 Monochrome Monitor

(3) HP 98623A BCD Interface

(4) HP 7470A Plotter

(5) HP 3421A Data Acquisition

(6) HP Think Jet Printer

(7) HP 9122 Dual Micro-Floppy Disc Drives

Miscellaneous Other Hardware

Hasting Raydist Model ST-10K Mass Flowmeters were used for the direct measurement of flow through the vortex coolers. Dynascan Corporation, Inc., Model 2831 digital multimeters were used to display voltage readouts for these flowmeters and other instruments. A Model LA-200 DC power supply was used to supply a steady current source to the fountain pump.

Auxiliary Test Facility

To aid in meeting program schedules, some of the preliminary tests were conducted using equipment already present in the Alabama Cryogenic Engineering, Inc., laboratories. In this section, a description of this auxiliary test facility will be given. This facility consisted of a large ^4He

pot and work space, and included a ^{3}He refrigerator for work extending down to 0.5 K. This facility allowed extra development to be done on vortex coolers and the ^{3}He refrigerator was used to test the heat switch principle before demagnetization runs were performed.

Auxiliary Cryostat

Figure 9 shows a schematic diagram of the auxiliary cryostat configuration. The cryostat consists of a 6 liter liquid helium pot which is suspended in a liquid nitrogen cooled Cryofab, Inc., Model CSM-85 dewar. The space around the ^{4}He pot is supported from the dewar top flange. A thin walled stainless steel pumping line serves as a helium vapor exhaust port. A large capacity mechanical pumping system was used to pump the ^{4}He pot down below the lambda transition to a minimum temperature of 1.4 K. At the bottom of the superfluid pot four mini-conflat connectors made access to the liquid helium in the pot possible. These connectors were welded to the pot and provide access to the liquid. The ^{4}He pot had a removable copper radiation shield attached to it to protect the experimental space from radiation leaks from the dewar's 77 K walls. Copper radiation baffle plates attached to the pumping line reduced radiation leaks to the pot from the room temperature dewar top flange.

All electrical leads were of 0.005" manganin wire, and were passed through the dewar top flange via room temperature ceramic feed-throughs. These leads were well heat sunk on the pumping line. This made use of the cold helium gas being removed from the system to minimize the heat leak to the pot

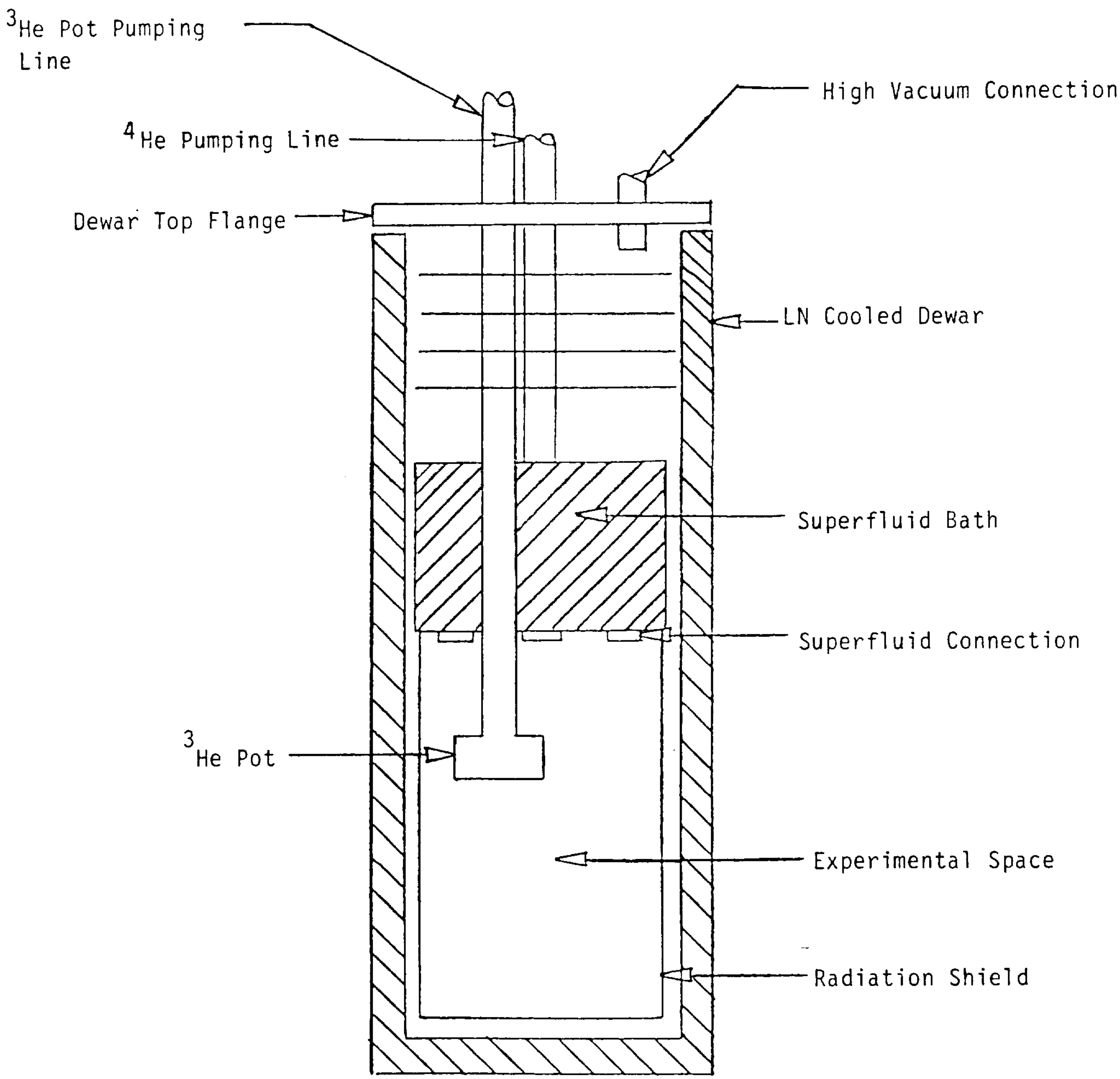

Figure 9 Auxillary Test Facility

via the leads. All capillaries and electrical leads were also well heat sunk to the superfluid pot itself.

When the cryostat radiation shield was in place, a thermal blanket consisting of 20 layers of NRC-2 superinsulation was wrapped around the pot and radiation shield to reduce the radiation leak to the pot from the 77 K dewar walls. The pot walls and radiation shield were also covered with a single layer of 3M No. 425 aluminum tape. Shu, Fast and Hart (1986) have shown that this combination of superinsulation and aluminum tape can significantly decrease heat leaks in cryogenic environments. With these precautions taken, the 6 liter helium pot could hold liquid for up to 24 hours.

The experimental space inside the copper radiation shield was a cylindrical chamber 7" in diameter and 11" in length. This space provided adequate room for various vortex cooler, fountain pump, and ^{3}He cryocooler devices. The auxiliary cryostat had no vortex cooler gas panel, since fountain pump drive systems were used almost exclusively.

One final feature of the cryostat design that facilitated modification of the apparatus was that the entire cryostat could be decoupled from support vacuum lines and electrical leads and be lifted from the dewar. Also, the dewar could be lowered as an optional method of obtaining access to the experimental space.

Electronic Instrumentation

The heart of the auxiliary cryostat electronic instrumentation system is a Biomagnetic Technologies Potentiometric Conductance Bridge (PCB). This bridge was used to measure the resistance of Cryocal Model CR100 and Lake Shore Cryotronics Model GR-200A-100 germanium thermometers. The PCB applies very small load currents (picowatts) to the resistors, and thus avoids self-heating in the thermometer elements.

Hastings ST Series mass flowmeters were used to measure the helium gas flow rates. These gauges give a 0-5 volt D.C. output that is linear with mass flow over their calibration range; also, these devices are pressure independent. Setra Pressure gauges were used to monitor pressures in the system. These gauges give out a 0-5 volt D.C. voltage that is linear with pressure.

Computational work and data reduction was done on a Hewlett-Packard Model 98165 computer. Data was printed out on a Hewlett-Packard Model 82905B printer and plotted using a Hewlett-Packard Model 7470A plotter. An H.P. statistical graphics package was also used for data reduction and presentation.

^{3}He Pumping and Gas Handling System

In Figure 10, a schematic representation of the ^{3}He cryocooler pumping system is given. In normal operation, the ^{3}He vapor was removed by the pump via the "out" port of the ^{3}He refrigerator. The vapor then passed through a

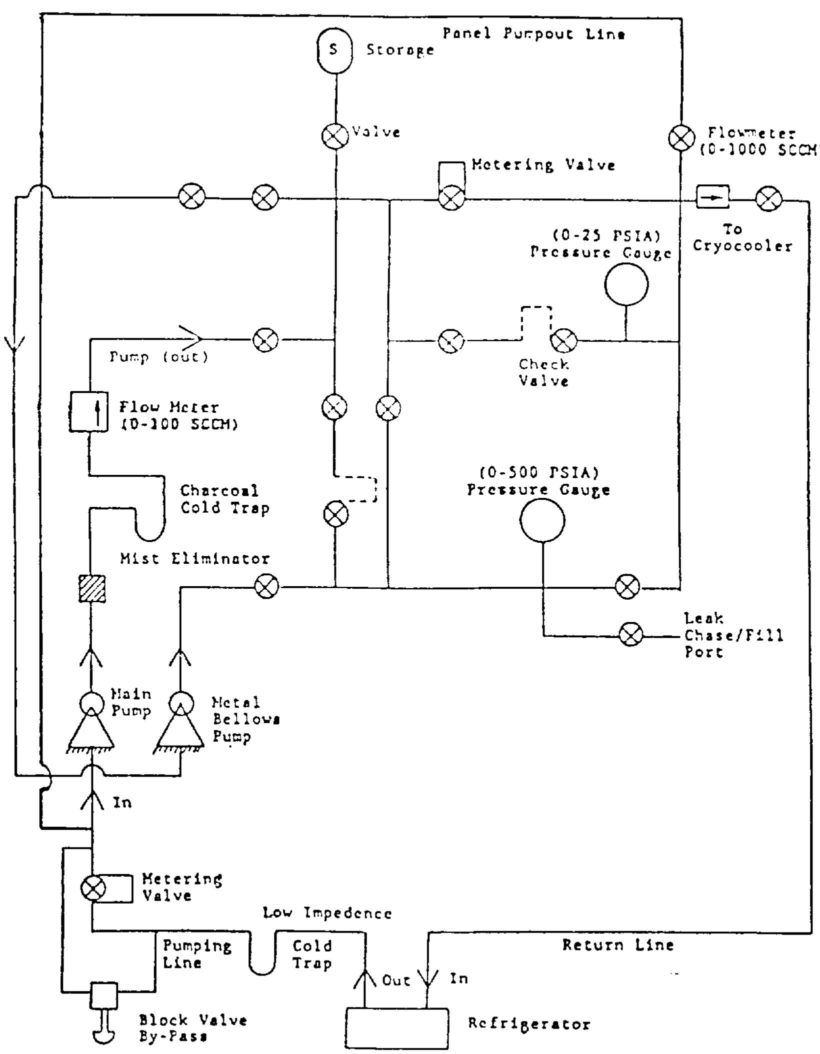

Figure 10 ^{3}He Cryocooler Gas Handling System low impedance cold trap designed to prevent back streaming of pump oil into the refrigerator. The pumping speed is then regulated by the block and metering valves shown at the pump inlet. After passing through the Alcatel Model 2012H hermetically sealed pump, the ^{3}He vapor passes through an oil mist eliminator, and into a charcoal cold trap. The output of the pump is then measured with a Hastings ST-100 mass flowmeter. The gas then enters the main body of the gas panel, where pressure monitoring is done with Setra pressure gauges. After passing through a needle metering valve, the ^{3}He enters a Hastings ST-100 mass flowmeter, and then back into the cryocooler via a return line. This is the typical configuration used during a continuous cycle run.

Other important features of the system are a 37.4 liter storage volume, where the ^{3}He sample (10 standard liters) is stored. A Metal Bellows hermetically sealed pump is also attached to the gas handling panel to facilitate removal of the gas from the storage can during its condensation into the cryocooler. This gas handling and pumping system offered great flexibility in controlling the refrigeration cycle and in monitoring system parameters. The ^{3}He cryocooler described above could reach a minimum temperature of approximately 0.4 K, and could provide 0.3 mW of cooling power at 0.65 K.

IV. VORTEX PRECOOLER DEVELOPMENT

In this section, a brief introduction to vortex coolers will be given along with a listing of some of the previous work conducted by others. A baseline vortex cooler system as required for use in the ADR heat switch will be defined. The design of the ACE, Inc. vortex cooler will be presented, and its performance will be discussed. A description will be given of the observed effect on vortex cooler performance when design parameters are varied.

Background Work

The concept of forcing superfluid helium through a tightly packed solid powder to achieve cooling was first proposed by Kapitza and Simon in 1945. In 1967, Olijhoek performed the first experiments to verify the feasibility of this idea. The experiments were repeated in 1969 by Stass and Severins, who named this type of device the vortex cooler. Other work followed (Olijhoek et al., 1973, 1974 and Satoh et al., 1982, 1983), but the device has not gained widespread use simply because its operating temperature range 0.7 - 2.2 K can be covered easily in terrestrial laboratories by more standard means, such as with ^{3}He evaporative refrigeration. However, the vortex cooler has the unique advantage over other cooling cycles that it does not involve a liquid-vapor phase separation; it is therefore ideally suited for zero gravity work.

In Figure 11 a schematic diagram of a vortex cooler is given. Superfluid helium is forced to flow through a superleak made of very fine packed powder. According to the two fluid model, the super component of the

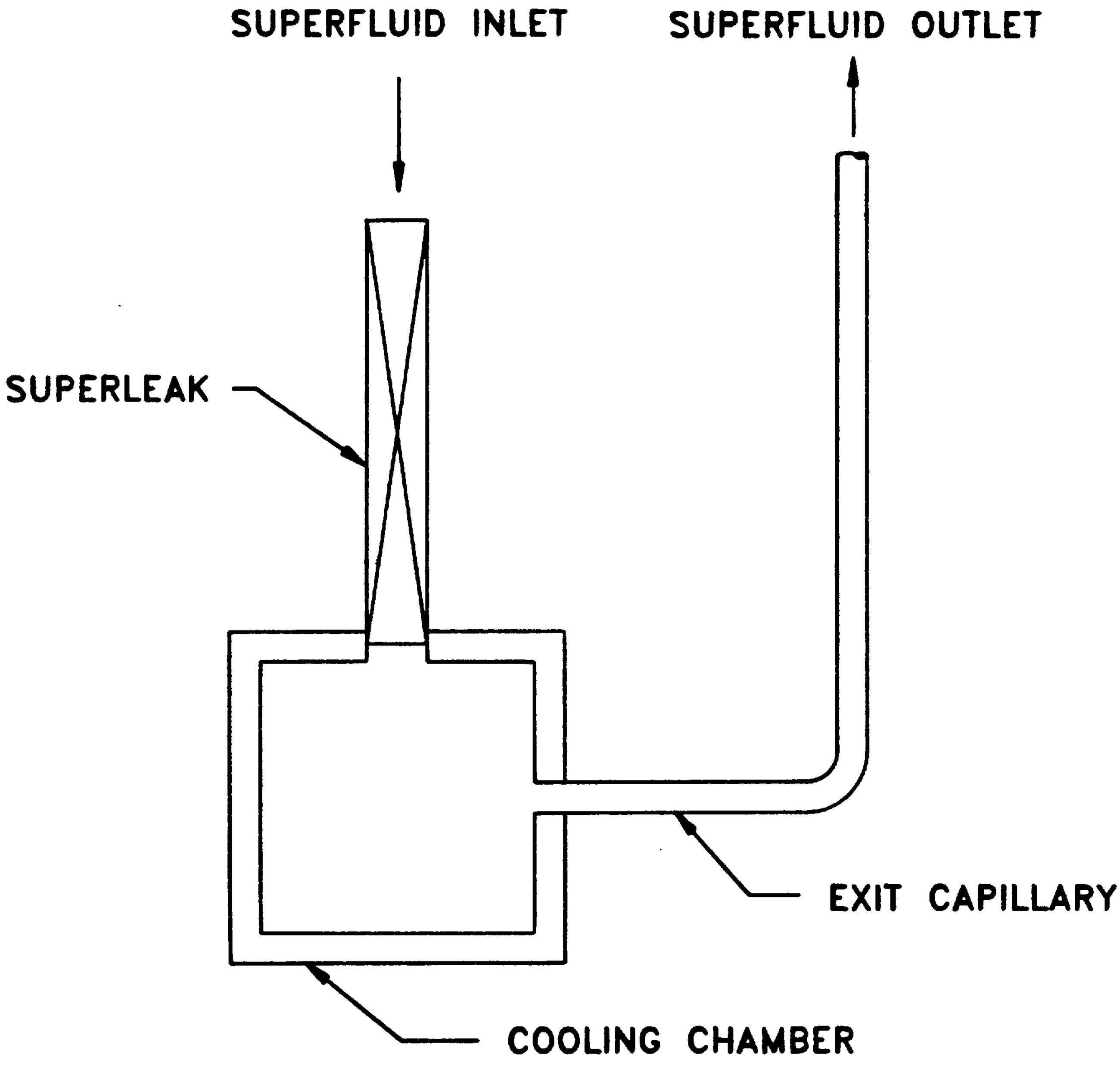

Figure 11 Schematic Representation of a Vortex Cooler

fluid is inviscid and will pass through the superleak while the normal component will not. Driving superfluid through the superleak thus should result in lowering the entropy of the fluid in the cooling chamber; a cooling effect is observed. Excitations are swept away from the cooling chamber by the flow through the small diameter exit capillary. Cooling of the chamber at the end of the superleak is thus achieved.

It should be noted that a coherent theory of the operation of the vortex cooler has not yet been developed. It is not clear why the operation of the cooler is limited to a minimum temperature of approximately 0.7 K. The purpose of our study has been to attempt to optimize the performance of the cooler by varying construction parameters. The results of this study will be presented, along with the design of the ACE, Inc. vortex cooler.

Determining Baseline Vortex Cooler Parameters

To set requirements for the vortex precooler system, we must first obtain desired baseline parameters on the function of the ADR. We will use the same idealized baseline ADR requirements as given by Castles (1980). These requirements are summarized in Table 2.

From these requirements, we see that the desired refrigerator power is 50 μW at 0.1 K. Assuming that the ADR cycle approaches Carnot efficiency gives the following value for the heat Q_B to be removed during magnetization:

$$Q_B/Q_A = T_B/T_A \quad (1)$$

TABLE 2 BASELINE REQUIREMENTS FOR GSFC ADR DESIGN. (From Castles, 1980)

Adiabatic Demagnetization Refrigerator

- Operating Temperature is 0.1°K
  - Higher operating temperatures possible with greater cooling power
  - Excellent temperature stability
- 50 μW Cooling Power
- Operating Time Greater than 90 Minutes
  - Recycle time of 10 minutes
- Highly Efficient
  - Thermodynamic efficiency approaches Carnot
  - Expels less than 2 mW to the liquid helium bath
- High Reliability
  - No moving parts
- Mechanical Design Appropriate for a Shuttle Launch

where Q_B = heat removed during magnetization

T_B = temperature of magnetization

Q_A = heat removed during cooling cycle

T_A = cooling temperature.

We will assume that using the vortex precooler will enable the heat of magnetization to be removed at 1.1 K. Taking T_A = 0.1 K and using a heat consumption rate of 50 mW for 90 minutes, we obtain Q_A = 0.27 J at 0.1 K using equation (1), we require Q_B = 2.97 J at 1.1 K. Note that if the vortex precooler was not used, and the cycle was run at the space dewar temperature of 1.8 K, a larger value Q_B = 4.86 J would have to be removed.

To meet the desired performance criteria, the vortex precooler for the ADR would have to remove the heat Q_A in the baseline recycle time of 10 minutes. From these numbers, we set the desired vortex cooler cooling power to be 4.95 mW at 1.1 K. This cooling could conceivably be achieved by a single vortex cooler, if its cooling power is great enough, or by the use of multiple vortex coolers in parallel.

For the vortex cooler to work effectively as a heat switch, it must be able to reach a sufficiently low operating temperature. This is because the heat switch is based on the thermal conduction characteristics of superfluid helium filled capillaries. These thermal transport characteristics are described in detail by Bertman and Kitchens (1968). The thermal conduction along a superfluid filled capillary is very strongly dependent on the temperature at each end of the capillary. In Figures 12 and 13, their results for the heat flow are shown. The quantity plotted is

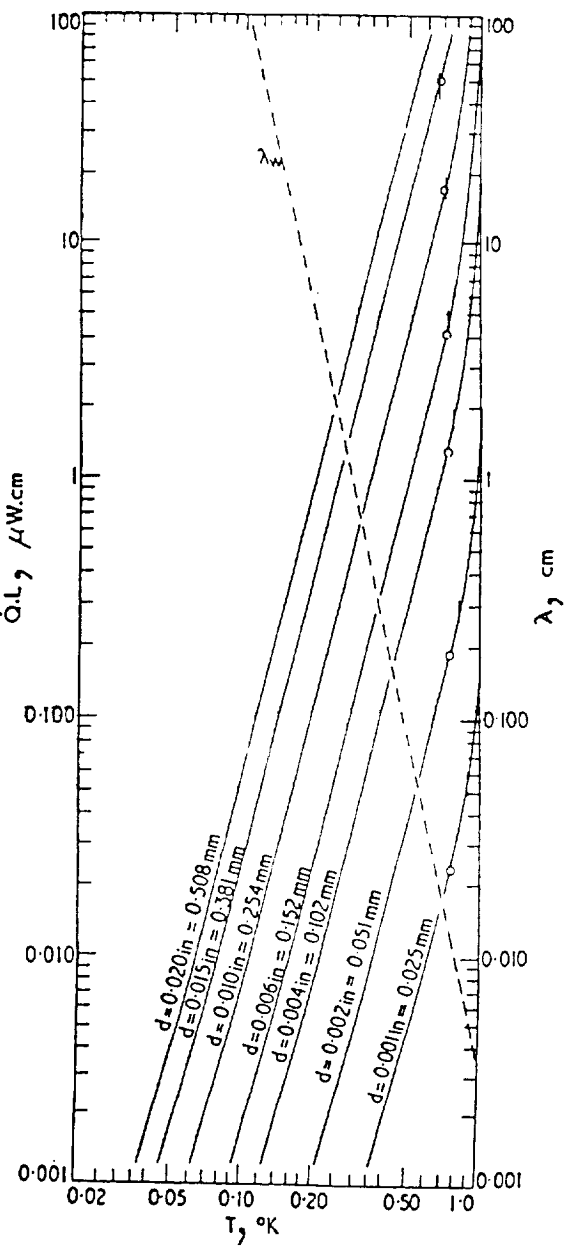

Figure 12 Heat Leak Along Superfluid Filled Capillaries as a Function of Temperature for (0.02 K < T < 1.0 K). The data is shown for different capillary diameters.

Graph from Bertman and Kitchens (1968)

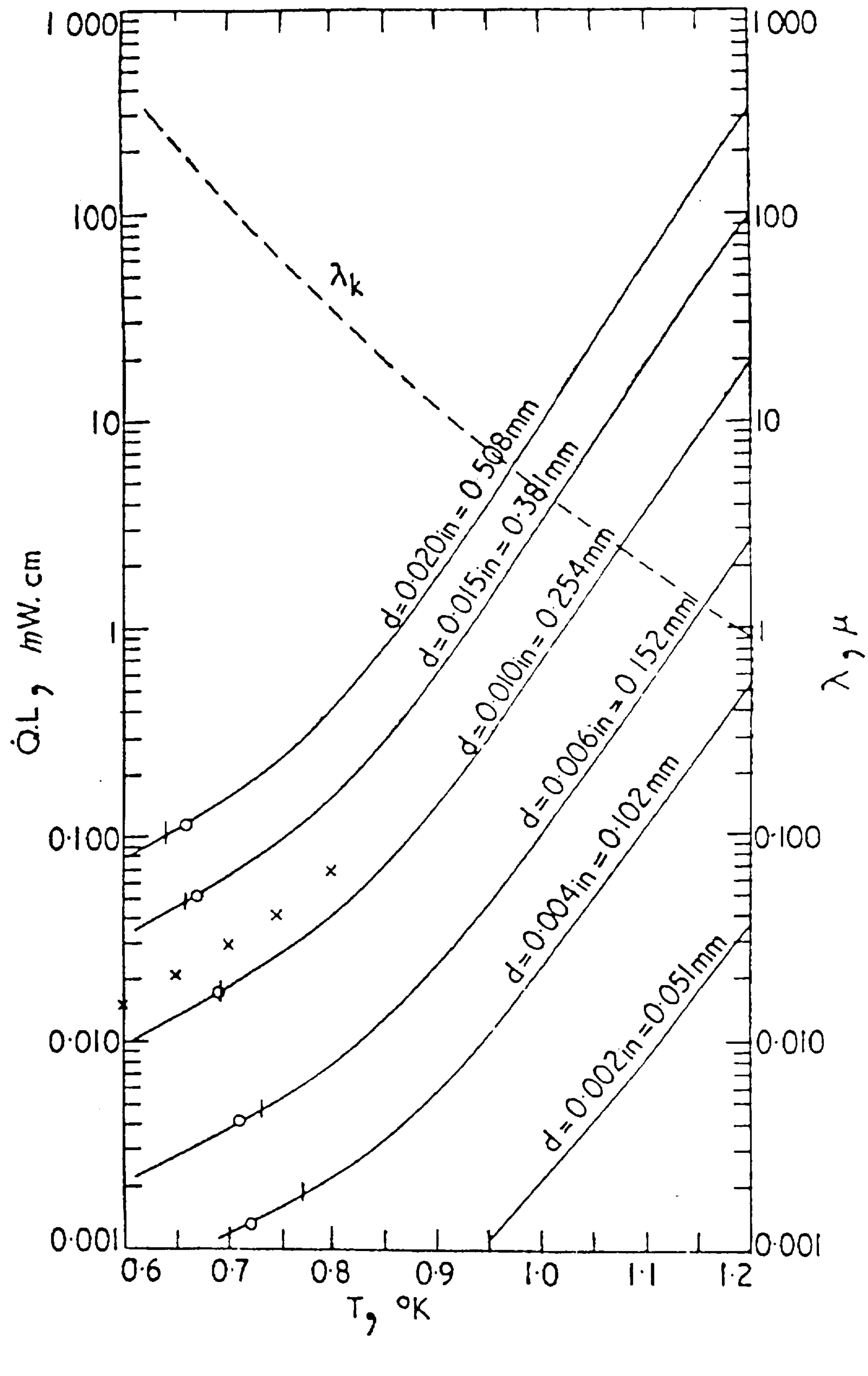

Figure 13 Heat Leak Along Superfluid Filled Capillaries as a Function of Temperature for (0.6 K < T < 1.2 K). The data is shown for different capillary diameters.

Graph from Bertman and Kitchens (1968)

$$Q \cdot L = \int K_L \, AdT \tag{2}$$

where Q = heat flow [W]

L = tube length [cm]

A = tube cross sectional area [cm^2]

K_L = thermal conductivity [W/(K·cm)]

If T_C is the temperature of the cold end of the tube and T_H is the temperature of the hot end, then the heat flow along the tube is given by

$$Q = 1/L \left[\int_0^{T_H} K_L \, AdT - \int_0^{T_C} K_L \, Adt \right] \tag{3}$$

which can be readily obtained from Figures 12 and 13 for a given temperature profile.

As an example, we will use these results to calculate the heat flow for a 0.254 mm inner diameter capillary 50 cm long. We will assume one end of this capillary is at the baseline ADR refrigeration temperature of 0.1 K, and plot the heat flow along the capillary in microwatts as a function of the temperature at the upper end of the capillary. These results are shown in Table 3. These capillary dimensions represent realistic prototype specifications. It should be noted that increasing T_H from 0.7 K to 1.3 K gives a heat switching ratio of almost 10^4.

From Table 3, it seems reasonable to set the required baseline minimum temperature of the vortex cooler module to be 0.80 K.

TABLE 3 HEAT CONDUCTION AS A FUNCTION OF THE TEMPERATURE AT THE HOT END OF A 50 CM LONG 0.3 MM I.D., CAPILLARY. THE OTHER END OF THE CAPILLARY IS ASSUMED TO BE HELD AT 0.1 K.

T_H (K)	Q (μW)
0.70	0.21
0.80	0.65
0.90	2.64
1.00	13.8
1.10	69.8
1.20	400.0
1.30*	2000.0

*Extrapolated

Development of the ACE, Inc. Vortex Precooler

In this section, the development program of the ACE, Inc. vortex cooler will be discussed. One of the developments was a comparison of (a) fountain pump and (b) room temperature gas supplies via concentric tube heat exchangers as methods of driving superfluid circulation in the vortex coolers. The effect of varying vortex cooler/fountain pump design parameters on the cooling power and minimum temperature of the cooler is also examined.

Superfluid Circulation Method

Two options are available to drive superfluid through vortex coolers. One method is to use a fountain pump, which is a device that drives superfluid helium via the thermomechanical effect (See Guenin and Hess, 1980). To build a fountain pump, a superleak is attached to a reservoir of helium held below the temperature of the superfluid transition (T_λ = 2.17 K). Heat applied to the other end of the superleak causes a pressure gradient according to

$$\Delta P = \rho S \, \Delta T \tag{4}$$

where ρ and S are the fluid density and entropy, respectively. This pressure difference then drives the fluid into a capillary. Before entering the vortex cooler, the fluid is returned to the helium reservoir temperature by means of a coiled tube heat exchanger which runs through the bath.

An alternate method of providing superfluid flow to the vortex cooler involves the use of a room temperature supply of pressurized helium gas. To prevent transport of excessive amounts of heat down to the low temperature stations, concentric tube heat exchangers are used. These exchangers take advantage of the enthalpy change of the cold gas returning from the vortex cooler to precool the incoming flow. Two such heat exchangers are used. The first one runs between the cryostat top plate down into the vacuum can, and operates between room temperature and 4.2 K. The second exchanger runs between the 4.2 K station down to the ^4He pot at approximately 1.4 K. The second exchanger is needed to prevent excessive boiloff in the ^4He pot. If the pot becomes depleted, the action of the vortex cooler is disrupted.

Both of these supply methods were tested, and vortex cooler temperatures lower than 0.85 K were achieved in both configurations. However, the liquid helium pot size in the ADR test apparatus was severely constrained by the superconducting magnet support interface. This constraint contributed to the difficulty of matching ^4He pot continuous fill rates with the pot depletion rate. This depletion rate depended on the conductive path of the superfluid in the second superfluid concentric tube heat exchanger and on the continuous fill tube flow rate, which had to be determined by trial and error. Therefore, the fountain pump drive method was judged to be superior for testing purposes, although it should be noted that the room temperature gas supply method is a viable alternative.

Optimizing Vortex Cooler Performance

As previously mentioned, the theory of operation of vortex coolers is not presently advanced enough to allow theoretical optimization of vortex cooler/fountain pump systems. Therefore, tests were conducted to optimize vortex cooler performance. These tests were divided into four categories: (1) varying the diameter of the vortex cooler exit capillary; (2) varying the cross-sectional area of the fountain pump capillary; (3) varying the diameter of the fountain pump superleak; (4) varying the length/diameter ratio of the vortex cooler superleak. These four tests were carried out on the auxiliary test apparatus described in Section III. The findings of each of these tests will now be briefly summarized. It should be noted that the examples shown in each of these tests do not represent the fully optimized ACE vortex cooler, but serve to investigate general performance characteristics.

Dependence on Vortex Cooler Exit Capillary Diameter

In Figure 14, the temperature of the vortex cooling chamber is plotted as a function of power input to the fountain pump. The datum shown here represent the results obtained for four different diameters of the vortex cooler exit capillary. For each of these tests, the exit capillary length and all other geometrical parameters of the rest of the vortex cooler/fountain pump system were held constant. Little difference is seen for the 0.4, 0.5, and 0.7 mm capillary data. The 0.25 mm capillary configuration is seen to be somewhat less able to reach a given temperature for the same fountain pump power as the other cases.

In Figure 15, the temperature of the vortex cooler cooling chamber is plotted as a function of heat load applied to the cooling chamber. For each curve, a different constant fountain pump power has been used. This fountain pump power has been selected for each capillary diameter by minimizing the temperature of the vortex cooler under zero vortex cooler heat load

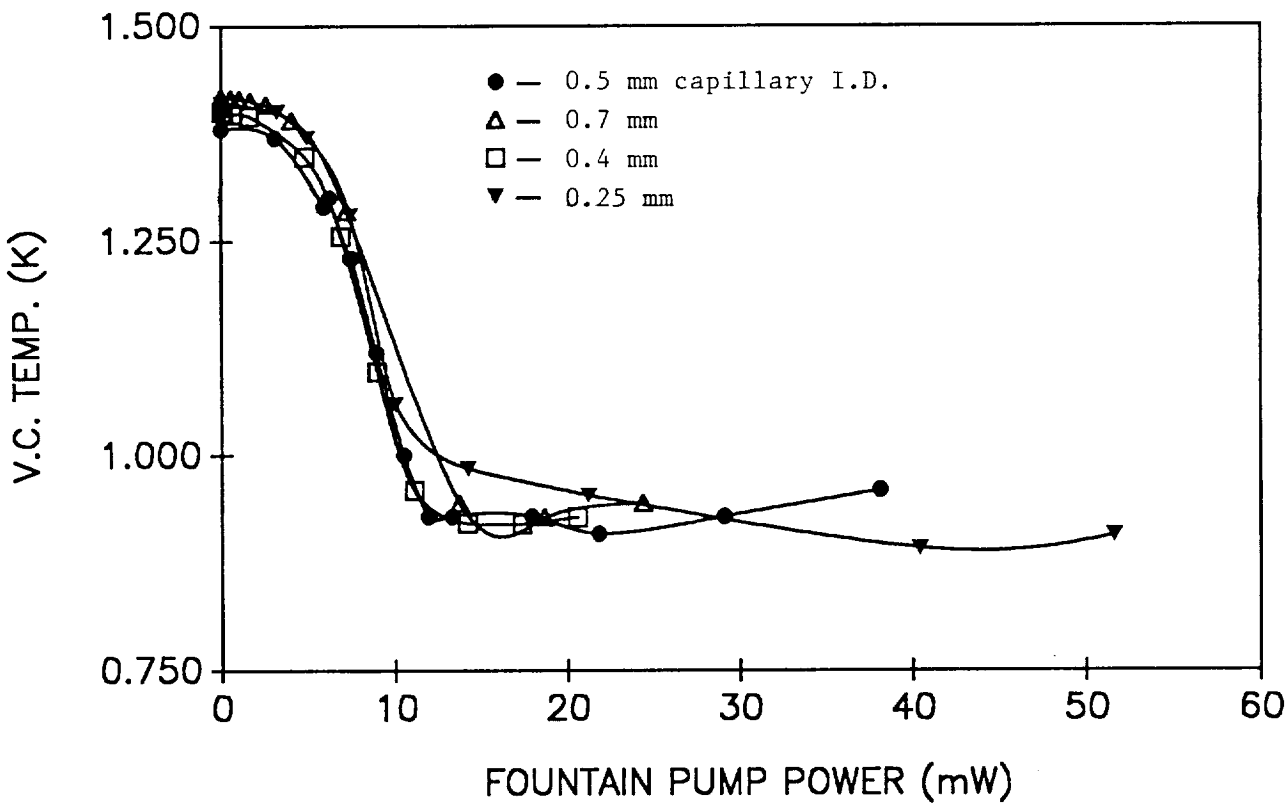

Figure 14 Vortex Cooler Temperature as a function of Fountain Pump Power for various diameter Vortex Cooler Exit Capillaries.

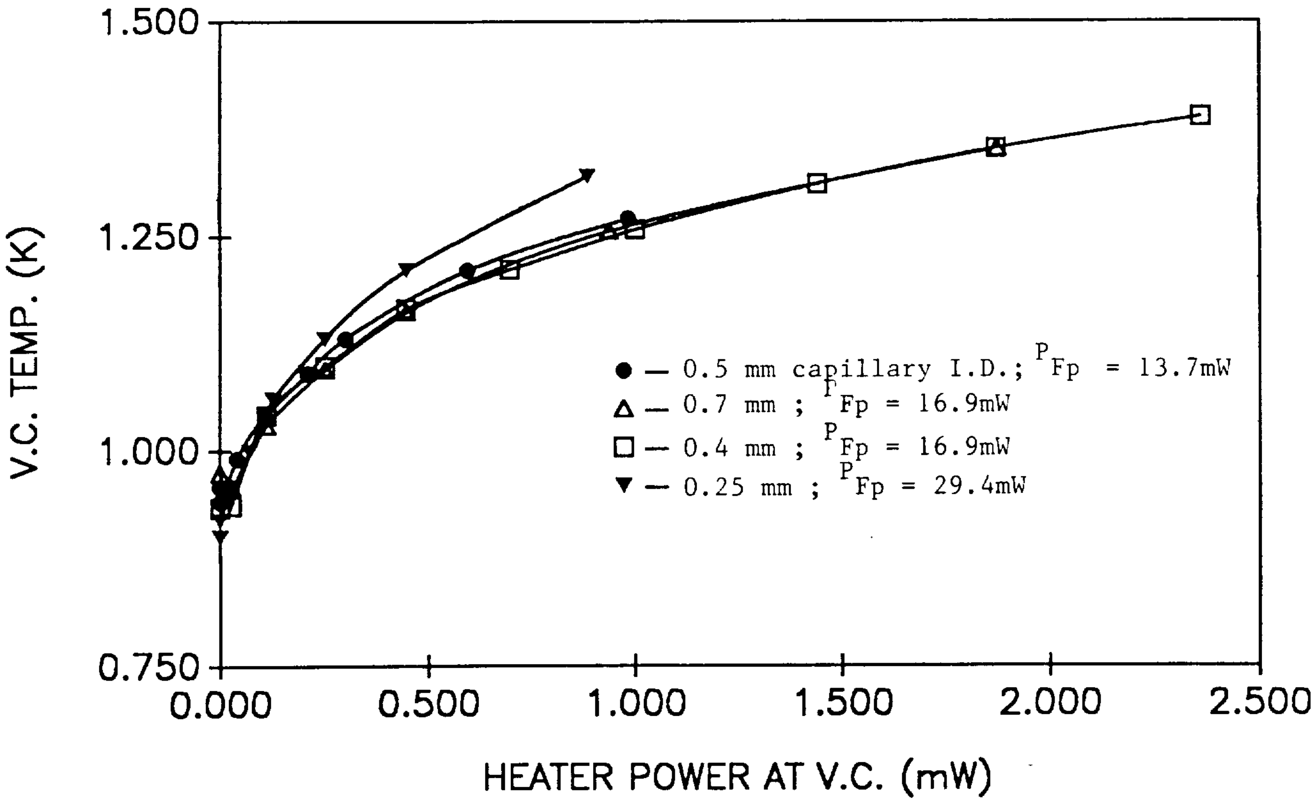

Figure 15 Vortex Cooler Temperature as a function of Heat Load applied to the Vortex Cooler for various Vortex Cooler Exit Capillary Sizes. The Fountain Pump Power P_{Fp} for each curve has been adjusted to minimize the temperature of the Vortex Cooler under zero Heat Load.

conditions. As was evident from Figure 14, for each vortex cooler configuration there is a unique fountain pump power that results in minimizing the temperature of the cooling chamber. The fountain pump power used for each data set is inset in Figure 15. For the 0.4 mm, 0.5 mm, and 0.7 mm capillaries, similar heat loading performance can be obtained for all three capillary sizes. However, it should be noted that the vortex with the 0.5 mm capillary achieves the same performance as the others while using significantly less fountain pump power. It therefore seems reasonable to infer that for a given vortex cooler/fountain pump configuration, there is an optimum vortex cooler capillary diameter that will result in the most efficient performance.

In Figure 16, the effect of decreasing the cross-sectional area of the fountain pump capillary while holding all other parameters fixed is illustrated. Vortex cooler temperature is plotted as a function of fountain pump power for two different capillary configurations. The solid circles show the results obtained when a 1 mm diameter capillary was used; the open circles show results for the same capillary after a 0.8 mm O.D. wire was placed inside it. From the graph, we see that inserting this wire significantly increased the efficiency of the fountain pump. Inserting the wire in the capillary decreased the cross-sectional area of the wire from 1.0×10^{-2} to 6.4×10^{-3} cm^2. The extent to which the change in performance depends on moving from cylindrical to annular capillary geometry is not known.

In Figure 17, vortex cooler temperature is shown as a function of fountain pump power for two differing geometries of the fountain pump

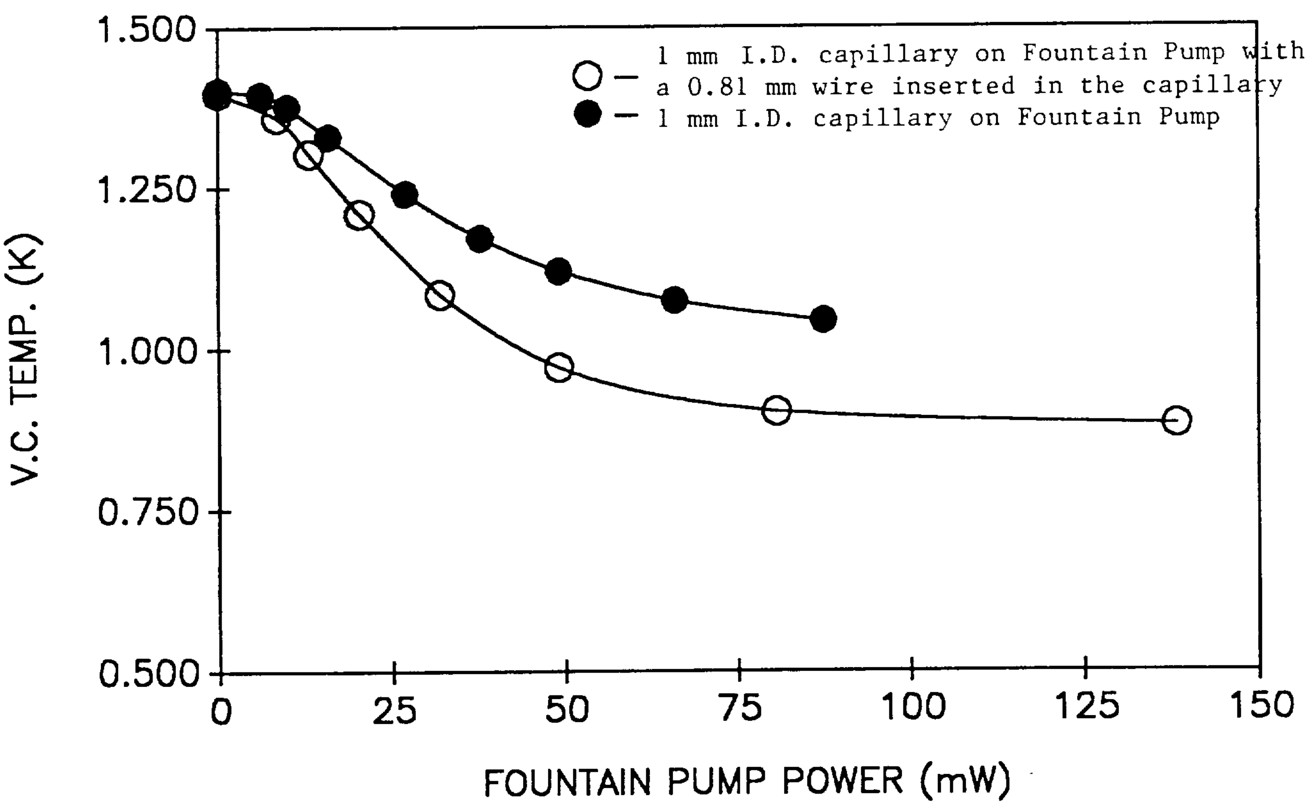

Figure 16 Vortex Cooler Temperature as a function of Fountain Pump power for two different Fountain Pump Capillary Configurations

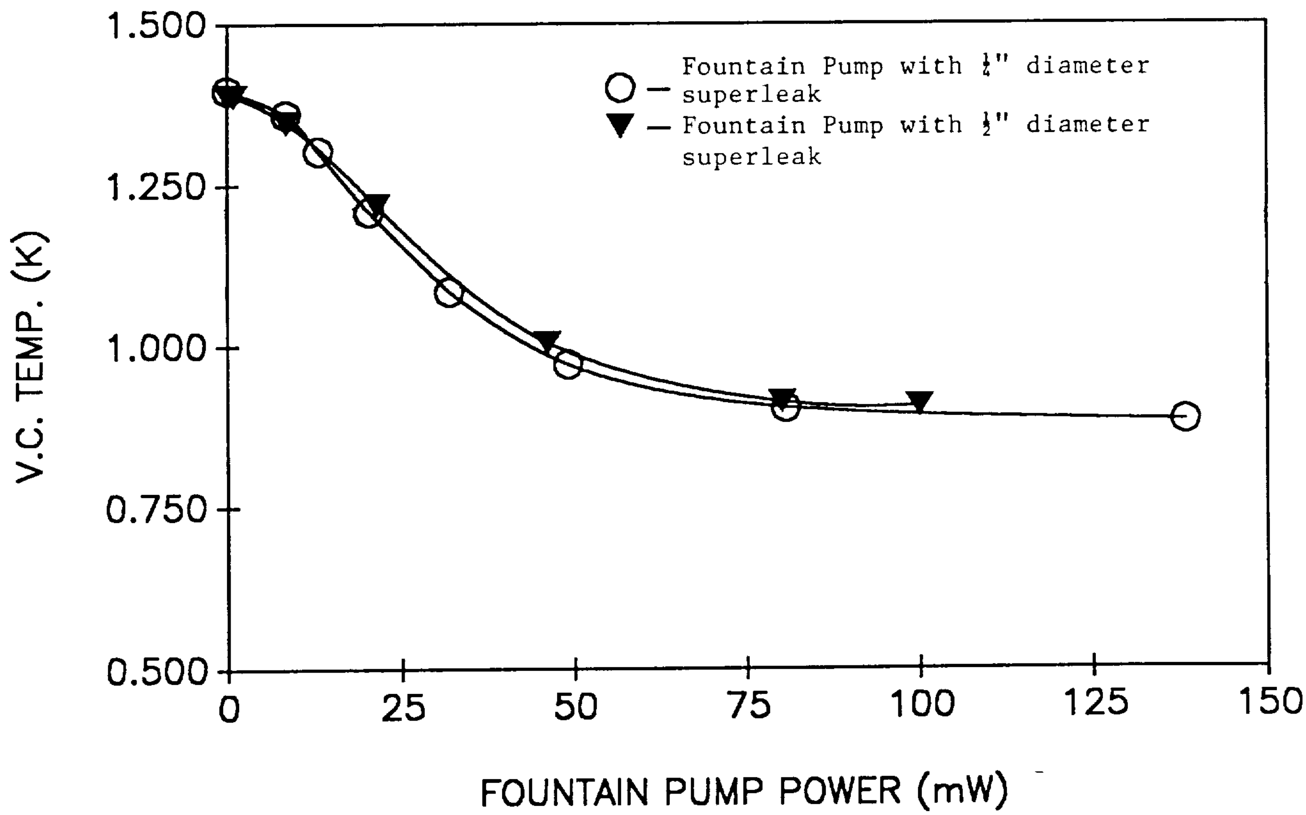

Figure 17 Vortex Cooler temperature as a function of Fountain Pump power for two different Fountain Pump Superleak diameters.

superleak. Changing the diameter of the fountain pump superleak from 1/4" to 1/2" seems to have had little effect on the efficiency of the cooling system.

In conjunction with other projects, ACE, Inc. has done extensive research on the performance effects of changing the length and width of vortex cooler superleaks. This work was performed for NASA in a Phase II experimental study for Marshall Space Flight Center. The work was entitled "Long Lifetime, Spaceborne, Closed Cycle Cryocooler" (Contract No. NAS8-35254). It was discovered that the length to diameter ratio (L/D) of the vortex cooler was important to performance. The optimal value determined from the study was L/D ≃ 10. For very large values (L/D = 160) poorer performance was observed and an anomalous effect was seen that would cause spontaneous disruptions of the liquid flow through the vortex cooler. This was thought to be due to localized dissipative heating in the superleak, which drove segments of the superleak above T_λ and thus disrupted the superfluid flow.

Design of the Finalized ACE, Inc. Vortex Cooler/Fountain Pump System

In this section, a detailed description of the ACE, Inc. vortex cooler used in the finalized version of the ADR apparatus will be give. Also, designs of the fountain pump and the coiled heat exchanger used with this vortex cooler will be shown.

Vortex Cooler

In Figure 18, a 1:1 scale drawing of the finalized ACE, Inc. vortex cooler is shown. This figure is a cross-sectional cutaway representation; all of the parts used to assemble the vortex cooler have cylindrical symmetry.

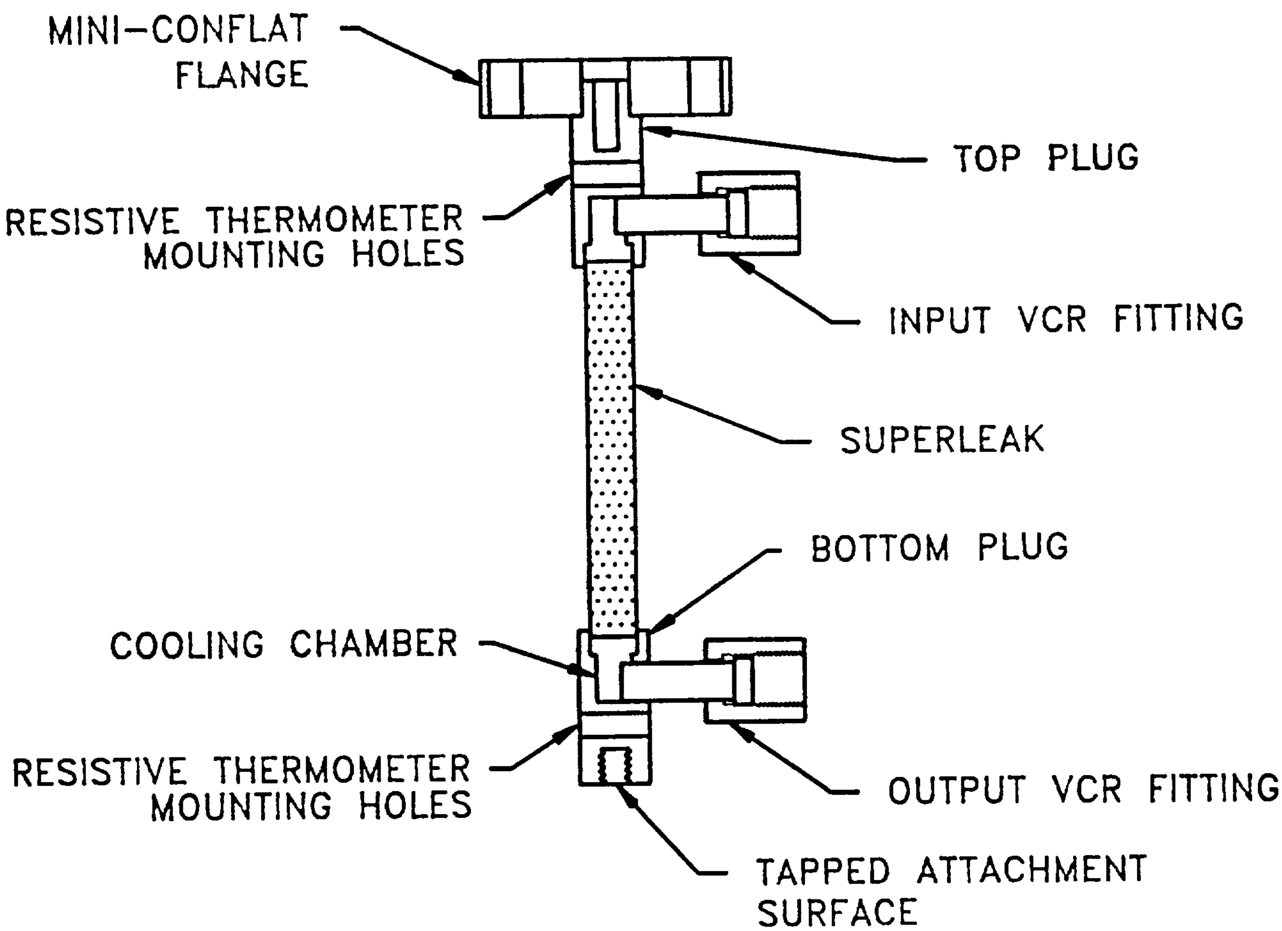

Figure 18 Scale Drawing of Finalized ACE, Inc. Vortex Cooler Design

Both the top and bottom plugs were made from OFE Copper to insure low thermal gradients across them. VCR detachable fittings were installed at the fluid inlet and outlet to allow easy replacement of capillaries, vortex coolers, or heat exchangers. The superleak consisted of a 1/4" diameter stainless steel tube with 0.010" wall thickness. This tube was 2" long and packed with jeweler's rouge (iron oxide) powder. The vortex cooler top plug was attached to a mini-conflat flange that fastened to the underside of the ^{4}He pot. Superfluid helium could pass through the flange into a dead-end chamber in the top plug to assure that the top plug is adequately heat sunk to the ^{4}He pot. A tapped hole in the bottom of the bottom plug allowed screw attachment of heaters or other devices to the bottom of the vortex cooler.

In Figure 19, an actual sized drawing of the fountain pump is shown. It is structurally very similar to the vortex cooler previously described. the only difference is that the top plug and mini-conflat flange are completely drill through, allowing superfluid to flow from the ^{4}He pot directly into the fountain pump superleak. A heater to run the pump was screwed to the fountain pump bottom.

Figure 20 shows a cross-sectional view of the ^{4}He pot coiled heat exchanger. Fluid from the fountain pump is admitted to the coils in the exchanger, which are bathed in superfluid from the ^{4}He pot. After traveling through these coils, the fluid is cooled from the temperature at the exit of the fountain pump back down to the ^{4}He pot temperature. The heat exchanger coil consists of a 4 foot length of 1/8" diameter copper tubing. Although they are not shown in Figure 20, 1/8" VCR fittings are used for connection to

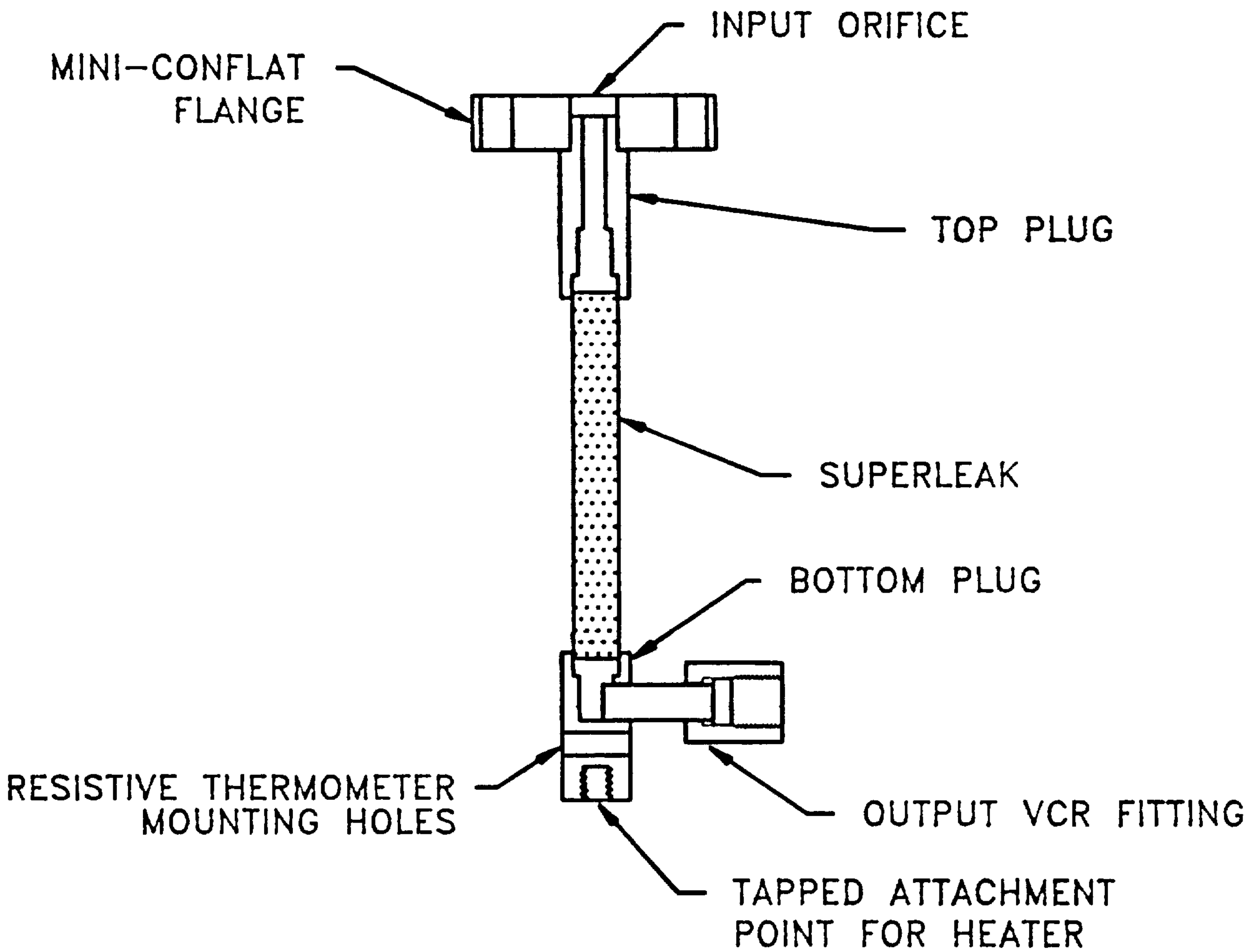

Figure 19 Scale Drawing of Finalized ACE, Inc. Fountain Pump

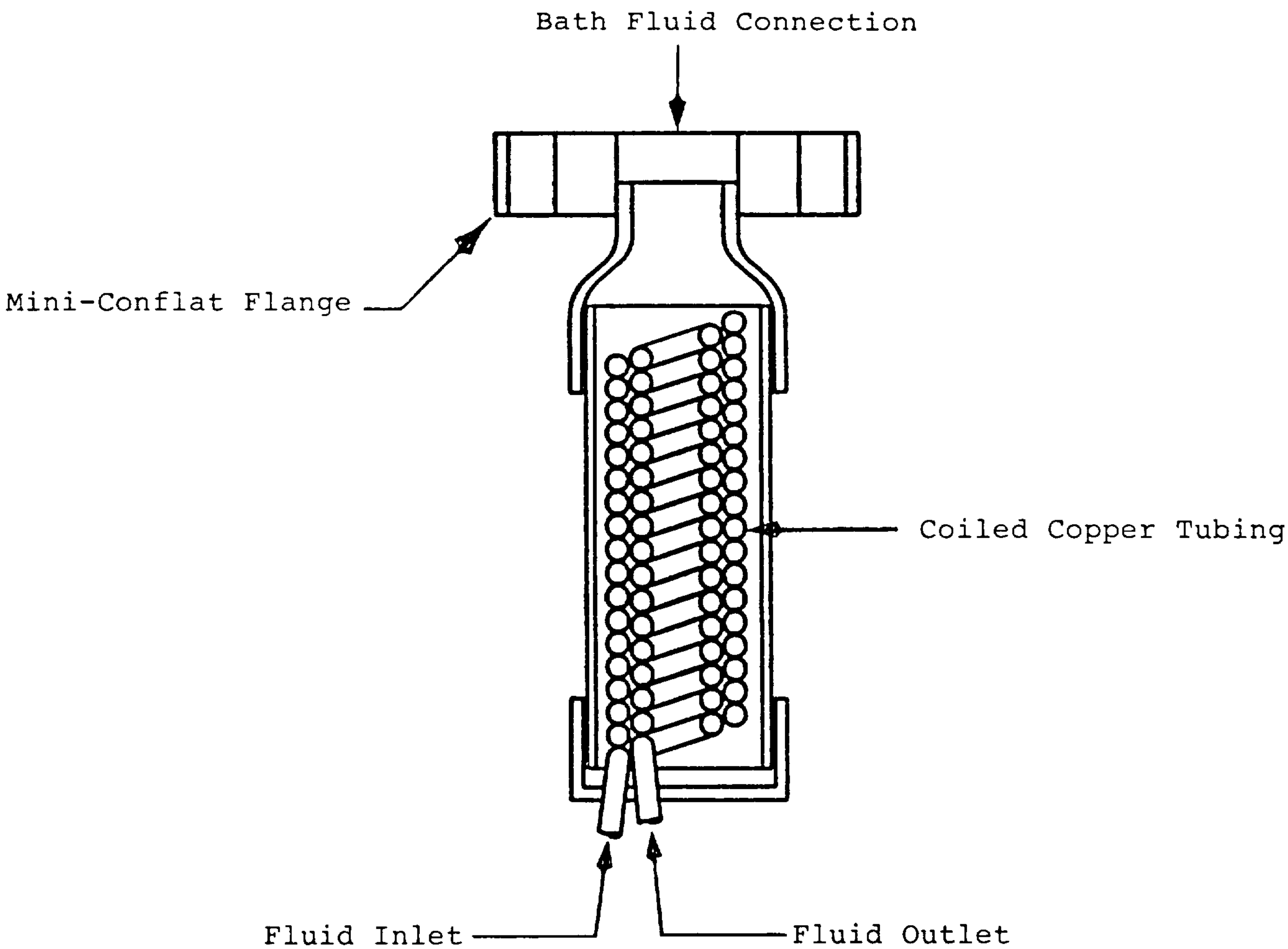

Figure 20 4He Pot Coiled Heat Exchanger the vortex cooler and fountain pump capillaries. In Figure 21 a schematic view of the vortex cooler/fountain pump/heat exchanger system is shown.

Performance of the Vortex Cooler/Fountain Pump System

Approximately twenty (20) different vortex cooler/fountain pump configurations were tested by ACE, Inc. in conjunction with the development program. Figure 22 shows performance curves for fifteen of these configurations that were tested on the auxiliary test facility. Increases in the efficiency and performance of devices can be seen from this graph.

Measurement of the cooling power of vortex coolers were obtained on the auxiliary test facility. A vortex cooler very similar to the one shown in Figure 19 obtained a minimum temperature of 0.78 K and achieved a cooling power of approximately 1 mW at 1.1 K. This result was achieved for a fountain pump input power of approximately 35 mW and with a 50 cm long 0.5 mm I.D. capillary. It should be noted that the vortex cooler responds very quickly when current is supplied to the fountain pump. Approximately 6 seconds are required for cooldown equilibrium when the vortex cooler is not under load. These results are summarized in Table 4. The finalized ACE, Inc. vortex met the required temperature minimum listed in the baseline specifications; however, its cooling power is approximately 1/5 of the baseline specification of 4.95 mW at 1.1 K. Five such vortex coolers can be placed in parallel to achieve the desired result. Another aspect of the vortex cooler performance that could be improved is efficiency. It has been suggested by Frederking et al. (1987) that much of the inefficiency in vortex cooler/fountain pump systems is due to the inefficiency of the fountain pump. He suggests using

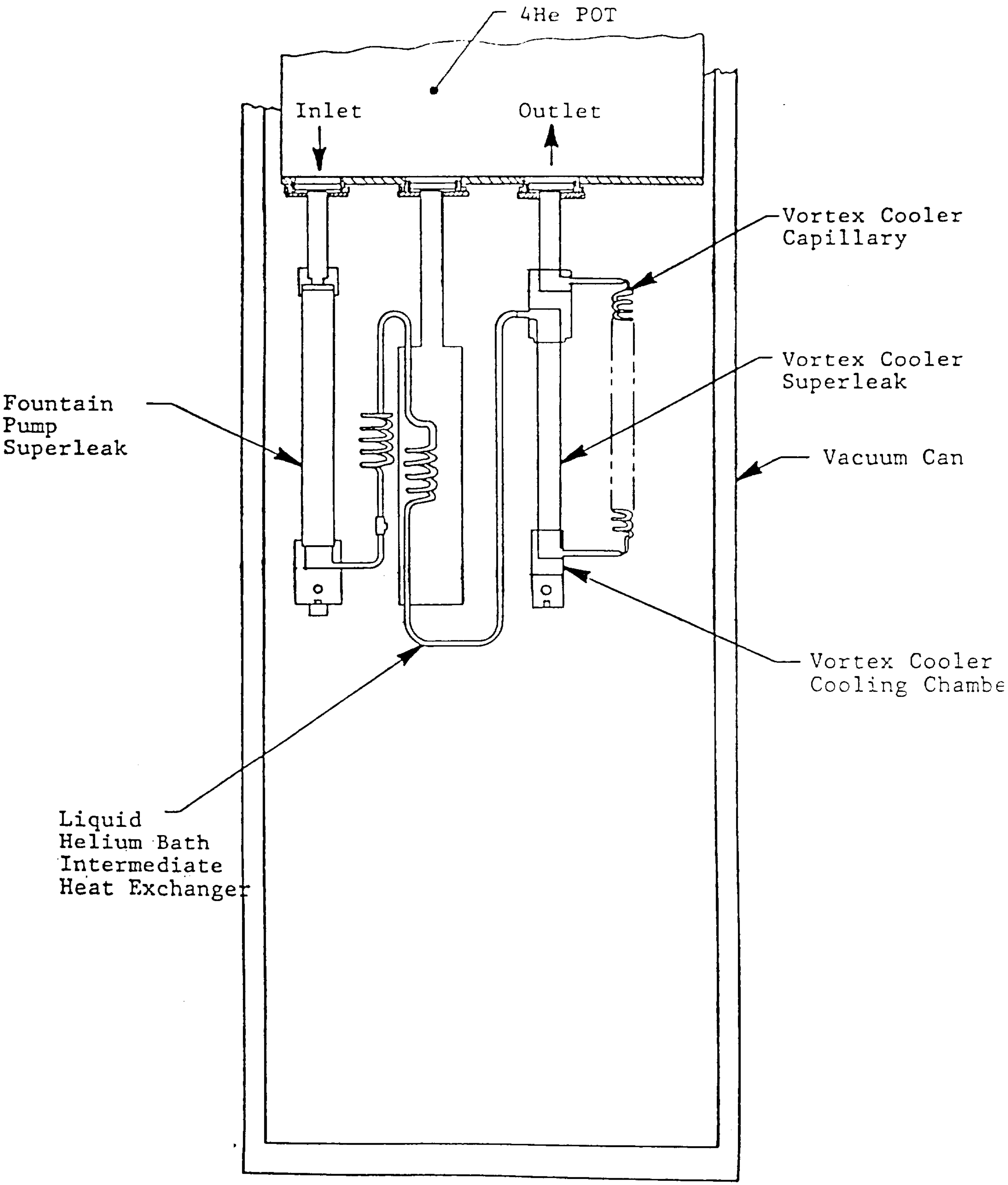

Figure 21 Fountain Pump/Vortex Cooler Configuration

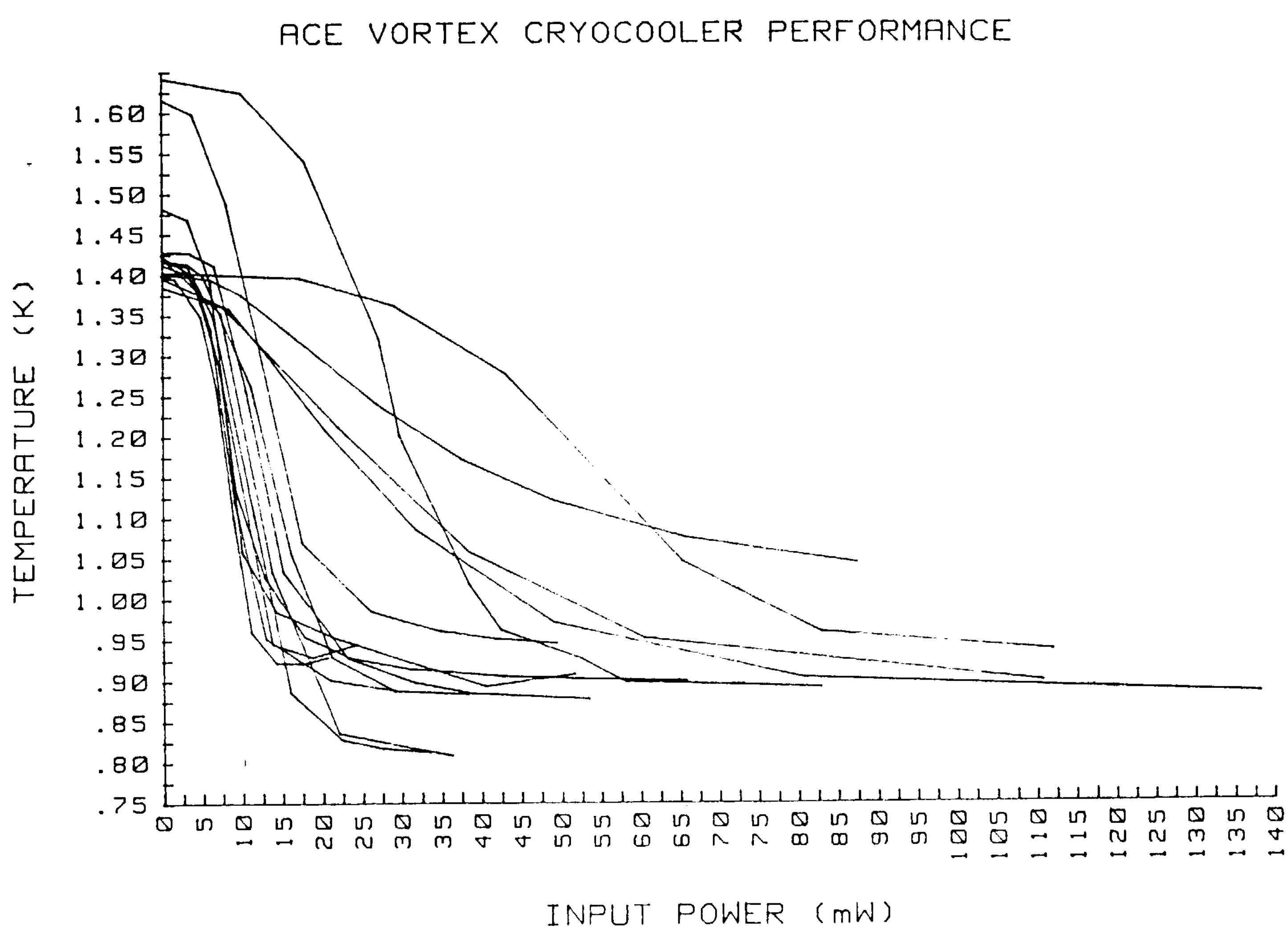

Figure 22 Performance Curves for Different Vortex Cooler Test Configurations

multi-stage series fountain pumps to improve this. The idea is based on the fact that the fountain pump approaches Carnot efficiency for sufficiently small ΔT values.

TABLE 4 ACE, INC. VORTEX COOLER PERFORMANCE

Minimum Temperature:	0.78 K
Required Fountain Pump Power:	35 mW
Cooling Power:	1.0 mW at 1.1 K
Cooldown Time:	$\approx$ 6 sec.

V. ADIABATIC DEMAGNETIZATION REFRIGERATOR

In this section, a summary of the development of the ACE, Inc. vortex precooled Adiabatic Demagnetization Refrigerator will be given, along with the design of the finalized ADR system. Results from successful tests of the device will be shown. Also, a detailed description of the salt pill and heat switch used in the device will be presented.

Basic Principle of Operation

The basic principle of the ACE, Inc. ADR is illustrated schematically in Figure 23. A superfluid filled capillary is used as the heat switch to alternately remove heat from the salt and then to thermally isolate the salt from its surroundings. As can be seen by the results of Bertman and Kitchens already presented in Figures 12 and 13 and discussed in Section IV, a superfluid filled capillary markedly changes its ability to conduct heat when the temperatures at the end of the capillary are changed. From Table 3, we see that a 50 cm length of 0.25 mm I.D. capillary conducts 2 milliwatts of heat if one end is at 0.1 K and the other end is at 1.3 K. However, if the hot end temperature is lowered to 0.7 K, the same capillary conducts only 0.3 microwatts of heat.

The operation of the cycle is as follows. Initially, both the salt pill and the ^{4}He pot are in equilibrium at (1.4 - 1.8 K). Then the superconducting magnet is energized and the field is brought up to its maximum value. Under these conditions, the capillary is a very good conductor of heat, and the heat of magnetization is then removed. At this time, the salt pill and the ^{4}He pot

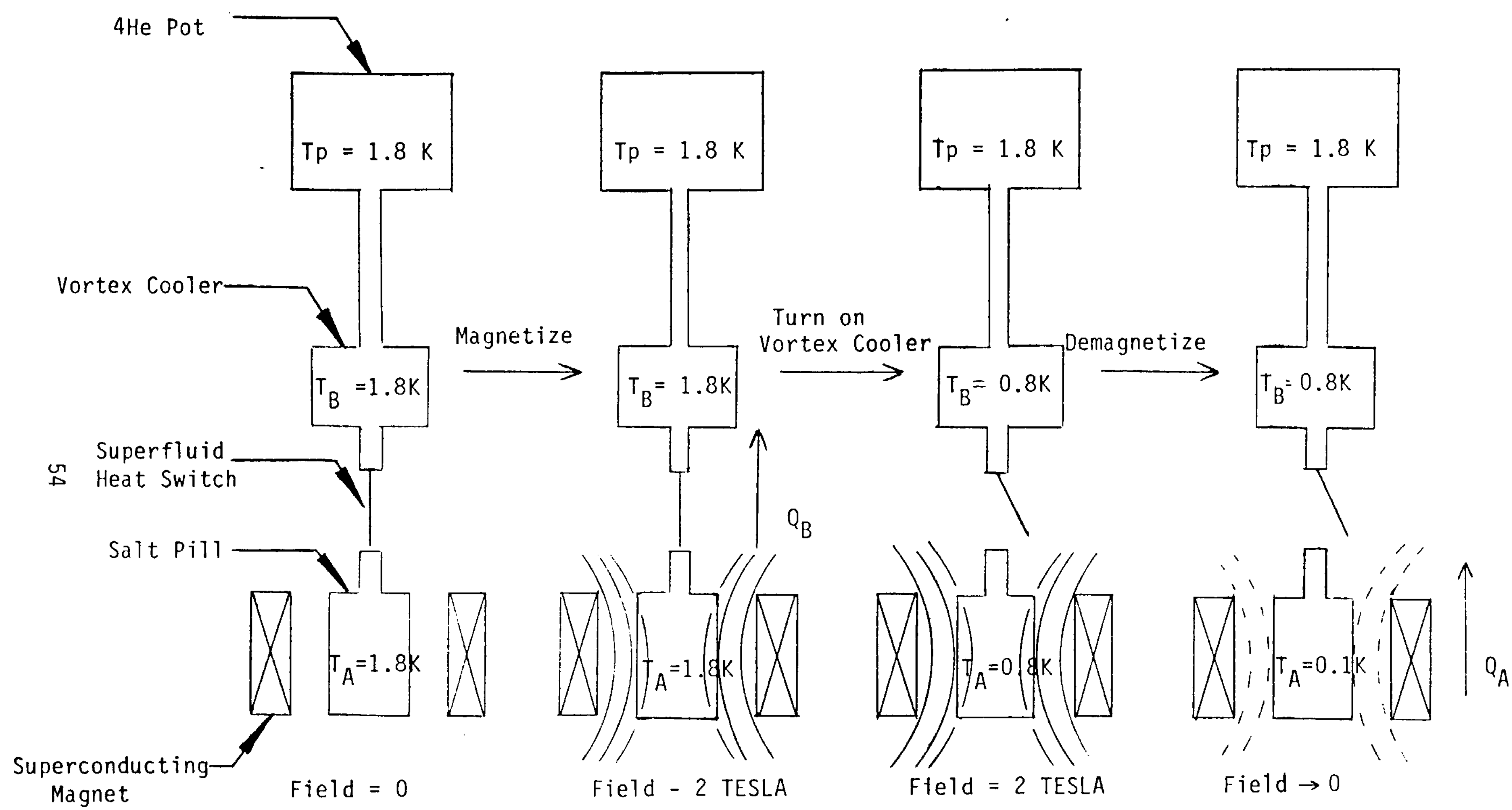

Figure 23 Schematic Representation of ADR Cycle

are again in equilibrium at the pot temperature. The vortex cooler is then turned on, which has the effect of anchoring the upper warm end of the portion of the capillary leading down to the salt pill at the vortex cooler temperature of 0.7 - 0.8 K. When the vortex cooler is turned on, the salt will decrease in temperature to approximately 1 K fairly rapidly, as the capillary is still a good conductor of heat in that range. Thus, the salt is also precooled by the vortex cooler. At this point, the magnetic field is quickly reduced. Entropy increases in the salt pill, and rapid cooling occurs. If a controlled temperature is needed, the magnetic field is initially reduced only enough to reach the desired temperature, and then slowly reduced using a feedback loop temperature controller. After the field has reached zero and the salt begins to warm up, the process may be repeated.

Preliminary Testing

Since the predictions for the heat conduction in superfluid filled capillaries given by Bertmann and Kitchens are partially heoretical in nature, it was desirable to test the operating principle of the superfluid filled capillary heat switch. To do this, the ^{3}He refrigerator of the auxiliary test device was used as shown in Figure 24. The ^{3}He pot was used to hold the lower end of the capillary at a constant 0.65 K, while the upper end temperature was varied by action of the vortex cooler. The heat flow into the ^{3}He pot was determined by measuring the mass flow of ^{3}He leaving the pot. Using the known heat of vaporization of ^{3}He and subtracting out the background heat leak gave the data shown in Figure 25. Here, the heat flow down the capillary is plotted as a function of the warm end temperature. The theory of Bertmann and Kitchens is shown as a solid line. The theoretical heat conduction can be

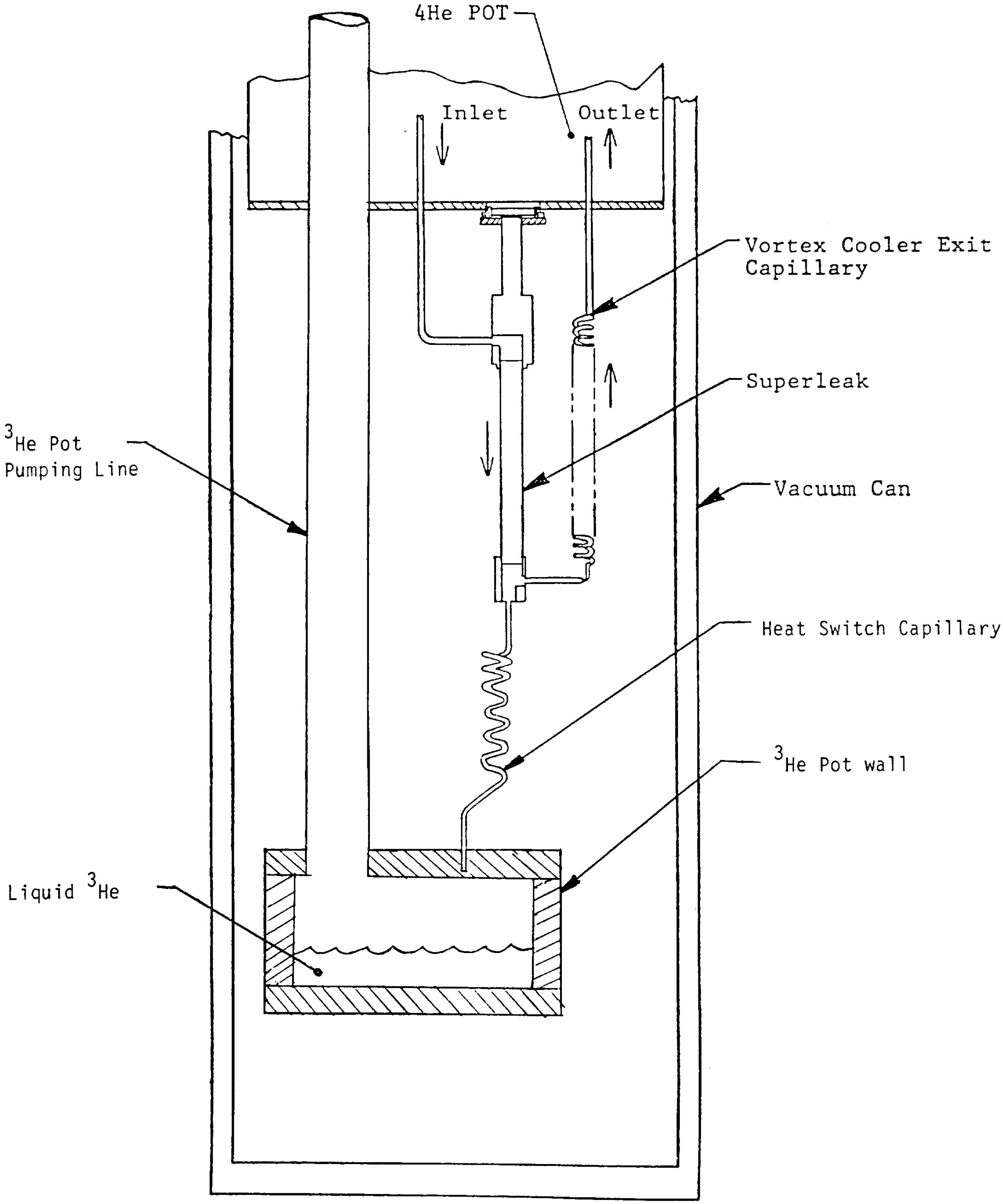

Figure 24 Auxillary Test Facility for evaluating the Conduction of Superfluid filled Capillaries. The Fountain Pump and Heat Exchanger are omitted from the drawing for clarity.

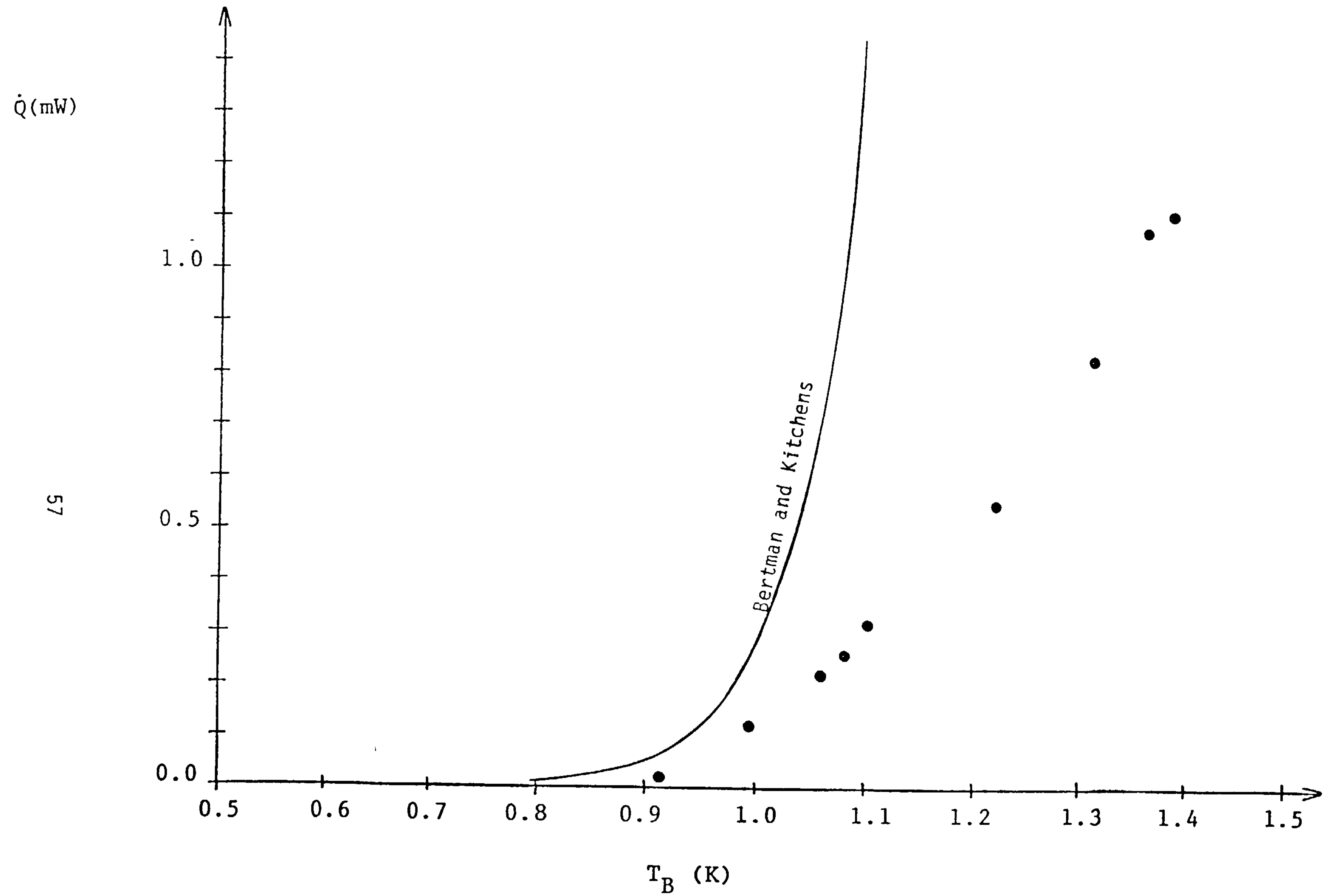

Figure 25 Heat conduction of a 0.5 mm I.D., 30 cm long superfluid filled capillary as a function of its warm end temperature T_B. The other end of the capillary is held constant at 0.65 K. The solid line represents the theory of Bertman and Kitchens (1968). The data is shown as solid circles.

seen to rise more steeply than the data that was obtained. We attribute this discrepancy to be due to Kapitza resistance effects at the lower end of the capillary, where it dead ends into the ^{3}He pot platform. From these tests, it was concluded that a superfluid heat switch was a viable technology.

Design of the ACE, Inc. Vortex Precooled ADR

In Figure 26, a schematic drawing of the design of the ACE, Inc. Adiabatic Demagnetization Refrigerator is presented. The finalized fountain pump/vortex cooler module described in Section IV is used as the precooler. A hole was drilled into the cooling chamber of the vortex cooler, and a 0.3 mm I.D. stainless steel capillary 50 cm long extended down to the salt pill. The other end of the chamber, which was filled with liquid when the capillary was. This canister had a volume of 0.31 cm^3 and an interior surface area of 2.55 cm^2. The chamber base was partially tapped so it could be attached with screws to the salt pill. The purpose of the canister was to allow for a bigger liquid-to-copper surface area, in order to avoid the resistive effects manifested in Figure 25. According to White (1979), the copper-to-superfluid Kapitza resistance is approximately given by

$$Q/A\Delta T = 200\ T^{3.4}\ \ W/m^2K \tag{5}$$

This yields a temperature difference at the canister of approximately 49 millikelvin per microwatt of heat transmitted through the capillary at 0.1 K.

The salt pill is shown schematically in Figure 27. It was mounted so as to be at the center of the magnet. The salt pill consisted of a 0.016"

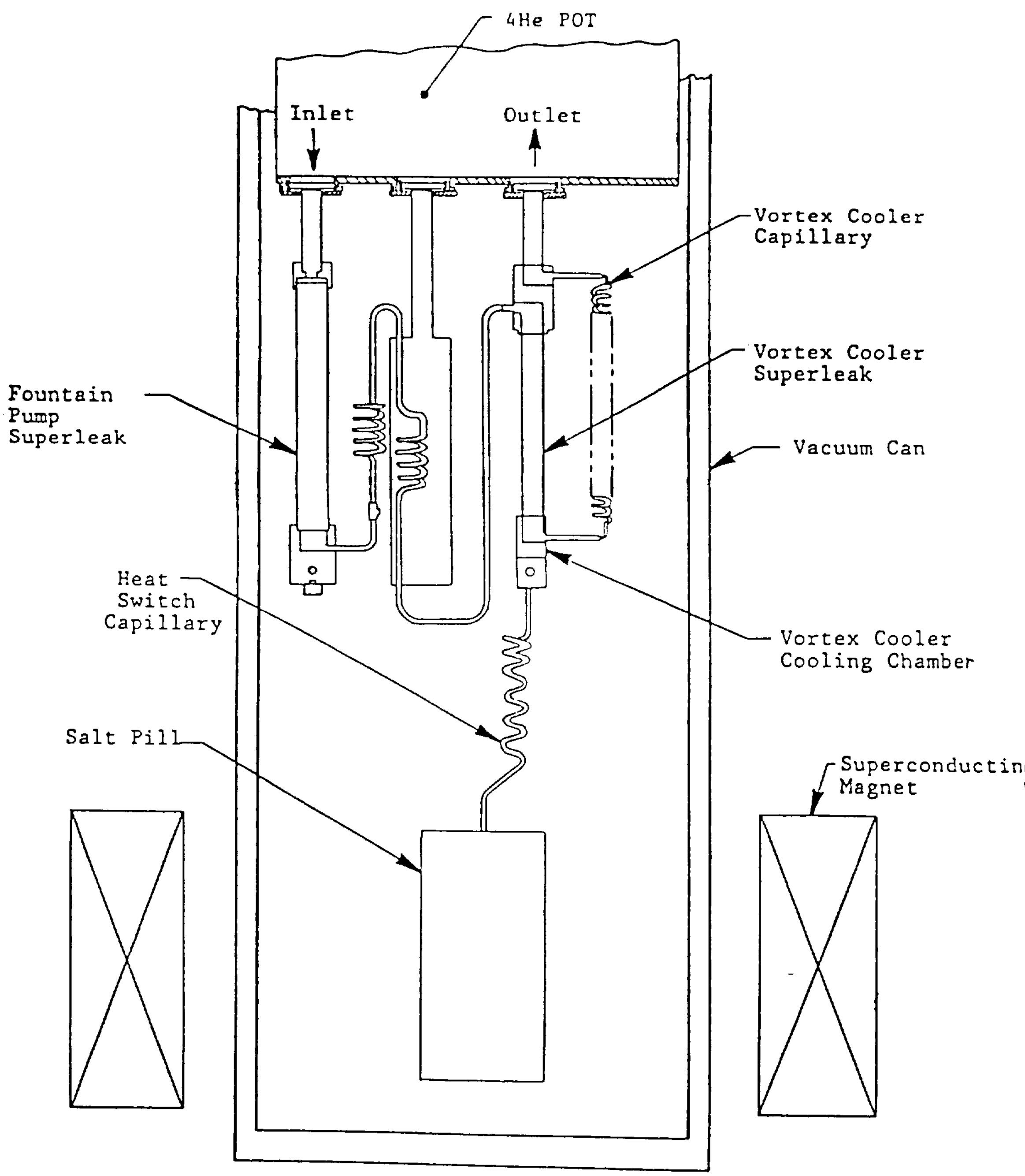

Figure 26 ADR Configuration

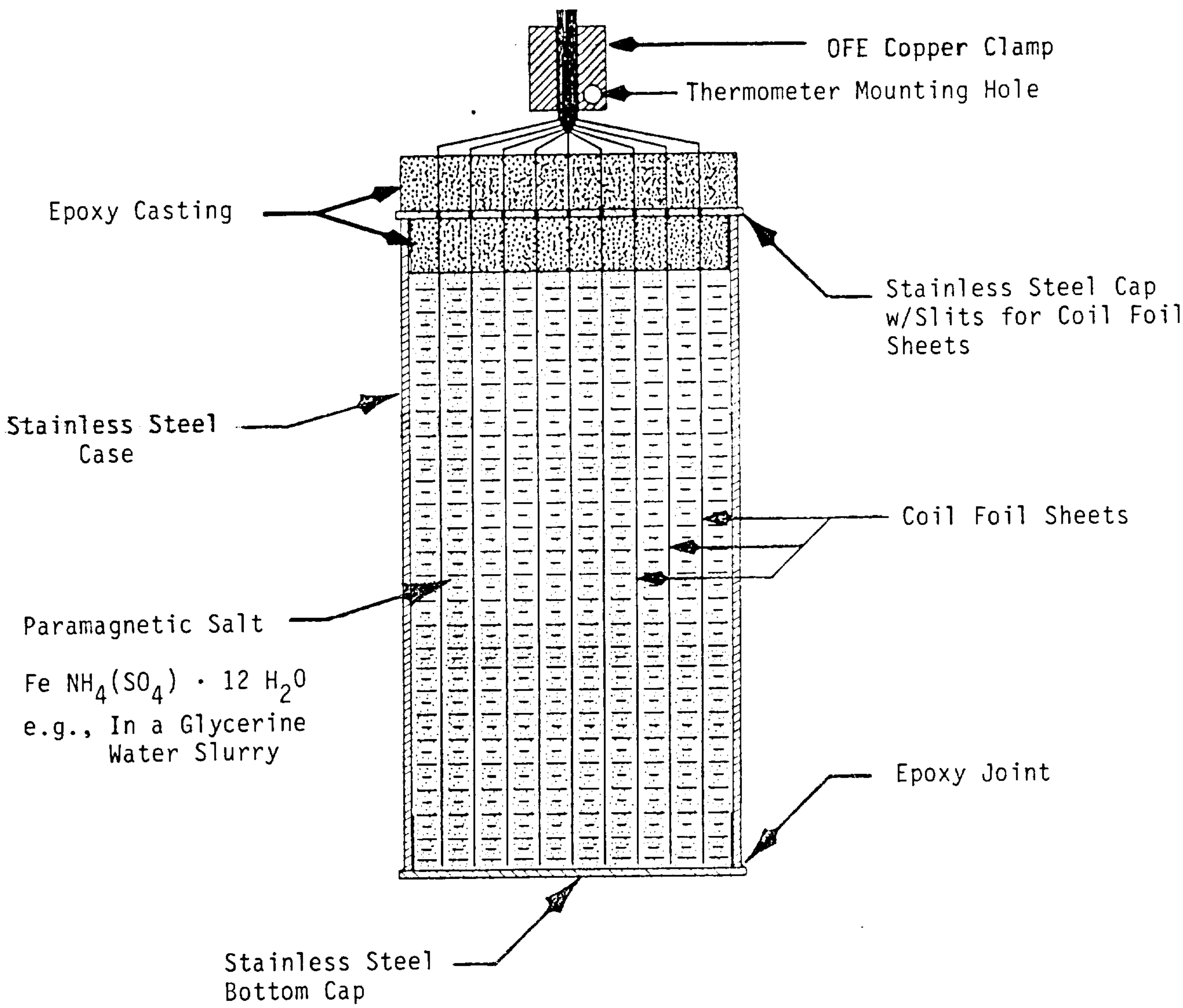

Figure 27 Schematic Representation of ADR Salt Pill

diameter thin walled stainless steel cylinder 3" long. Sheets of coil foil made from 0.10" diameter copper wire and held together with Ge 7031 varnish were fitted into a stainless steel top cap, and cast into place with Stycast 2850 GT epoxy. The total surface area of the coil foil was 490 cm^2; each sheet was spaced 1/8" apart. The top cap was then epoxied to the stainless steel cylinder. From the bottom side, the salt pill was packed with the salt ferric ammonium sulfate $(FeNH_4(SO_4)\cdot 12H_2O$ in a glycerin and water slurry. The slurry consisted of 65.1 gms of the salt in 25 gms of water and 94.6 gms of glycerin. The slurry technique is standard, and has been described by Kurti et al. (1956). It should be noted that the quantity of salt used in our test salt pill in approximately 37% of the amount used in the baseline design of Castles. Our test results can be scaled appropriately as required. After filling the salt pill with slurry, the bottom cap was epoxied in place. Thus, a leak-tight enclosure for the salt pill was formed. The coil foil sheets were coated with vacuum grease and secured by a copper clamp at the top of the salt pill. This clamp had a hole for mounting the germanium thermometer. The clamp had a hole in it that allowed the helium filled canister at the end of the heat switch capillary to be attached. The salt pill was supported by a network of 0.010" diameter nylon threads. The threads were secured to a cage consisting of four 6" long stainless steel tubes that passed through 2 nylon discs 2 5/8" in diameter and 3/32" thick. The cage was supported by the ^4He pot. The heat conduction from the pot through the cage and threads to the salt pill was less than 1 microwatt.

Two Vortex Cooler Model ADR

In the Phase II proposal, a method of using two vortex coolers as a heat switch for the ADR was proposed. The concept involved here was that a second vortex cooler would be placed directly in contact with the salt pill, while the first vortex cooler would be used to intercept the heat leak along the capillary of the second one. Such as arrangement is illustrated schematically in Figure 28. To avoid cluttering the picture, the two fountain pumps and two coiled ^{4}He pot heat exchanger required to drive these vortex coolers are not shown. The idea behind this arrangement was that the first vortex cooler could act as a heat switch, as before. The second vortex cooler could act to directly remove heat from the salt pill, and hasten ADR recycle time. It was assumed that a proper input pressure to both vortex cooler (1) and (2) could be found so that vortex cooler (1) would remain at approximately 0.7 K, and vortex (2) would experience no flow. Since the superleak of vortex cooler (2) should prevent entropy transfer, it was thought that the cooling chamber of the second vortex cooler would be thermally isolated. However, it was discovered after building testing a vortex cooler arrangement of this type that the problem of backflow through vortex cooler (2) could not be controlled. The ^{4}He pot caused vortex (2) to act like a fountain pump. This effect put a large thermal load on vortex (1) and made the concept impractical.

It was desirable to try this scheme with vortex coolers driven by concentric tube heat exchangers in which the flow could be valved off at room temperature, but difficulties associated with the use of these heat exchangers

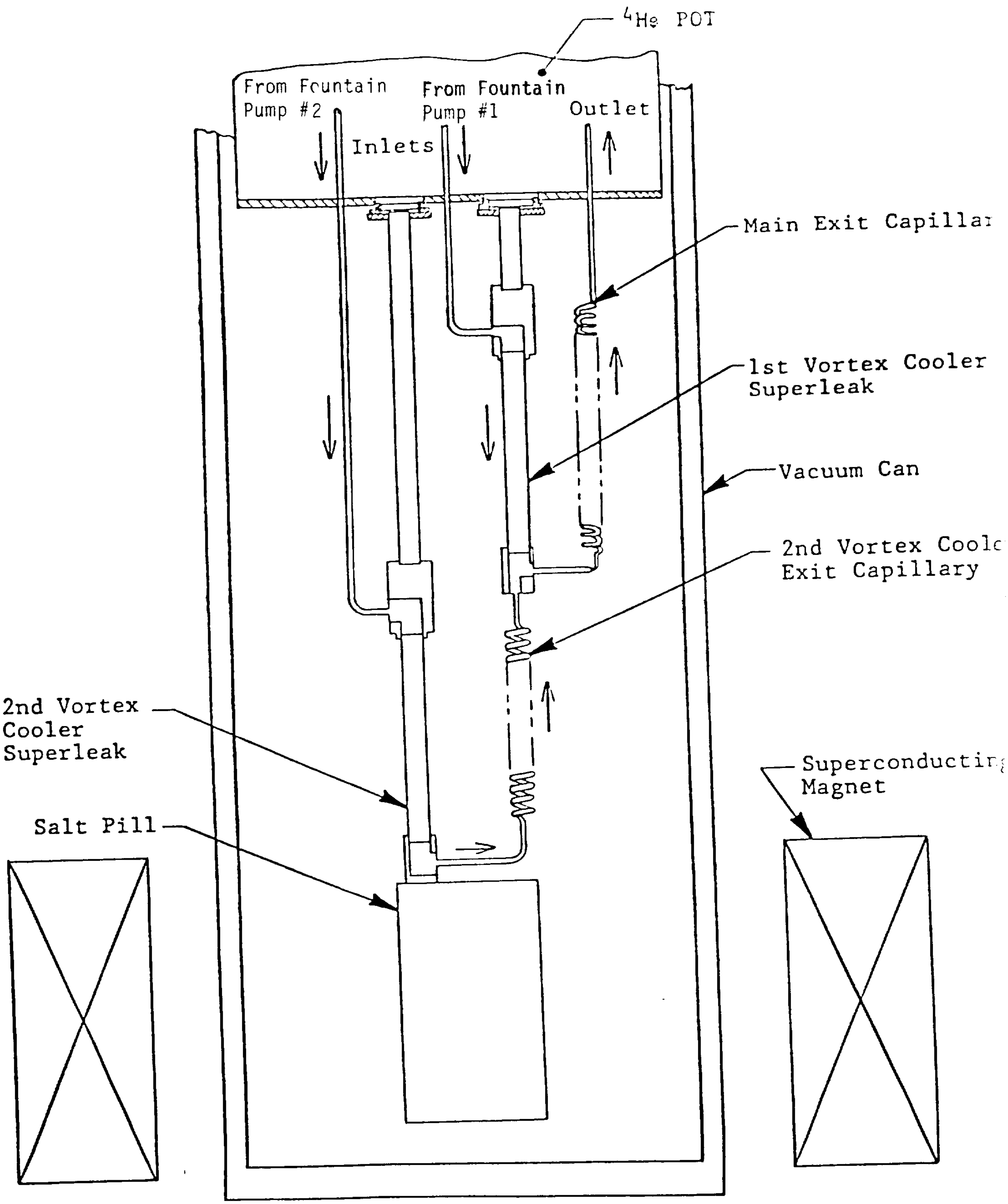

Figure 28 ADR Configuration Using Two Vortex Coolers

prevented this testing from taking place. Therefore, the one vortex cooler model was selected as the finalized design.

Results

In Figure 29, the results of one of the ACE, Inc. vortex precooled ADR test runs are shown. The plot shows the temperature of the salt pill as a function of the total elapsed time. Initially, the salt pill, vortex cooler, and ^{4}He pot were in equilibrium at approximately 0.8 K at t = 0, the magnetic field began to increase. An immediate heating effect was seen in the salt pill. After approximately 11 minutes, the magnetic field reached its maximum value of 2 Tesla. This field was roughly half that used in the Castles baseline; however, as we will see later, precooling the salt to 1.1 K from 1.8 K reduces the required field by a factor of two.

After the field reaches a maximum value, an additional waiting period of 48 minutes was required until the salt pill returned to the ^{4}He pot temperature. The length of this waiting period depended on the heat conduction of the liquid in the vortex cooler exit capillary. If several vortex coolers were placed in parallel to achieve the required baseline cooling power, a substantial increase in capillary heat conduction and a much lower waiting period here would have been expected.

When thermal equilibrium was reached between the salt pill and the ^{4}He bath, the vortex cooler was turned on. Then, the salt pill slowly began the cool toward the minimum temperature of the vortex cooler. Without the presence of the magnetic field, this is a fast process; however, since the

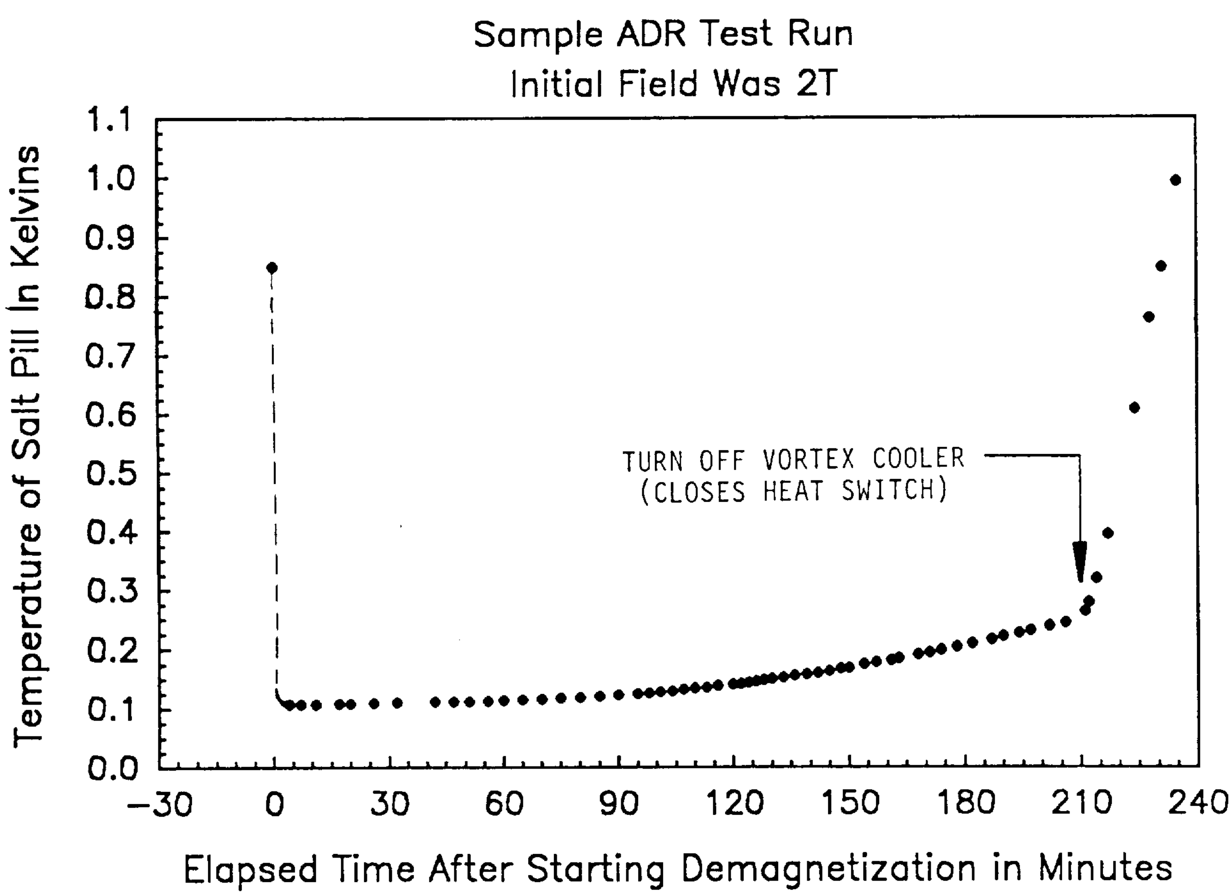

Figure 29 Cooling curve for the ACE, Inc. Adiabatic Demagnetization Refrigerator

field was on, additional ordering and decrease of entropy in the salt caused heat to be expelled that had to be dissipated by the vortex cooler. This heat dissipation took about 141 minutes for the run described here. The system was allowed to come to equilibrium at approximately 0.85 K in this run which is significantly lower than the 1.1 K target temperature. The thermal conduction in the heat switch capillary became very small below approximately 1 K, causing a long equilibrium time. It was decided in this run to wait for equilibrium rather than guess at the temperature; the germanium thermometers did not provide accurate measurements when the field was on. Also, having several coolers in parallel would have increased cooling power and could have served to substantially reduce the time required for this portion of the cycle.

After equilibrium was reached at approximately 0.85 K, demagnetization was begun. Since no temperature controller was present on this apparatus, the field was reduced all the way to zero. Turning off the field took a period of approximately 40 seconds. Immediate cooling in the salt pill was seen upon lowering the field. As can be seen from the graph, the refrigerator reached a minimum temperature of 0.107 K and stayed below 0.125 K for a period of 90 minutes.

One of the external heat loads on the salt pill during the demagnetization run was due to thermal conduction of the manganin leads that lead down to the germanium and film resistors. A total of eight manganin leads 0.0035 inches in diameter and 6 1/4 nches long were heat sunk at both the ^{4}He pot temperature of 1.4 K and the vortex cooler temperature of 0.8 K. According to Lounasmaa (1974) the thermal conductivity of manganin is 0.005

W/m·K at 0.1 K and 0.04 W/m·K at 0.8 K (page 246). Let us assume an average value of 0.022 W/m·K. For our eight leads, only 5×10^{-3} microwatts of heat was conducted down to the salt pill.

Eddy current heating was calculated for the copper coil foil wires in the salt pill and for the copper clamp at the end of the wires. These were the only copper present in the high field region. Following the treatment of Castles (1980), the eddy current heating per unit volume in a cylindrical conductor is given by

$$Q/v = (r^2 B_0^2/\rho\ t) \times 4.2 \times 10^{-13}\ \mathrm{erg/cm^3} \tag{6}$$

where r = radius of cylinder [cm]

ρ = electrical resistivity of the conductor

t = duration of the magnetization.

B_0 = Magnetic Field (K gauss)

Thus, a worst case estimate of 4 microwatts in a 5 minute demagnetization has been obtained for eddy current heating. Radiation leaks from 4.2 K to the salt pill have also been calculated using

$$Q = \varepsilon\sigma A\ (T_H^4 - T_C^4)]/(2-\varepsilon) \tag{7}$$

where Q = heat radiated from hot body to cold body

$\sigma = 5.67 \times 10^{-12}\ \mathrm{W/cm^2\ K^4}$

ε = emissivity of hot body.

Let us assume $\varepsilon \simeq 1$ to get an upper limit on the heat leak. Taking $T_A = 4.2$ K and $T_C = 0.1$ K, with $A \simeq 876$ cm^2 gives a worst case heat leak due to radiation of 1.8 μW.

VI. MAGNETIC SHIELD SYSTEM

In this section, the design of the ACE, Inc. soft iron magnetic shield is presented. Values for the radial magnetic field of the ADR superconducting solenoid have been measured in both the shielded and unshielded configurations. These results are presented and are compared with the predictions of the ACE magnetic shielding computer program SHIELDIN. The design of a superfluid helium cooled superconducting magnet will be discussed.

ACE, Inc. Soft Iron Magnetic Shield

The ACE, Inc. soft iron magnetic shield is designed to surround the superconducting solenoid, as was shown in Figure 6. Sketches of the top and bottom and cylindrical body of the shield are shown in Figures 30 and 31. The shield was fabricated according to ACE, Inc. specifications by Ad-Vance Magnetics, Inc. in Rochester, Indiana. The shield was fabricated from their AD-MU-00 material, a low permeability, high saturation induction soft iron. In Figure 32, the hysterisis curve for this material is given, and some of its magnetic and physical properties are described in Table 5 The permeability, which is the slope of the B-H plot, can be seen to change as a function of the magnetic field, and approaches zero when the material is saturated.

AD-MU-00 was chosen for the shield material for two reasons: First, it has a high saturation induction; also, since it is soft iron, its magnetic properties are very weakly temperature dependent. A disadvantage to the material is that it has low permeability, but the first two considerations are much more important. Each of these considerations will now be discussed.

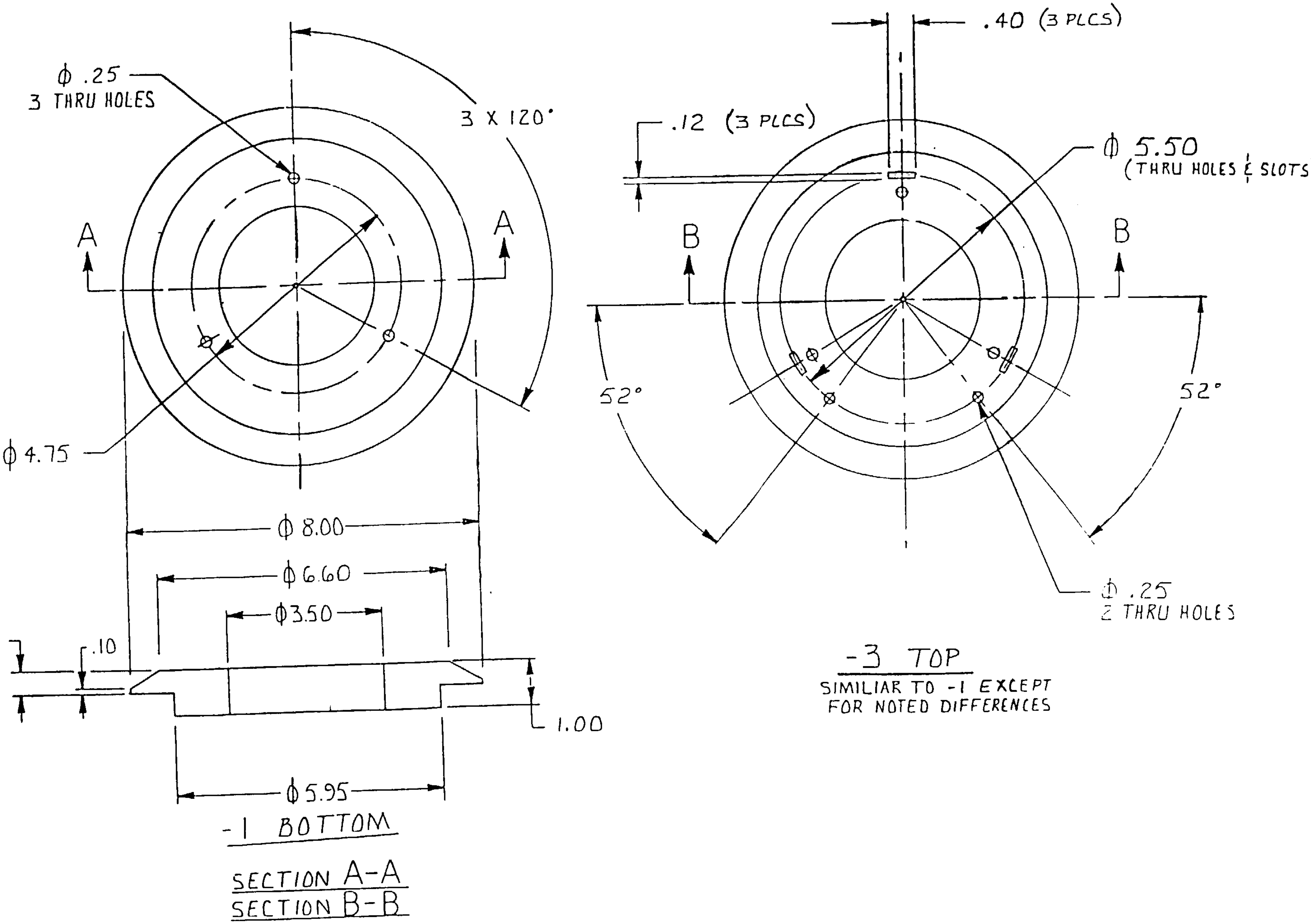

Figure 30 Top and Bottom of Magnetic Shield Assembly

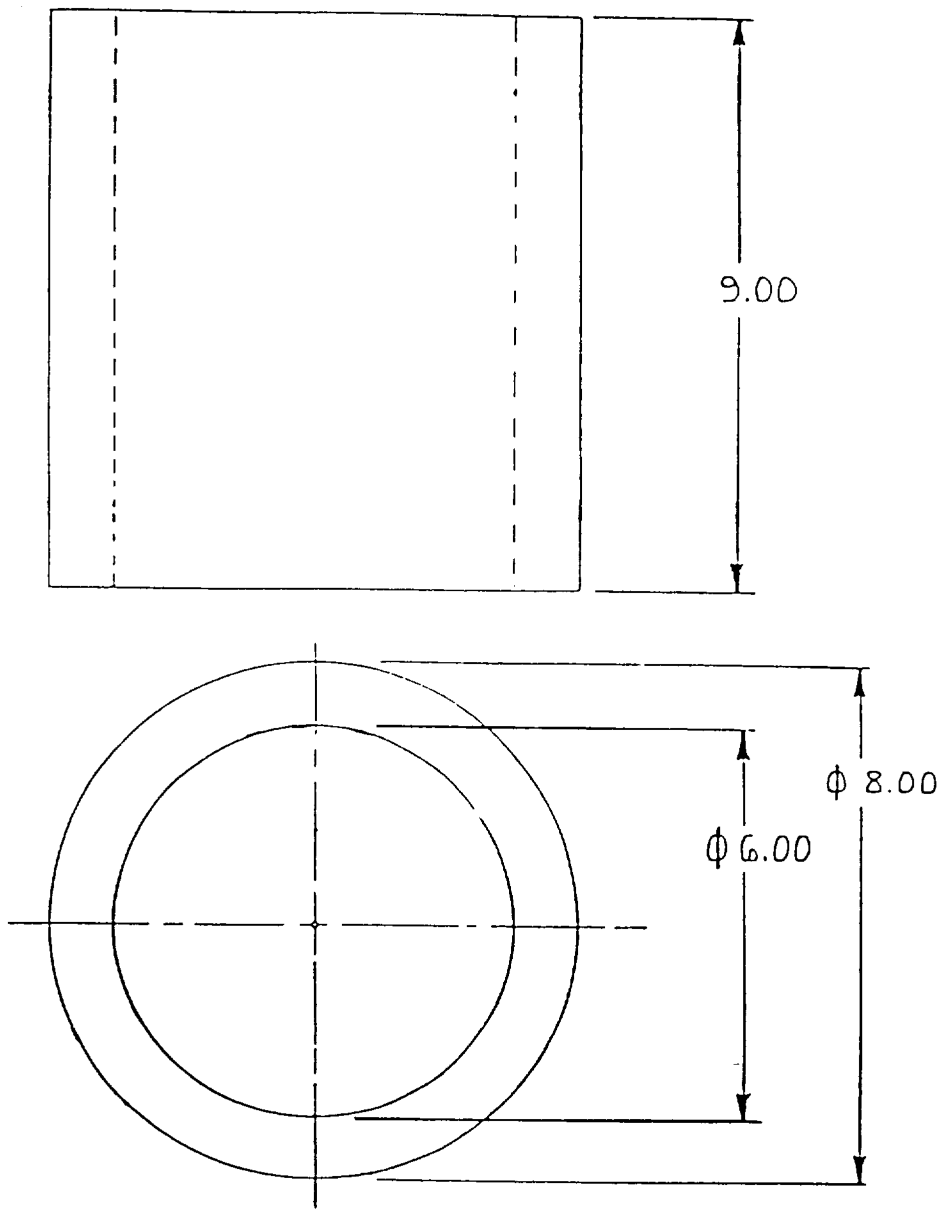

Figure 31 Side View of Magnetic Shield

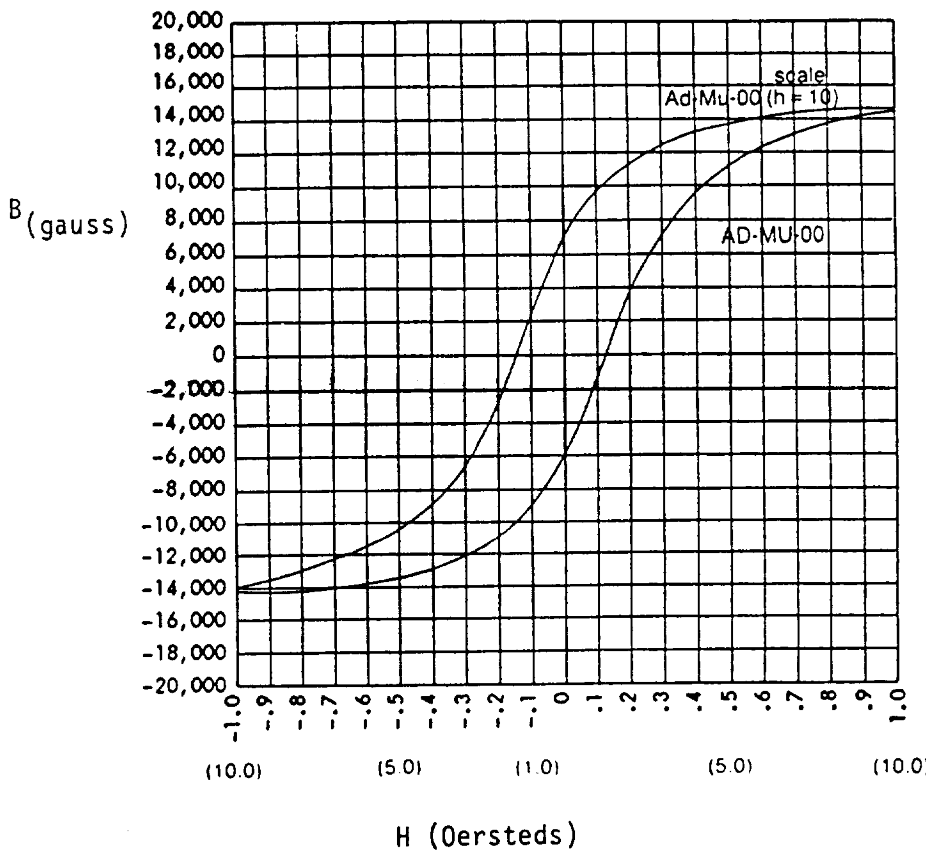

Figure 32 Magnetic flux density B vs. Magnetizing Force H for the Ad-Vance Magnetics, Inc. Alloy AD-MU-00.

TABLE 5 PROPERTIES OF THE SHIELDING MATERIAL

(Ad-Vance Magnetics AD-MU-00)

Saturation Induction	22,000 gauss
Initial Permeability	300
Permeability at 200 gauss	500
Maximum Permeability	3000
Coercive Force H_c	1.0 Oersteds
Electrical Resistivity	14 microhm/cm
Shielding Efficiency H_o/H_{in} ...	975.0
Density	7.80 gms/cm^3

Ackermann, Klawitter and Drautman (1971), show data for iron and other magnetic alloys as a function of temperature. Their work extends from room temperature (300 K) down to liquid helium temperature (4.2 K). Both pure iron (99.8% Fe) and silicon iron (2.5% Si-Fe) show less than a 10% drop in magnetic saturation and D.C. permeability, and less than a 10% rise in coercive force when decreased from room temperature to 4.2 K. This is to be contrasted with the high permeability alloy 4 Mo-Permalloy, which experiences a 90% drop in permeability under the same conditions. The permeability of the material is a measure of the effectiveness of its shielding properties. The permeability is defined as

$$\mu = B/H \tag{8}$$

where B is the flux density and H is the magnetic field strength. From the hystersis loop shown in Figure 32, it is clear that μ depends on both the flux density and the location on the hystersis loop. If the flux density becomes too large, μ approaches zero and the magnetic shielding ability of the material is lost.

In designing the shield, it was desirable to avoid completely saturating the material even for the high field tests at 4 Tesla. Since Dewar spatial constraints forced the shield to be very close to the magnet, the worst case assumption was made that all the field in the magnet would pass into the shield material. This assumption allowed a value of the shield thickness to be chosen that would insure that saturation did not occur. The 1" thick iron walls of the shield made it very heavy (approximately 70 lbs), but a much lighter shield could be used it it were not in such close proximity to the magnet.

ACE, Inc. Magnetic Shielding Program

A FORTRAN computer program called SHIELDIN has been developed by ACE employee Dr. Carl Brans. This program is designed to predict the effects of placing the soft iron magnetic shield around the superconducting solenoid. A Legendre expansion is used with different coefficients in each of three regions: (1) Interior to the shield; (2) In the shield material itself; (3) outside the shield. Matching magnet static boundary conditions at the transition between the three regions and requiring that the field vanish at infinity provides the equations that determine these coefficients. For spherical shield surfaces, these equations can be solved in closed form; however, this is not true in the general case. Non-spherical shields surfaces are evaluated by keeping a finite number of terms in the Legendre expansion and then imposing the boundary matching conditions at a discrete number of points. The program evaluates shields that have cylindrical symmetry about the z-axis and reflective symmetry about the x-y plane. Allowance is made for a hole to be in both ends of the shield, determined by an angle θ_H with respect to the z-axis. Both inner and outer shield surfaces can be input by entering ordered pairs to determine these surface, or by using a mouse to input the points as they appear on the screen. The program assumes a constant permeability μ of the magnetic material. The program uses Lahey, Inc. FORTRAN 77L, which is compatible with IBM PC computers. An 8087 coprocessor is required for the execution of the program. Also, a Hercules graphics card and Metawindow graphics routine (By Metagraphics Software Corporation) are required. A mouse and mouse driver compatible with the IBM PC is needed to enter points while viewing their location on the screen. A complete listing of the program, with documentation, is given in Appendix II. Output from an example test run of the program is also included.

Results of the Magnetic Shielding Tests

In Figure 33 the measured and predicted values of the magnetic field are shown for both the shielded and unshielded case with a central field value of 2.0 Tesla, which is the proposed operational field for a vortex precooled ADR. The magnetic field strength in gauss is plotted against radial distance along the central place of the magnet. Here, a constant relative permeability of 300 is assumed. This is equal to the initial permeability of the shielding material AD-MU-00. The data is represented by discrete points and the predictions generated by the program SHIELDIN are shown as solid lines. Good agreement is seen between measured and predicted values for both the unshielded and shielded cases. In the shielded case, the remaining field was almost too small to measure. Thus, very adequate shielding has been demonstrated by the soft iron shield method. An HP flux gate magnetometer was used for the field measurements.

In Figure 34, the same quantities are presented for a central field of 4 Tesla. Agreement between predictions and data are quite good for the unshielded case, but the measured field is somewhat larger than the field predicted by the program. We attribute this discrepancy to be due to the fact that in the 4 Tesla field case, the field is approaching saturation of the shield material, and has caused the true value of the permeability to be much

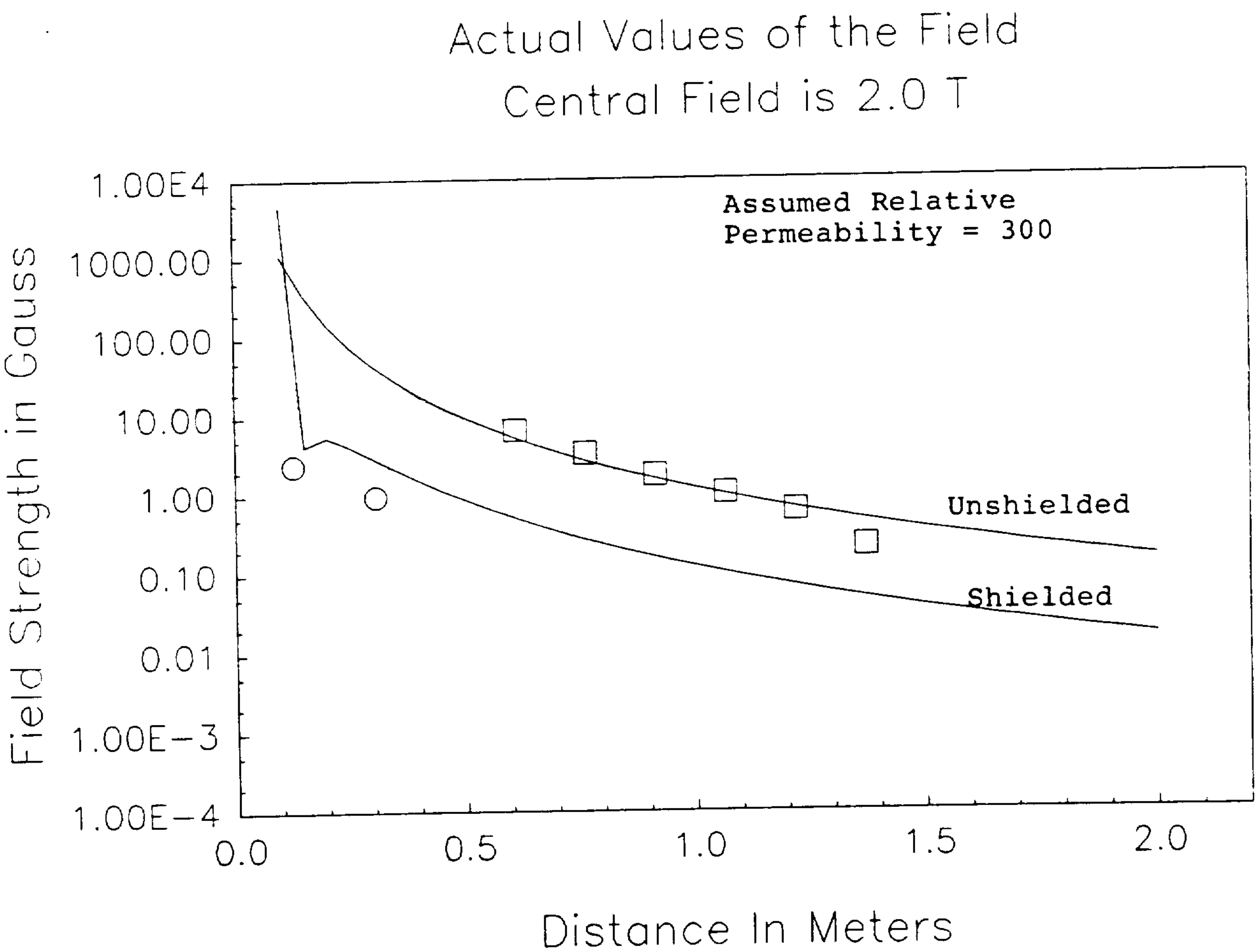

Figure 33 Effect of the Soft Iron shield on the magnetic field of the Superconducting magnet. Field strength is plotted as a function of radial distance from the center of the magnet. The upper solid line represents the field strength in the unshielded case evaluated by the ACE, Inc. Computer Program SHIELDIN. The square symbols are direct measurements of the unshielded field. The lower line gives the predicted shielded field, and the circles represent measured shielded value. The central field of the magnet is 2 Tesla.

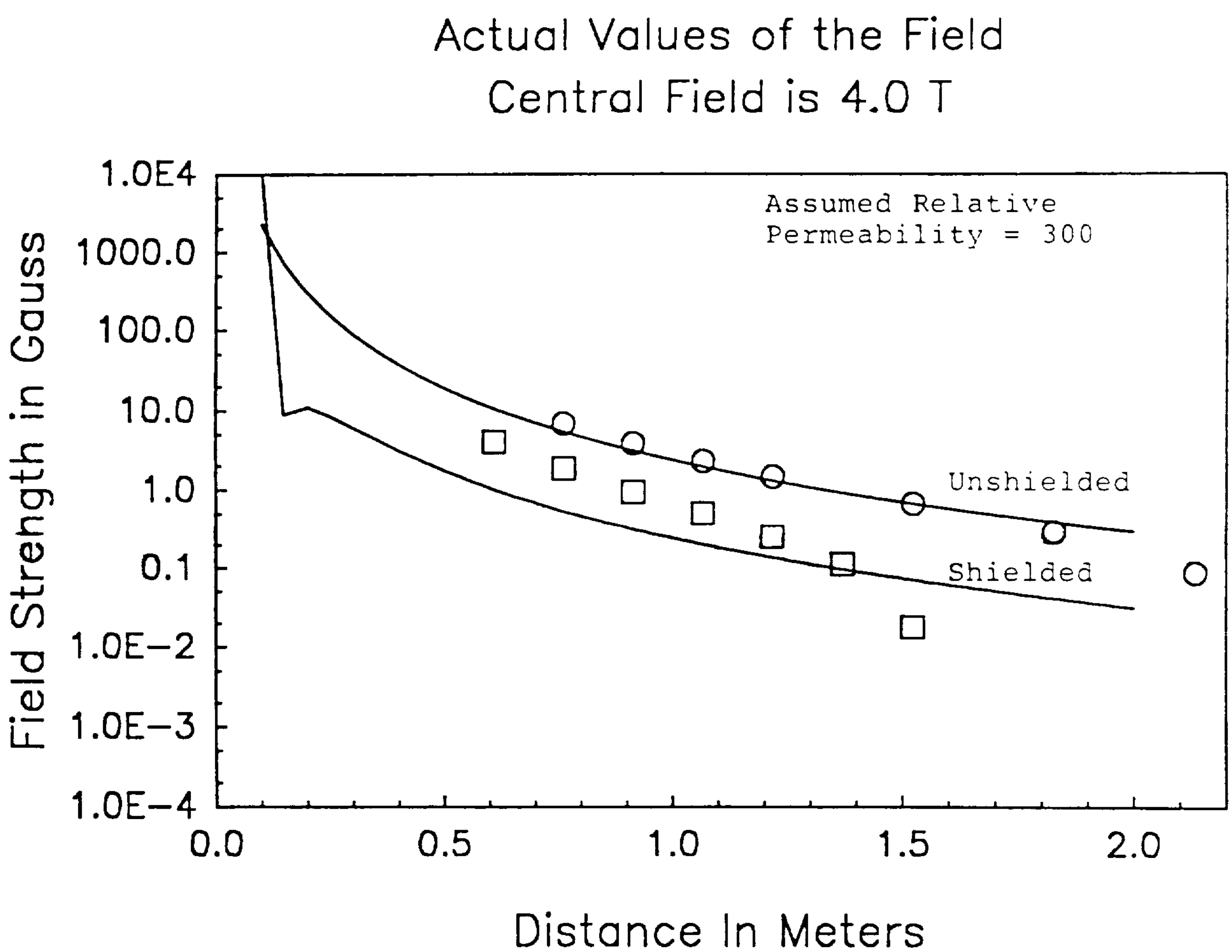

Figure 34 Effect of the Soft Iron Shield on the Magnetic field of the Superconducting Magnet. Field strength is plotted as a function of radial distance from the center of the magnet. The upper solid line represents the field strength in the unshielded case evaluated by the ACE, Inc. Computer Program Shieldin. The circular symbols are direct measurements of the unshielded field. The lower solid line gives the predicted shielded field, and the squares represent measured shielded values.

smaller than the assumed value of 300. Error analysis for the shielding program, particularly in the region very near the shield, was not achieved due to the complexity of the expansion approximation used. The excellent shielding observed in the 2 Tesla case demonstrate the feasibility of the soft iron shield technique.

Feasibility of a Superfluid Cooled Magnet

Studies were undertaken to design a pressurized superfluid cooled superconducting magnet with optimized current leads. As development continued, it was judged that the concept had some drawbacks when compared to newly built small size conventional superconducting magnet systems. Thus, on a basis of cost, reliability, and complexity, a pressurized superfluid cooled magnet is not recommended for the space based ADR.

Castles (1986) has shown the feasibility of a reduced size, indirectly cooled superconducting magnet suitable for space usage. This magnet produces a field of approximately 4 Tesla, but requires only 3 amps of current. For the ACE, Inc. maximum field requirement of only 2 Tesla this implies a current of only 1.5 amps with the same magnet design. Thus, conventional solid magnet leads should be feasible.

VII. CONCLUSIONS

In this section, a summary is presented of the conclusions reached from the development programs described in the previous 3 sections. These programs resulted in the successful testing of a magnetically shielded, vortex precooled adiabatic demagnetization refrigerator.

1. Vortex Precooler System

(A) A baseline of required performance for the vortex precooler was set. A total cooling power of 4.95 mW at 1.1 K was desired to sufficiently remove heat during initial magnetization. A minimum operating temperature of 0.8 K was needed to insure proper operation of the heat switch.

(B) Vortex precoolers were successfully tested using both fountain pump/coiled bath heat exchanger and external gas supply/concentric tube heat exchanger systems. Of these two methods, the fountain pump drive system was judged to be better suited for the ADR test rig configuration. However, both methods were found to be viable in a vortex precooler system.

(C) The performance of vortex cooler systems was found to depend sensitively on vortex cooler exit capillary dimensions, fountain pump exit capillary dimensions, and vortex cooler superleak geometry. This conclusion is based on the results of tests conducted where these parameters were varied.

(D) The finalized ACE, Inc. vortex precooler reached a minimum temperature of 0.78 K, and could deliver approximately 1 mW of cooling power at 1.1 K, with a fountain pump input power of 35 mW.

(E) The baseline vortex precooler system requirement given in (A) can be met by combining five of the vortex coolers described in (D) together in parallel.

2. Vortex Precooled Adiabatic Demagnetization Refrigerator

(A) Preliminary measurements of the conductance of superfluid ^{4}He in a capillary were conducted using a ^{3}He cryocooler to achieve a capillary temperature of 0.65 K on the cold end, and varying the upper end temperatures from 0.9 K to 1.4 K. The data showed the expected sharp dependence of the heat conduction on the temperature of the warm end of the capillary, and agreed qualitively with the predictions of Bertman and Kitchens. The discrepancy between data and theory is believed to be due to Kapitza resistance effects at the end of the capillary.

(B) Ruby and other exotic refrigerants offer promise as ADR materials. Preliminary tests were conducted on using ruby as a refrigerant for an ADR. A brief description of these tests and a discussion of ruby and other materials as ADR refrigerants is included in Appendix 1.

(C) A single vortex cooler configuration was found to be the best for the ADR heat switch apparatus. It was discovered experimentally that proposed methods of using two fountain pump driven vortex coolers in the heat switch would not work because of backflow through the fountain pumps. Two vortex cooler heat switch configurations with room temperature gas supplies were not tested due to the unsuitability of the cryostat for this type of vortex driver.

(D) A vortex precooled adiabatic demagnetization refrigerator was successfully demonstrated. A vortex cooler was used to precool the Ferric Ammonium Alum salt pill to approximately 0.85 K. The Ferric Ammonium Sulfate salt pill was used to reach a minimum temperature of 0.107 K and to maintain the temperature below 0.125 K for more than 90 minutes. Demagnetization for this run was from a two Tesla field.

3. Magnetic Shield Development

(A) A soft iron shield can be developed that gives significant shielding to an ADR apparatus. The ACE, Inc. magnetic shield reduced the radial field of a 2 Tesla magnet to 1 gauss at 0.25 meters.

(B) A FORTRAN computer program has been developed to predict the shielded and unshielded values of the magnetic field for the tests described in (A). Good agreement between predictions and measured data were observed.

(C) Advances in magnet design, such as by Castles (1986) make the additional compilation of a superfluid cooled magnet system unneccessary. Substantial reductions in magnet size and charging currents, coupled with the reduced field requirement of the ACE, Inc. ADR, greatly reduce required magnet weight and dimensions.

It is therefore concluded that a vortex precooled, soft iron shielded adiabatic demagnetization system for zero gravity use is feasible, and offers significant advantages over standard techniques. These include reduced magnetic field requirements, smaller magnet size, and decreased interference from fringing magnetic fields.

VIII RECOMMENDATIONS

The development and successful testing of a vortex precooled adiabatic demagnetization refrigerator has led to a detailed understanding of the design issues involved in the construction of a space qualified system of this type. Based on this understanding, the following recommendations are given:

(1) A program should be undertaken to further optimize the vortex precooler system. This program would be dedicated to the following objectives:

(A) Improve the cooling power of the vortex precooling system. This would involve tests of linking several individual vortex coolers together in parallel to obtain a better cooling power. Also, further work would be done to improve the design of the individual vortex cooler element.

(B) Improve the thermodynamic efficiency of the vortex cooler/fountain pump cooling cycle. Increased fountain pump efficiency would lessen heat loads on the 1.8 K superfluid helium storage dewar, thus increasing system lifetime. Recently, Frederking (1987) has addressed the issue of thermal efficiency in vortex cooler /fountain pump systems. He points to the fountain pump performance as a major limiting factor. However, increases in efficiency can be achieved by using multiple stage fountain pump /heat exchangers in series. This reduces the ΔT across the pump, and gives overall efficiencies nearer to the Carnot limit.

(2) A space capable prototype system should be designed for a specific set of user baseline requirements. The final results from the study described in (1) should be used as a guide to identify the perspective uses of the system.

(3) After a user has been selected, and a prototype design developed, this design should be constructed and flight tested.

IX. REFERENCES

Ackermann, F., Klawitter, W., and Drautman, J. (1971) "Magnetic Properties of Commercial Soft Magnetic Alloys at Cryogenic Temperatures", Advances in Cryogenic Engineering, vol. 16, Plenum Press, NY.

Bertman, B. and Kitchens, T.A. (1968), "Heat Transport in Superfluid Filled Capillaries", Cryogenics, 8, Feb. 1168.

Castles, S. (1980) "Design of an Adiabatic Demagnetization Refrigerator for Studies in Astrophysics", NASA Report X-732-80-9, Goddard Space Flight Center.

Castles, S. (1986) Private Communication.

Frederking, T., Tran, H., and Caranbang, R. (1987), "Refrigeration and Heat Pump Systems Based on He II Vortex Control", Preprint.

Guenin, B.M. and G.B. Hess (1980), "Liquid Helium II Counterflow in a Tube Packed with Angular Particles and Application as a Superfluid Pump," Physica 101B, p. 285.

Kapitza, P., J. Phys. USSR 5 (1941) 59 and Phys. Rev. 60 (1941) 354.

Kurti, N., Robinson, F.N.H., Simon, F. and Spohr, D.A., (1956), Nature, London, 178, 450.

Lounasmaa, O.V. (1974), Experimental Principles and Methods Below 1 K, Academic Press, New York, pp. 5-16.

Olijhoek, J.F., J.K. Hoffer, H. Von Beelen, R. de Bruyn Oubouter, and K.W. Taconis (1973), "Cooling by Convective Heat Transport in Superfluid Helium," Physica, 64, p. 289.

Olijhoek, J.F., H. Von Beelen, R. De Bruyn Oubouter, K.W. Taconis, and W. Koops (1974a), "Thermal Effects in Adiabatic Flow of He II - Part I," Physica, 72, p. 355.

Olijhoek, J.F., et al. (1974b), "Thermal Effects in Adiabatic Flow of He II - Part II", loc. cit., p. 381.

Satoh, T. and T. Satoh (1983), "Studies of Cooling Effects in a Vortex Cooler" Proc. ICEC9, Butterworth, Berks U.K., p. 279.

Satoh, T., H. Shinada, and T. Satoh (1982), "Turbulence and Cooling Effect in Adiabatic Flow of He II", Physica, 114B, p. 167.

Shu, Q., Fast, R., and Hart, H. (1986), "Heat Flux from 277 to 77 K Through a Few Layers of Multilayer Insulation", Cryogenics, Vol 26, 12.

Staas, F.A. and A P. Severijns (1969), "Vorticity in He II and its Application in a Cooling Device," Cryogenics, 9, p. 422.

White, G.K. (1979) Experimental Techniques in Low Temperature Physics, Clarendon Press, Oxford, pp. 202-207.

APPENDIX I

ALTERNATE ADIABATIC DEMAGNETIZATION MATERIALS

When Adiabatic Demagnetization (AD) is used as a refrigeration technique, the AD material must be carefully chosen. The materials commonly used are salts of a paramagnetic ion. These salts are well suited for AD because the magnetic species is sufficiently diluted to reduce magnetic interactions between the ions, while the properties of the salts act to reduce the crystal field splitting of the ion levels. However, due to the nature of these salts, they are fragile and must be handled carefully. The salts also contain H_2O as a necessary ingredient, thus one must not let them warm to room temperature under a vacuum. While these complications can be easily overcome in a typical laboratory environment, they pose considerable problems for use in space flight applications; therefore, we have considered possible alternatives.

Ruby is a form of sapphire (Al_2O_3) which has had a few per cent of the Al atoms replaced by Cr^{3+}. This ionic species has a total spin $J = 3/2$ ($L=3$, $S=3/2$), which implies a theoretical Lande′ g factor of 0.40. However, EPR measurements of ruby[1,2] have found a g factor of $g_{||} = 1.99$ and $g_{\perp} = 1.98$ (the direction is relative to the Al_2O_3 c-axis). This indicates that most of the orbital angular momentum L has been quenched, as is typical for crystalline materials, resulting in a spin only moment of $S=3/2$. The measured g factors further indicate that the magnetic response is quite isotropic. In addition, the EPR measurements show that the crystal field splitting, δ, of the Cr^{3+} levels is only 0.55 K.

One may calculate the Curie constant C (given by M = c H/T, where M = magnetization) from the spin and the measured g factors. For comparison, the molar Curie constant C_m is most convenient and is given by

$$C_m = NA\,[\mu_B^2\ g^2\ J\ (J=1)]\ /\ (3\ k_B) \qquad (I\text{-}1)$$

where NA = Avogadro's number,

μ = the Bohr Magneton, and

k_B = Boltzmann's constant.

This gives a value of C^m = 1.85 cm^3 - K/mole Cr for ruby.

In order for a material to be useful for AD, it must have a reasonable magnetic moment (so that it will interact with the applied field) and have δ small compared to k_BT. Ruby has been shown from the EPR data to have a reasonable moment and small δ; however, to confirm its usefulness comparison should be made to paramagnetic salts. The salts CPA ($Cr\ K(SO_4) \cdot 12H_2O$) and CMA ($Cr\ (CH_3\ NH_3)(SO_4)_2 \cdot 12H_2O$) both have C_m values very similar[3] to ruby, 1.84 and 1.83 cm^3 - K/mole Cr, respectively, and measured g factors very near 2. Furthermore, the crystal field splitting of the ion levels is 0.39 K for CPA[6] and 0.255 K for CMA[7].

From simple thermodynamic considerations, the initial and final temperatures can be related through the magnetic field by the relation[8]

$$b + (C_m H^2/R) = [(b/T_f^2) - (2a/3)\ T_f^3]\ T_i^2 + (2/3)a\ (T_i^5), \qquad (I\text{-}2)$$

where a and b are related to the heat capacity by

$$C = RaT^3 + RbT^{-2}, \tag{I-3}$$

where b is due to magnetic interaction and a is due to phonons. For CPA and CMA, assuming an initial temperature of 1 K and a field of 10 kOe, $T_f = 0.089$ K while for ruby this is 0.089-0.18 K depending on the value of b used. In addition, the change in entropy per mole is dependent on the magnetic field, temperature and C_m ($\Delta S/\text{mole} = -C_m H^2/2T^2$), thus all three materials would have the same change in entropy.

From these findings we feel that ruby, with a Cr concentration of 4.6 wt %, would potentially perform as well as CPA or CMA as an AD material, but does not possess the complications associated with H_2O content. In addition, ruby (that is Al_2O_3) has a hardness of 9 making it much stronger than any paramagnetic salt. Ruby single crystals may be obtained from Union Carbide Corporation[9] with a 1 wt % Cr concentration. These crystals would have to be custom grown at an approximate cost of $12,000.00 and a delivery time of 3-4 months; however this would result in a 2.5 in dia. x 8 in. long crystal.

This investigation of ruby further suggests that some other laser materials should be considered. Most glasses used in laser applications are doped with Nd:YAG and the Nd glass ceramic. All of these are commercial available and have the highest concentrations of Nd (eg. the phosphate glass Q100, made by Kigre, contains 9 wt % Nd).

Nd^{3+} has a total spin J = 9/2 (S = 3/2, L=6) which results in a theoretical g factor of 0.727. Unlike the case of ruby, the host matrix here is a glass, thus the periodicity of the crystal lattice does not exist. The crystal fields, which quenched the orbital angular momentum of the Cr ions, are not present and the resulting C_m is 1.63 cm^3- K/mole Nd. This value is slightly smaller than ruby; therefore, it is felt that one could use these glasses, as is, as a first test.

As a further improvement the glass could be formed with Gd^{3+} (J = 7/2, L=0, S = 7/2; g=2) as the dopant rather than Nd. The covalent radius of Nd is 1.64 A while, for Gd, it is 1.61 A, thus substitution should not be a problem. Since Gd has L=0, the lack of orbital spin quenching will have no detrimental effect on the Curie constant. The resulting C_m is 7.88 cm-K/mole Gd; almost 5 times larger than Nd and 4.3 times larger than ruby. Therefore, doping the glass with Gd would provide 4-5 times more entropy change (i.e., cooling power).

This discussion suggests that ruby, phosphate laser glass and Nd:YAG are possible alternatives to the standard salts for AD. Since all these materials are available, it is felt that AD experiments should be performed using these materials.

A test of ruby as an ADR material was performed at Alabama Cryogenic Engineering, Inc. A 5 cm^3 sample of ruby with an estimated 1% chromium concentration by weight was used as a sample. The ruby was completely demagnetized from a field of 4.2 Tesla and an initial temperature of 4.2 K. The demagnetization took place in approximately 5 minutes and the sample reached a minimum temperature of 1.5 K. In a second test, the ruby was demagnetized at the same rate from 4.2 Tesla and an initial temperature of 3.2 K. A minimum temperature of 0.39 K was reached.

REFERENCES

1. T. Change and A.H. Kahn, NBS Special Publication 260-59 (1978), p. 5.
2. G. Kido and N. Muira, Appl. Phys. Lett. 41 (1982) 569.
3. R.P. Hudson, Principle and Application of Magnetic Cooling (North Holland, Amsterdam, 1972) p. 110.
4. Ibid., p. 125.
5. J.M. Baher, Proc. Phys. Soc. B69 (1956) 633.
6. B. Blearez, Proc. Roy. Soc. A204 (1950) 203.
7. M. Durienx, H. van Dijk, H. ter Harmsel and C. van Rijn, Temperature - Its Measurement and Control in Science and Industry, Vol. 3 (Reinhold, New York, NY, 1962) p. 383.
8. R.P. Hudson, Principle and Application of Magnetic Cooling (North-Holland, Amsterdam, 1972) pp. 5-13.
9. Private communication, Jack Mellone, Union Carbide Corporation, Seeking, MA.
10. N. Neuroth, Optical Engineering 26 (1987) 96.
11. Sargent-Welch, "Table of Periodic Properties of the Elements", Sargent-Welch Scientific Co., 1968.

APPENDIX II

ALABAMA CRYOGENIC ENGINEERING, INC. MAGNETIC SHIELDING PROGRAM

```
      PROGRAM SHIELD
C
C     FILE IS SHIELD.FOR
C     5/25/88
C     FINAL ALTERATIONS BY CHRIS MILTENBERGER
C     "INCLUDE" FILES AND "NUMRCP2.*" FILES CONCATENATED INTO ONE FILE
C
C     TO LINK, TYPE =>   C:\F77L>LINK SHIELD+MWSUBS,SHIELD,NUL,WNDOWF77
C
C     ----------------------------------------------------------------------
C     DELIVERED TO ACE 12/7/87
C     COMPILE AND LINK WITH FMG2, WHICH LINKS WITH NUMRCP2 AND MWSUBS
C
C     THIS IS THE VERSION FOR STRAIGHT LINE SEGMENTS FOR
C     SHIELD SHAPES.
C
C     IN THIS VERSION, USER ENTERS SHIELD SHAPES IN TERMS OF
C     STRAIGHT LINE SEGMENTS, DEFINED BY UP TO 20 POINTS.
C     ENTRY CAN BE EITHER BY POINTS DEFINED BY MOUSE ON SCREEN
C     OR THRU DIRECT ENTRY OF X,Y COORDINATES.  IN EITHER CASE
C     USER ENTERS POINTS DEFINING INNER SHIELD, STORED AS CARTESIAN
C     COORDINATES IN XA,YA, FROM WHICH RADIUS AND ANGLE, R_A, THA
C     ARE COMPUTED.  THE ANGLE THE ITH LINE MAKES WITH RESPECT TO
C     THE Y-AXIS IS STORED AS ALP(I).
C     THE FOLLOWING CONDITIONS MUST BE MET DURING DATA ENTRY,
C     OR DATA IS REJECTED:
C             EACH X>=0, Y>=0, THA(I+1)>=THA(I), ALP(I)<=PI
C     THUS, INFORMALLY, DATA MUST BE ENTERED IN CLOCKWISE MANNER MAKING
C     CONVEX SET WITH NO RE-ENTRIES.
C     THE FIRST POINT ENTERED IS TAKEN TO DETERMINE THE GAP ANGLE.
C     NOTE: THIS DIFFERS FROM PREVIOUS, SHIELDIN, WHERE GAP ANGLE
C     WAS ENTERED SEPARATELY.
C
C     THE OUTER SHIELD POINTS, XB,YB, ARE DEFINED TO BE ON
C     INTERSECTION OF LINES PARALLEL TO INNER ONES, BUT SEPARATED
C     BY DISTANCE EQUAL TO USER ENTERED THICK.
C     SEE HANDWRITTEN NOTES:  FMAGS GEOMETRY, 11/29/87.
C
C     NOTE: FIRST POINT ON OUTER SURFACE IS ON RADIUS THRU FIRST POINT
C     ON INNER SURFACE, I.E., THB(1)=THA(1)=GAP ANGLE
C
C     -->  FOLLOWING PRECEDE 12/5/87
C     ALL LENGTHS ARE STORED INTERNALLY(FOR GRAPHICS DISPLAY)
C     IN UNITS OF CURRENT SOURCE CYLINDER RADIUS, S_SCALE
C
C     DELIVERED TO ACE 9/13/87
C     PREVIOUS DISAGREEMENTS WITH TBASIC VERSION WERE FOUND TO
C     DUE TO EVALUATION OF MU(TH) EXACTLY AT SHIELD EDGE.
C     TBASIC VS F77L PRODUCE DIFFERENT VALUES BECAUSE THEY HANDLE
C     ROUNDOFFS AND VALUE OF PI DIFFERENTLY.  THIS INCONSISTENCIES
C     IS ELIMINATED BY INCREASING NUMBER OF AVERAGES UP TO 10, WHICH
C     IS CURRENT RECOMMENDED VALUE.
C
C     -->  FOLLOWING REPLACE 9/13/87
C     TEST VERSION MAILED TO ACE 9/3/87.  DISAGREES WITH TBASIC
C     VERSION IN VALUES OF FIELD!!!!  EITHER ERROR IN CONVERSION
C     OR DISAGREEMENT IN NUMERICAL CALCULATION !!!!
C
C     CONVERSION FROM .TRU VERSION 9/3/87
C     AS OF 9/3/87, AFTER GRAPH DISPLAY, WILL PAUSE, AWAITNG
```

```
C     CHARACTER IN UNIT 10 (CONSOLE, TRANSPARENT).
C     ENTRY OF P WILL CAUSE A CALL TO MW$PRINT(1), UNIT1=PRINT
C     ONLY IF -1313 HAS BEEN ENTERED FOR FIRST NUMBER(SHIELD RADIUS).
C     THIS IS CODE ACTIVATTING OKIDATA PRINT ROUTINE. SWITCH IS
C     LOGICAL OKI_PRINT.
C     IN DELIVERY TO ACE, THIS IS NOT NEEDED, BECAUSE THEY
C     HAVE SATISFACTORY DUMP-TO-PRINT PROCEDURE OF THEIR OWN.
C
C     --> MODIFIED 7/24/87 AS FOLLOWS:
C     <<<<<< NOTE 12/5/87 >>>>>>
C     THIS PARAGRAPH 1 IS OBSOLETE FOR SHIELDN2.  SEE ABOVE
C     1) IF USER ENTERS SHIELD SHAPE GRAPHICALLY USING MOUSE,
C        HE ONLY ENTERS INNER SHIELD SURFACE AND THICKNESS.
C        SYSTEM THEN COMPUTES FOURIER SERIES TO APPROXIMATE
C        OUTER SURFACE AT NORMAL DISTANCE=THICKNESS.  WARNING:
C        IF INNER SURFACE IS TOO CURVED, OUTER SURFACE SO
C        GENERATED MAY BE AT A NORMAL DISTANCE SIGNIFICANTLY
C        DIFFERENT FROM THE ENTERED THICKNESS IN CERTAIN REGIONS.
C        THIS INVOLVES NEW SUBROUTINE OUTER BASED ON HANDWRITTEN
C        NOTES: CONSTANT NORMAL THICKNESS REGION 7/20/87
C
C     2) USER ENTERS CURRENT PER UNIT LENGTH IN "CENTRAL FIELD"
C        UNITS (TESLAS), EQUIVALENT TO MU0*CURRENT/LENGTH.
C
C     3) PRINTED OUTPUT INCLUDES UNSHIELDED FIELD STRENGTH.
C
C     4) FOLLOWING GRAPHICAL DISPLAYS, USER IS ASKED TO HIT
C        MOUSE BUTTON/RETURN TO CONTINUE.  THIS GIVES HIM TIME
C        LOOK AT DISPLAY AND, IF DESIRED, USE ALTPRTSCRN TO
C        "GRAB" IMAGE FOR DRHALO.  MOUSE POINTER CAN BE MOVED
C        OFF SCREEN IF DESIRED.
C
C     --> FOLLOWING REPLACED 7/24/87
C     THESE WERE NOTES FOR TBASIC VERSION
C     MODIFIED 3/26/87 TO GET .EXE VERSION, BY CHANGING
C     LIBRARY "NUMRCP" TO LIBRARY "NUMRCP.TRC"
C     CARL H. BRANS    1/8/87
C     THIS IS THE FINAL DELIVERY VERSION TO ACE OF PROGRAM
C     WHICH PRODUCES AN ESTIMATE TO THE CONSTANTS INVOLVED IN
C     THE LEGENDRE EXPANSION IN SPHERICAL COORDINATES OF THE
C     MAGNETIC FIELD DUE TO A CYLINDRICAL SOURCE, SURROUNDED
C     BY A SHIELD WHOSE SURFACES ARE INPUT BY USER.  THE SHIELD
C     MAY ALSO HAVE A VACUUM GAP DEFINED BY A USER ENTERED ANGLE
C     WITH RESPECT TO THE Z-AXIS.  THE PROBLEM IS ASSUMED TO HAVE
C     FULL SYMMETRY UNDER ROTATIONS ABOUT THE Z-AXIS AS WELL AS
C     INVERSION ABOUT THE Z-AXIX, I.E., Z --> -Z.  THE APPROACH
C     USED IN THIS PROGRAM MAKES USE OF THE LEGENDRE EXPANSION
C     WITH DIFFERENT COEFFICIENTS IN EACH OF THE THREE REGIONS:
C     R<SRA(TH), SRA(TH)<=R<SRB(TH), AND R>SRB(TH), WHERE THE
C     INNER AND OUTER SHIELD SURFACES ARE DEFINED BY THE FUNCTIONS
C     SRA(TH), SRB(TH).  THE USUAL MAGNETOSTATIC CONTINUITY
C     AND CONDITIONS AT INFINITY THEN IMPOSE SUFFICIENT EQUATIONS
C     TO DETERMINE THESE COEFFICIENTS.  HOWEVER, IF THE SHIELD
C     SURFACES ARE NOT SPHERICAL, THESE CONDITIONS CANNOT BE
C     SOLVED EXPLICITLY IN CLOSED FORM.  THE APPROXIMATION OF
C     THIS PROGRAM CONSISTS IN KEEPING ONLY A FINITE NUMBER OF
C     TERMS IN THE LEGENDRE EXPANSION AND THEN IMPOSING THE
C     MATCHING CONTINUITY CONDITIONS AND ONLY A DISCRETE NUMBER
C     OF POINTS, SUFFICIENT TO FULLY DETERMINE THE NOW FINITE
C     NUMBER OF LEGENDRE COEFFICIENTS.  THIS IS EXPLAINED IN DETAIL
```

```
C     IN THE NOTES: "ACE MAGNETIC SHIELDING PROGRAM",1/7/87 AND
C     "DISCRETE SHIELDING, SYMMETRIC", 12/10/86.
C
C     THIS PROGRAM MAKES USE OF SUBROUTINES, CONTAINED IN COMPILED
C     LIBRARY, NUMRCP.  THEY INCLUDE THE FOLLOWING.
C
C     GAUSSJ   FROM NUMERICAL RECIPES FOR SOLVING LINEAR EQUATION SET
C     MAKEP    MAKES LEGENDRE FUNCTIONS
C     GFNC     FUNCTION USED CYLSRC
C     CYLSRC   SUBROUTINE TO COMPUTE FREE (UNSHIELDED) FIELD FROM
C              CYLINDRICAL SOURCE
C     QROMB, TRAPZD, POLINT, FROM NUMERICAL RECIPES, FOR DOING NUMERICAL
C              VOLUME INTEGRAL FOR SHIELD
C     SRA,SRB,NRA,NTHA,NRB,NTHB FUNCTIONS GIVING RADIUS AND NORMAL
C              COMPONENTS OF SHIELD
C     MU       FUNCTION GIVING (REAL) MU AS FUNCTION OF THETA.  VALUE IS
C              MU_0 IN GAP, ELSE MU_1. THIS IS RELATIVE MU
C     OUTER    FINDS OUTER SHIELD SHAPE FROM INNER ONE AND USER ENTERED
C              THICKNESS
C     FUNC     FUNCTION USED IN NUMERICAL INTEGRATION, QROMB, FOR THE
C              SHIELD VOLUME.
C     GRAPH    SETS UP GRAPHICS MODE
C     GETXY    GETS MOUSE X,Y COORDINATES
C
C     IN ADDITION, GRAPHICS NEEDS META WINDOW SUPPORT, INCLUDING
C     F77L SUBROUTINES IN MWSUBS.  THUS, AFTER F77L COMPILE,
C
C     LINK MWSUBS+SHIELDIN,SHIELDIN,NUL,WNDOWF77
C
C     THIS SECTION IS CONCERNED WITH BACKGROUND DATA DEFINITION,
C     GETTING THE SHAPE OF THE SHIELD SURFACES.  IT ALSO STARTS THE
C     PROGRAM'S MASTER LOOP, SO THAT PROGRAM CAN BE RE-RUN.
C
C     FOR MATHEMATICAL ANALYSIS AND FURTHER INFORMATION ON NOTATION
C     SEE HANDWRITTEN NOTES:
C
C        SPHERICAL MAG SHIELD   8/20/86
C        DISCRETE SHIELDING, SYMMETRIC 12/10/86
C        MAGNETIC SHIELDING PROGRAM 1/7/87
C        SPHERICAL SHIELDING FACTOR 1/12/87
C        FMAGS GEOMETRY 11/29/87
C
C     XA,YA ARE INNER POINTS ENTERED
C     XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C     R_A,THA,R_B,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
C     ALP(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),ALP(20)
      REAL R_A(20),R_B(20)
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,ALP,R_A,R_B,NP
      LOGICAL OKI_PRINT!CONTROLS CALL TO MW$PRINT NOT USED AT ACE
      CHARACTER*1 FF_P$,EMP$*2,STEMP$*2,FN$*20,FOUT$*20
      DIMENSION A(100,100),B(100),C21(50),C12(50),C22(50),C13(50)
      DIMENSION PMN(0:1,0:50),QMN(0:1,0:50),RMN(0:1,0:50)
C     PMN,QMN,RMN HOLD THE LEGENDRE POLYNOMIALS EVALUATED AT VARIOUS
C     ANGLES
      DIMENSION FA(0:19),FB(0:19)
      REAL MUI,MU0,MU,MU_0,MU_1,NRA,NTHA,NRB,NTHB,NRI,NTHI
      COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
      CHARACTER*70 TITL$,MES$,SP$*1,DATEX*8,TIMEX*8,TIMEY$*8,XP$*1
```

```
      CHARACTER*8 X$,Y$,C*20,MES2$*70,MES3$*70
      CHARACTER*70 CH$,MSG*50
      COMMON/GMODE/G_M!G_M=0/1 FOR GRAPH OFF/ON
      EXTERNAL FUNC
      OPTION BREAK(*9999)
      OKI_PRINT=.FALSE.
      CALL GETCL(X$)
      IF(X$.EQ.' OKI') OKI_PRINT=.TRUE.
      FF_P$=CHAR(12)!PRINTER FORM FEED
      EMP$=CHAR(27)//CHAR(69)!PRINTER EMPHASIZES
      STEMP$=CHAR(27)//CHAR(70)!STOP PRINTER EMPHASIS
C     A AND B ARE THE MATRICES DEFINING THE LINEAR EQUATION
C
C        AX = B
C
C     TO BE SOLVED FOR X BY SUBROUTINE GAUSSJ.
C     THE MATRICES C21,C12,C22,C13 ARE THE COEFFICIENTS IN THE
C     LEGENDRE EXPANSION OF THE FIELD IN THE THREE REGIONS, AS
C     EXPLAINED IN THE NOTES: "DISCRETE SHIELDING, SYMMETRIC", 12/10/86
      TITL$="ALABAMA CRYOGENIC ENGINEERING  MAGNETIC SHIELDING PROGRAM"
      CALL DATE(DATEX)
      CALL TIME(TIMEX)
      MES$="SHIELDN2 F77L VERSION 12/5/87, RUN AT: "//DATEX//
     X ", "//TIMEX
      MES2$="THIS VERSION USES STRAIGHT LINE SEGMENTS TO"
      MES3$="DEFINE SHIELD SURFACES"
      SP$=" "
      PI=4.*ATAN(1.)
      OPEN(UNIT=10,FILE='CON',ACCESS='TRANSPARENT')
C     10=CONSOLE FOR USER CONTINUATION AFTER GRAPH DISPLAY PAUSE
      OPEN(UNIT=1,FILE='LPT1')
C     1=PRINTER
1     CONTINUE
      G_M=0!GRAPH MODE OFF
C     THIS IS THE BEGINNING OF THE PROGRAMS'S MASTER LOOP
      CLOSE(2)
      CLOSE(7)
      WRITE(*,*) " "
      WRITE(*,*) "    ",TITL$
      WRITE(*,*) "    ",MES$
      WRITE(*,*) "    ",CHARNB(MES2$)//" "//MES3$
      WRITE(*,*) " "
      WRITE(*,*)' '
      WRITE(*,*)' AFTER GRAPH DISPLAY, PROGRAM PAUSES TO ALLOW USER'
      WRITE(*,*)' TO VIEW GRAPH AND DUMP TO PRINTER IF WANTED. HIT'
      WRITE(*,*)' ANY KEY TO CONTINUE AFTER SUCH PAUSE.'
      IF(OKI_PRINT) WRITE(*,*)' >>>  THIS IS SET TO DUMP GRAPHICS',
     X ' TO OKIDATA 192 PRINTER.'
      WRITE(*,*)' '
C     ENTER SOURCE CYLINDER PARAMETER
      WRITE(*,*) "ENTER SOURCE CYLINDER RADIUS IN METERS: "
1001  READ(*,*)S_SCALE
C     S_SCALE IS USED INTERNALLY FOR ALL LENGTHS, SO THAT THEY ARE
C     IMMEDIATELY AVAILABLE FOR GRAPH.  HOWEVER, WHEN DATA IS
C     OUTPUT IN NUMERIC FORM, EACH LENGHT MUST BE MULTIPLIED
C     BY S_SCALE
C     S_SCALE WILL SET THE SCALE OF PLOTS AND INTERNAL CALCULATIONS
C     SEE NOTES: "MAGNETIC SHIELDING PROGRAM 1/7/87"
      WRITE(*,*) "ENTER SOURCE CYLINDER TOTAL LENGTH IN METERS: "
      READ(*,*)H
```

```
      H=H/S_SCALE
C     GET SHAPES OF SHIELD SURFACES
      WRITE(*,*)
      WRITE(*,*) "SHIELD SURFACES WILL NOW BE DEFINED.  ENTER 1 TO "//
     X "ENTER NEW SHAPES, OR"
      WRITE(*,*) "2 TO USE THE SHAPES DEFINED BY A PREVIOUS RUN AND "//
     X "CONTAINED IN A KNOWN FILE"
      WRITE(*,*) " "
      WRITE(*,*) "1. NEW SHAPE"
      WRITE(*,*) "2. OLD SHAPE"
      N=0
2     WRITE(*,*)  "ENTER CHOICE (1 OR 2): "
      READ(*,*)N
      IF(N.NE.1.AND.N.NE.2) GOTO 2
      FN$=" "                            ! FN$=FILE NAME
      IF(N.EQ.2) THEN
      WRITE(*,*) "ENTER FILE NAME: "
      READ(*,100)FN$
100   FORMAT(A)
      END IF
      IF(FN$.EQ." ") THEN!                              --- IF FN$ START
      NP=0!NP=NUMBER OF USER ENTERED POINTS
      WRITE(*,*) "SHIELD SURFACES CAN BE DEFINED BY DIRECT "//
     X "ENTRY OF COORDINATES,"
      WRITE(*,*) "OR BY MOVING CURSOR TO A NUMBER OF POINTS, "//
     X "WHICH WILL THEN"
      WRITE(*,*) "DEFINE STRAIGHT LINE SEGMENTS.  CHOOSE 1 TO "//
     X "ENTER COORDINATES DIRECTLY,"
      WRITE(*,*) "OR 2 TO ENTER POINTS"
      WRITE(*,*) " "
      WRITE(*,*) "1. ENTER COEFFICIENTS"
      WRITE(*,*) "2. ENTER POINTS ON GRAPH"
      NC=0
4     WRITE(*,*) "ENTER YOUR CHOICE (1 OR 2): "
      READ(*,*)NC
      IF(NC.LT.1.OR.NC.GT.2) GOTO 4
6     WRITE(*,*)'ENTER NUMBER OF POINTS(LESS THAN 21): '
      READ(*,*)NP
      IF(NP.GT.20) GO TO 6
      WRITE(*,*)"ENTER THICKNESS IN CENTIMETERS: "
      READ(*,*)THICK
      THICK=THICK/S_SCALE
      THICK=THICK/100                ! CARRY IT IN METERS INTERNALLY
      IF(NC.EQ.2) GO TO 500
103   DO 401 I=1,NP
601   WRITE(*,*)'X,Y: '
      READ(*,*)X,Y
      XA(I)=X
      YA(I)=Y
      IF(X.LT.0.OR.Y.LT.0) THEN
      WRITE(*,*)' CANNOT BE NEGATIVE. RE-ENTER'
      GOTO 601
      ENDIF
      IF(I.GT.1) ALP(I-1)=F_ATAN(XA(I)-XA(I-1),YA(I)-YA(I-1))
      THA(I)=F_ATAN(XA(I),YA(I))
      IF(I.GT.1) THEN
      IF(THA(I).LT.THA(I-1).OR.ALP(I-1).GT.PI)THEN
      WRITE(*,*)' INVALID. MUST BE CONVEX'
      GOTO 601
      ENDIF
```

```
      ENDIF
 401  R_A(I)=(XA(I)**2+YA(I)**2)**.5
      ALP(NP)=PI
      TH_MU0=THA(1)
      X=TH_MU0*180/PI
      CALL OUTER(IERR)
      IF(IERR.NE.0) THEN
      WRITE(*,*)' ERROR: OUTER SHIELD POINTS ILLEGAL'
      WRITE(*,*)' EITHER NOT CONVEX, OR NEGATIVE X OR Y'
      WRITE(*,*)' '
      GOTO 103
      ENDIF
      CALL GRAPH(S_SCALE,H)
      G_M=1
      CALL MW$LOCAT(1.,8.,0)
      CALL DRAWSTRING(MSG)
      CALL MW$LOCAT(1.,7.5,0)
      WRITE(MSG,300)X
      CALL DRAWSTRING(MSG)
      GOTO 700
 500  CALL GRAPH(S_SCALE,H)
      G_M=1
      CALL MW$LOCAT(1.,8.,0)
      CALL DRAWSTRING(MSG)
      TH=0
      X=0
      Y=0
      DO 701 I=1,NP
      CALL GETXY(X,Y)
 101  XA(I)=X
      YA(I)=Y
      IF(X.LT.0.OR.Y.LT.0) THEN
      CALL ERRS(X,Y)
      GOTO 101
      ENDIF
      IF(I.GT.1) ALP(I-1)=F_ATAN(XA(I)-XA(I-1),YA(I)-YA(I-1))
      THA(I)=F_ATAN(XA(I),YA(I))
      IF(I.GT.1) THEN
      IF(THA(I).LT.THA(I-1).OR.ALP(I-1).GT.PI)THEN
      CALL ERRS(X,Y)
      GOTO 101
      ENDIF
      ENDIF
 701  R_A(I)=(XA(I)**2+YA(I)**2)**.5
      ALP(NP)=PI
      TH_MU0=THA(1)
      X=TH_MU0*180/PI
      WRITE(MSG,300)X
 300  FORMAT(' GAP ANGLE = ',F6.2,' DEGREES')
      CALL MW$LOCAT(1.,7.5,0)
      CALL DRAWSTRING(MSG)
      CALL OUTER(IERR)
      IF(IERR.NE.0) THEN
      CALL MW$TURNOFF(1,1)
      G_M=0
      WRITE(*,*)' ERROR: OUTER SHIELD POINTS ILLEGAL'
      WRITE(*,*)' EITHER NOT CONVEX, OR NEGATIVE X OR Y'
      WRITE(*,*)' '
      GOTO 103
      ENDIF
```

```
      ELSE!DATA FROM FILE                                    ----- FN$ ELSE
      OPEN(2,FILE=FN$,STATUS='OLD',FORM='UNFORMATTED')
C     2 IS DATA FILE
      READ(2)X$
      IF(X$.NE.'SHIELDN2') THEN
      WRITE(*,*) 'DATA FILE NOT PREPARED BY THIS SYSTEM.'
      GOTO 1
      ENDIF
      READ(2)X$,Y$,NP,X1,X2,X3,X5,X6,SA,SB,X7,THICK
      SA=SA/S_SCALE
      SB=SB/S_SCALE
      THICK=THICK/S_SCALE
      X=S_SCALE
      DO 719 I=1,NP
      READ(2)XA(I),YA(I),R_A(I),THA(I),XB(I),
     X YB(I),R_B(I),THB(I),ALP(I)
      XA(I)=XA(I)/X
      YA(I)=YA(I)/X
      XB(I)=XB(I)/X
      YB(I)=YB(I)/X
      R_A(I)=R_A(I)/X
719   R_B(I)=R_B(I)/X
      TH_MU0=THA(1)
      CLOSE(2)
      CALL GRAPH(S_SCALE,H)
      G_M=1!GRAPH MODE ON
      CALL MW$LOCAT(2.,8.,0)
      CH$="FROM FILE "//FN$
      CALL DRAWSTRING(CH$)
      END IF!END OF NEW OR OLD SHAPE IF                      ------ FN$ ENDIF
C     NOW PLOT CURVES FROM COEFFICIENTS
700   L=0
      DO 15 TH=TH_MU0,PI/2,.01
      SRS=SRA(TH)
      XS=SRS*SIN(TH)
      YS=SRS*COS(TH)
      CALL MW$LOCAT(XS,YS,L)
15    L=1
      L=0
      DO 16 TH=TH_MU0,PI/2,.01
      SRS=SRB(TH)
      XS=SRS*SIN(TH)
      YS=SRS*COS(TH)
      CALL MW$LOCAT(XS,YS,L)
16    L=1
      READ(10,116)XP$  ! GIVE USER CHANCE TO SEE OR GRAB
      IF(XP$.EQ."P".AND.OKI_PRINT)CALL MW$PRINT(1)   ! ONLY FOR OKIDATA
116   FORMAT(A1)
      CALL MW$TURNOFF(1,1)
      G_M=0!GRAPHMODE OFF
C     THIS PART CONTINUES ENTRY OF PARAMETERS.
      WRITE(*,'(A,F6.2)') " ANGLE OF GAP IN DEGREES: ",TH_MU0*180/PI
      WRITE(*,*) "ENTER RELATIVE MU "
      READ(*,*)MU_1
      WRITE(*,*) "COMPUTING VOLUME, PLEASE WAIT . . ."
      CALL QROMB(FUNC,TH_MU0,PI/2,SS)
C     THIS IS NUMERICAL INTEGRATION, SEE PART 3
      VOL=(S_SCALE**3)*SS*4*PI/3
      CALL GRAPH(S_SCALE,H)
      G_M=1!GRAPH MODE ON
```

```
      SA=0
      SB=0
C     SA,SB ARE AVERAGE VALUES OF RADIUS ON INNER AND OUTER SURFACES
C     USED IN SCALING POWERS OF R IN LEGENDRE EXPANSION
      N=0
      DO 17 TH=TH_MU0,PI/2,.01!SHOW VOLUME OF SHIELD
      R1=SRA(TH)
      R2=SRB(TH)
      STH=SIN(TH)
      CTH=COS(TH)
      CALL MW$LOCAT(R1*STH,R1*CTH,N)
      CALL MW$LOCAT(R2*STH,R2*CTH,1)
      SA=SA+R1
      SB=SB+R2
17    N=N+1
      SA=SA/N
      SB=SB/N
      WRITE(CH$,117)VOL
117   FORMAT("VOLUME=",E10.4," CUBIC METERS")
      CALL MW$LOCAT(2.,8.,0)
      CALL DRAWSTRING(CH$)
      READ(10,116)XP$                  !   LET USER SEE
      IF(XP$.EQ.'P'.AND.OKI_PRINT) CALL MW$PRINT(1)   ! ONLY FOR OKIDATA
      CALL MW$TURNOFF(1,1)
      G_M=0!GRAPH OFF
      WRITE(*,*) "DATA OUTPUT FILE: "
      READ(*,*)FOUT$
      OPEN(7,FOUT$,FORM='UNFORMATTED')
C     7=OUTPUT FILE
18    WRITE(*,*) "SEND RESULT TO SCREEN(S)/PRINTER(P): "
      READ(*,*)SP$
      IF(SP$.NE.'P'.AND.SP$.NE.'S')GOTO 18
      MU_0=1     !  THIS IS MU VALUE IN GAP. USED BY FUNCTION MU(TH)
      WRITE(*,*) "NUMBER OF AVERAGES (SUGGEST 10) "
      READ(*,*)NAVE
C     THE PROCESS OF MATCHING THE FIELD AT A DISCRETE NUMBER OF POINTS
C     ACROSS SHIELD BOUNDARIES, THEN USING THE RESUTING EQUATIONS TO
C     DETERMINE THE EXPANSION COEFFICIENTS WILL BE REPEATED NAVE TIMES
C     AND THE RESULT AVERAGED.  THIS HAS THE ADVANTAGE OF SAMPLING THE
C     BOUNDARIES AT MORE POINTS, WITHOUT INCREASING THE NUMER OF UNKOWNS
C     TO BE SOLVED FOR.  EXECUTION TIME IS THUS LINEAR IN NAVE, BUT
C     APPROXIMATELY QUADRATIC IN NT
      WRITE(*,*)"NUMBER OF EXPANSION TERMS (MUST BE ODD, SUGGEST 9) "
      READ(*,*)NT
      NT=9
      WRITE(*,*) "NEXT, ENTER CENTRAL FIELD WHICH "//
     X "IS MU0 * CURRENT/LENGTH"
      WRITE(*,*) "ENTER CENTRAL FIELD(TESLAS): "
      READ(*,*)CF
      MU0=4*PI*10.**(-7)    ! STANDARD VALUE IN NEWTONS/AMP**2
      CRNT=CF/MU0
      CRNTF=CRNT*MU0/2    ! THIS MULTIPLIES OUTPUT OF CYLSRC
      WRITE(*,*) "PRINTED OUTPUT WILL BE ALONG RADII, FROM "//
     X "R1 TO R2, STEP DELTA"
      WRITE(*,*) "ENTER R1, R2, AND DELTA IN METERS: "
      READ(*,*)R1,R2,DELTA
      R1=R1/S_SCALE
      R2=R2/S_SCALE
      DELTA=DELTA/S_SCALE
      WRITE(*,*)"RADII WILL BE FOR N DIVISIONS OF PI/2. ENTER N "
```

```
      READ(*,*)NRTH
      NT0=((NT+1)/2)
      IF(NT.EQ.NT0*2) THEN
      NT=NT+1
      NT0=((NT+1)/2)
      WRITE(*,*) "NT RAISED TO ODD NUMBER", NT
      END IF
      HS=H*S_SCALE
      SAS=SA*S_SCALE
      SBS=SB*S_SCALE
      THICK_S=THICK*S_SCALE
      X$='SHIELDN2'
      WRITE(7)X$
      WRITE(7)DATEX,TIMEX,NP,CRNT,HS,NT,TH_MU0,MU_1,SAS,SBS,
     X S_SCALE,THICK_S
      X=S_SCALE
      DO 19 I=1,NP
19    WRITE(7)XA(I)*X,YA(I)*X,R_A(I)*X,THA(I),XB(I)*X,
     X YB(I)*X,R_B(I)*X,THB(I),ALP(I)
      IF(SP$.EQ."P") THEN
      WRITE(1,119)FF_P$//EMP$//"      "//TITL$
119   FORMAT(1X,A)
      WRITE(1,119)" "
      WRITE(1,119)MES$
      WRITE(1,120)"CENTRAL FIELD =",CF," TESLAS"
120   FORMAT(1X,A,E10.4,A)
      WRITE(1,121)"RADIUS AND LENGTH OF CURRENT CYLINDER = ",S_SCALE,
     X " AND ",H*S_SCALE," METERS"
121   FORMAT(1X,A,F6.3,A,F6.3,A)
      WRITE(1,122)"AVERAGE RADIUS OF SHIELD INNER SURFACE = ",
     X SA*S_SCALE," METERS"
122   FORMAT(1X,A,F8.5,A)
      WRITE(1,122)"AVERAGE RADIUS OF SHIELD OUTER SURFACE = ",
     X SB*S_SCALE," METERS"
      WRITE(1,123)"SHIELD THICKNESS = ",THICK*100*S_SCALE,
     X " CENTIMETERS"
123   FORMAT(1X,A,F10.5,A)
      WRITE(1,1230)"TOTAL VOLUME OF SHIELD = ",VOL," CUBIC METERS"
1230  FORMAT(1X,A,E10.4,A)
125   FORMAT(1X,A,I3)
      WRITE(1,125)"NUMBER OF LEGENDRE TERMS = ",NT
126   FORMAT(1X,A,I2,A)
      WRITE(1,126)" THIS AVERAGES ",NAVE," TIMES BEFORE SOLVING"
127   FORMAT(1X,A,F5.2,A,F8.2)
      WRITE(1,127)"    THIS IS FOR GAP(DEGREES)= ",TH_MU0*180/PI,
     X ", RELATIVE MU= ",MU_1
      WRITE(1,119)"OUTPUT FILE = "//FOUT$//
     X "     MAGNITUDE OF B IS IN TESLAS"
      WRITE(1,119)STEMP$//" "
      END IF
C     THIS IS THE MAIN COMPUTATIONAL PART.  THE MATRICES A AND B
C     ARE FILLED IN WITH THE COEFFICIENTS REPRESENTING THE MATCHING
C     OF THE NORMAL AND (1/MU)*TANGENTIAL B COMPONENTS ACROSS THE
C     SHIELD SURFACES.  THIS PROCEDURE IS DONE JAVE TIMES AND THE
C     RESULTING COEFFICENT VALUES ARE AVERAGED
      CALL TIME(TIMEY$)
      IF(SP$.EQ."P") WRITE(1,119)"START PREP AT "//TIMEY$
      DO 302 I=1,4*NT0
      B(I)=0
      DO 302 J=1,4*NT0
```

```
302   A(I,J)=0
      S1=(1+H*H/4)**.5
      Z0=H/(2*S1)
      CALL MAKEP(NT+1,Z0,QMN)!MAKEP MAKES THE LEGENDRE FUNCTIONS FOR COS(TH)
C     EQUAL SECOND ARGUMENT, AND STORES RESULT IN LAST ARGUMENT ARRAY
      DO 30 JAVE =0,NAVE-1
      DO 30 I=1,NT,2
      II=(I+1)/2
      XIAVE=II-FLOAT(JAVE)/FLOAT(NAVE)
      TH=PI*XIAVE/(2.*NT0+2.)
      R=SRA(TH)  ! THIS R, TH DESCRIBE THE "DISCRETE" POINT ON THE INNER
C     SHIELD SURFACE
      STH=SIN(TH)
      CTH=COS(TH)
      Z1=0
      IF(R.GT.1) Z1=(1-(1/(R*R)))**.5
      CALL MAKEP(NT+1,CTH,PMN)
      CALL MAKEP(NT+1,Z1,RMN)
      CALL CYLSRC(NT,R,BR0,BTH0,Z0,Z1,S1,PMN,QMN,RMN)
C     CYLSRC MAKES FIELD DUE TO CURRENT CYLINDER, WITH 2PI*I=1
      BR0=BR0*CRNTF  ! CORRECT TO CURRENT ENTERED
      BTH0=BTH0*CRNTF
      NRI=NRA(TH)
      NTHI=NTHA(TH)
      B(II)=-NRI*BR0-NTHI*BTH0+B(II)
      B(II+NT0)=-NRI*BTH0+NTHI*BR0+B(II+NT0)
      B(II+2*NT0)=0
      B(II+3*NT0)=0  ! THESE ENTER THE INHOMOGENEOUS PART OF FIELDS, DUE
C     TO CURRENT SOURCE
      MUI=1/MU(TH)
      DO 30 J=1,NT,2
      JJ=(J+1)/2
      R=SRA(TH)
      NRI=NRA(TH)
      NTHI=NTHA(TH)
      PJI=PMN(0,J)
      P1JI=PMN(1,J)
      RAUP=(R/SA)**(J-1)
      RADN=(R/SA)**(-J-2)
      RB=(R/SB)**(J-1)
C     NOW MATCHING ACROSS INNER SURFACE
      A(II,JJ)=RAUP*(PJI*NRI-P1JI*NTHI/FLOAT(J))+A(II,JJ)
      A(II,JJ+NT0)=RADN*(-PJI*NRI-P1JI*NTHI/FLOAT(J+1))+A(II,JJ+NT0)
      A(II,JJ+2*NT0)=RB*(-PJI*NRI+P1JI*NTHI/FLOAT(J))+A(II,JJ+2*NT0)
      A(II,JJ+3*NT0)=0
      A(II+NT0,JJ)=RAUP*(-PJI*NTHI-P1JI*NRI/FLOAT(J))+A(II+NT0,JJ)
      A(II+NT0,JJ+NT0)=MUI*RADN*(PJI*NTHI-P1JI*NRI/FLOAT(J+1))+
     X A(II+NT0,JJ+NT0)
      A(II+NT0,JJ+2*NT0)=MUI*RB*(PJI*NTHI+P1JI*NRI/FLOAT(J))+
     X A(II+NT0,JJ+2*NT0)
      A(II+NT0,JJ+3*NT0)=0
C     NOW ON OUTER SURFACE
      R=SRB(TH)
      NRI=NRB(TH)
      NTHI=NTHB(TH)
      RA=(R/SA)**(-J-2)
      RBUP=(R/SB)**(J-1)
      RBDN=(R/SB)**(-J-2)
      A(II+2*NT0,JJ)=0
      A(II+2*NT0,JJ+NT0)=RA*(PJI*NRI+P1JI*NTHI/FLOAT(J+1))+
```

```
     X A(II+2*NT0,JJ+NT0)
      A(II+2*NT0,JJ+2*NT0)=RBUP*(PJI*NRI-P1JI*NTHI/FLOAT(J))+
     X A(II+2*NT0,JJ+2*NT0)
      A(II+2*NT0,JJ+3*NT0)=RBDN*(-PJI*NRI-P1JI*NTHI/FLOAT(J+1))+
     X A(II+2*NT0,JJ+3*NT0)
      A(II+3*NT0,JJ)=0
      A(II+3*NT0,JJ+NT0)=RA*MUI*(-PJI*NTHI+P1JI*NRI/FLOAT(J+1))+
     X A(II+3*NT0,JJ+NT0)
      A(II+3*NT0,JJ+2*NT0)=RBUP*MUI*(-PJI*NTHI-P1JI*NRI/FLOAT(J))+
     X A(II+3*NT0,JJ+2*NT0)
      A(II+3*NT0,JJ+3*NT0)=RBDN*(PJI*NTHI-P1JI*NRI/FLOAT(J+1))+
     X A(II+3*NT0,JJ+3*NT0)
30    CONTINUE
      N=4*NT0
1313  CALL TIME(TIMEY$)
      WRITE(*,*) "START GAUSSJ "//TIMEY$
      IF(SP$.EQ."P") WRITE(1,119)"START GAUSSJ AT "//TIMEY$
      CALL GAUSSJ(A,N,100,B,1,1)
C     THIS IS THE LINEAR EQUATION SOLVER FROM NUMRCP
      CALL TIME(TIMEY$)
      WRITE(*,*) "END GAUSSJ "//TIMEY$
      IF(SP$.EQ."P") WRITE(1,119)"END GAUSSJ AT "//TIMEY$
C     CARL H. BRANS   1/13/87
C     THIS PART PUTS SOLVED COEFFICENTS INTO C21, ETC, THEN PRINTS
C     RESULT. AS NOTED BELOW, IT COULD BE INCORPORATED INTO A
C     PROGRAM WHICH READS DATA FROM FILE, THEN PRINTS RESULTS
      DO 31 I=1,NT,2
      II=(I+1)/2
      C21(II)=B(II)
      C12(II)=B(II+NT0)
      C22(II)=B(II+2*NT0)
      C13(II)=B(II+3*NT0)
31    WRITE(7)C21(II),C12(II),C22(II),C13(II)
C     THIS TRANSFERS FROM SOLVED B TO C MATRICES
      IF(SP$.EQ."P") THEN
      NFILE=1                    ! PRINTER
      ELSE
      NFILE=6                    ! TERMINAL
      ENDIF
      WRITE(NFILE,131)
131   FORMAT(" R(METERS)",T12,"THETA",T20,"MAGNITUDE OF B",
     X T36,"UNSHIELDED B",T50,"SHIELDING FACTOR")
C     THE FOLLOWING COULD BE INCORPORATED INTO A PROGRAM TO PRINT
C     FIELD FROM PRE-COMPUTED DATA IN FILE 7.  HOWEVER, CARE MUST BE
C     TAKEN TO GET ALL NECESSARY INFO FROM FILE:H,SA,SB,NT,CRNT
C     ALSO, H,R1,R2,DELTA, ETC ARE ASSUMED TO BE IN SCALE, I.E.,
C     METERS/S_SCALE
C     THE NEXT FOUR LINES RECOMPUTE DATA TO ALLOW THIS PART TO BE
C     STANDALONE
      MU0=4*PI*10.**(-7)        ! STANDARD VALUE IN NEWTON/AMP**2
      CRNTF=CRNT*MU0/2
      S1=(1+H*H/4)**.5
      Z0=H/(2*S1)
      CALL MAKEP(NT+1,Z0,QMN)
      DO 32 NTH=1,NRTH!PRINT RESULTS
      TH=NTH*PI/(2.*NRTH+2.)
      STH=SIN(TH)
      CTH=COS(TH)
      CALL MAKEP(NT+1,CTH,PMN)
      DO 32 R=R1,R2,DELTA
```

```
      BR=0
      BTH=0
      Z1=0
      IF(R.GT.1) Z1=(1-1/(R*R))**.5
      CALL MAKEP(NT+1,Z1,RMN)
      CALL CYLSRC(NT,R,BR0,BTH0,Z0,Z1,S1,PMN,QMN,RMN)
      BR0=BR0*CRNTF
      BTH0=BTH0*CRNTF
      B0=(BR0*BR0+BTH0*BTH0)**.5         ! SAVE FOR SHIELD FACTOR
      IF(R.LT.SRA(TH)) THEN
      BR=BR0
      BTH=BTH0
      END IF
      DO 33 I=1,NT,2
      II=(I+1)/2
      IF(R.LT.SRA(TH)) THEN
      RR=(R/SA)**(I-1)
      BR=BR+C21(II)*RR*PMN(0,I)
      BTH=BTH-C21(II)*RR*PMN(1,I)/FLOAT(I)
      ELSE
      IF(R.LT.SRB(TH)) THEN
      RRA=(R/SA)**(-I-2)
      RRB=(R/SB)**(I-1)
      BR=BR+(C12(II)*RRA+C22(II)*RRB)*PMN(0,I)
      BTH=BTH+(C12(II)*RRA/FLOAT(I+1)-C22(II)*RRB/FLOAT(I))*PMN(1,I)
      ELSE
      RRB=(R/SB)**(-I-2)
      BR=BR+C13(II)*RRB*PMN(0,I)
      BTH=BTH+C13(II)*RRB*PMN(1,I)/FLOAT(I+1)
      END IF
      END IF
33    CONTINUE
      BB=(BR*BR+BTH*BTH)**.5
      WRITE(NFILE,133)R*S_SCALE,TH*180/PI,BB,B0,BB/B0
133   FORMAT(1X,F7.3,T10,F6.3,T23,E9.3,T38,E9.3,T52,E9.3)
32    CONTINUE
      CLOSE(7)
      GOTO 1                  ! MAIN LOOP
9999  IF(G_M.NE.0)CALL MW$TURNOFF(1,1)
      STOP
      END

      SUBROUTINE ERRS(X,Y)
C     WRITES ERROR MESSAGE,
C     GETS NEW X,Y, THEN BLANKS ERROR MESSAGE AND OLD SQUARE
C     AT X,Y
      CHARACTER*40 C
      INTEGER*2 MX,MY,XR(4)
      X0=X
      Y0=Y
      CALL MW$LOCAT(1.,7.,0)
      C='UNACCEPTABLE POINT. RE-ENTER'
      CALL DRAWSTRING(C)
      XX=X
      YY=Y
      CALL GETXY(XX,YY)
      X=XX
      Y=YY
      X0=X0-.05
      Y0=Y0-.05
```

```
      CALL MW$GETCS(MX,MY,X0,Y0)
      XR(1)=MX
      XR(2)=MY
      X0=X0+.1
      Y0=Y0+.1
      CALL MW$GETCS(MX,MY,X0,Y0)
      XR(3)=MX
      XR(4)=MY
      CALL ERASERECT(XR)
      CALL MW$LOCAT(1.,7.,0)
      C='                                                   '
      CALL DRAWSTRING(C)
      CALL MW$LOCAT(XX,YY,0)
      RETURN
      END

      FUNCTION F_ATAN(DX,DY)
C     COMPUTES ATAN(DX/DY) WITH VALUE IN RANGE 0 TO PI
C     ALSO WORKS FOR DY=0
      PI=ATAN(1.)*4
      IF(DY.EQ.0) THEN
      TH=PI/2
      ELSE
      TH=ATAN(DX/DY)
      ENDIF
      IF(TH.LT.0)TH=TH+PI
      IF(DX.LT.0.AND.DY.LT.0) TH=TH+PI
      F_ATAN=TH
      RETURN
      END

C     FORTRAN VERSION OF TBASIC SUBS FOR SHIELDIN, FLUXPLOT
C     GAUSSJ IS TAKEN DIRECTRLY FROM BOOK, NUMERICAL RECIPES, BY
C     PRESS ET AL, SEE PAGE 28 FOR DOCUMENTATION AND DISCUSSION
C     CHANGED HERE TO SET NMAX=100 AND HANDLE INTERRUPTS DEPENDING
C     ON GRAPH MODE ON/OFF THRU SWITCH, G_M
      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
      PARAMETER (NMAX=100)
      DIMENSION A(NP,NP),B(NP,MP),IPIV(NMAX),INDXR(NMAX),INDXC(NMAX)
      COMMON/GMODE/G_M!FOR GRAPH MODE ON/OFF
      DO 11 J=1,N
      IPIV(J)=0
11    CONTINUE
      DO 22 I=1,N
      BIG=0.
      DO 13 J=1,N
      IF(IPIV(J).NE.1)THEN
      DO 12 K=1,N
      IF (IPIV(K).EQ.0) THEN
      IF (ABS(A(J,K)).GE.BIG)THEN
      BIG=ABS(A(J,K))
      IROW=J
      ICOL=K
      ENDIF
      ELSE IF (IPIV(K).GT.1) THEN
C     PAUSE 'SINGULAR MATRIX'
      GOTO 9999
      ENDIF
12    CONTINUE
      ENDIF
```

```
13      CONTINUE
        IPIV(ICOL)=IPIV(ICOL)+1
        IF (IROW.NE.ICOL) THEN
        DO 14 L=1,N
        DUM=A(IROW,L)
        A(IROW,L)=A(ICOL,L)
        A(ICOL,L)=DUM
14      CONTINUE
        DO 15 L=1,M
        DUM=B(IROW,L)
        B(IROW,L)=B(ICOL,L)
        B(ICOL,L)=DUM
15      CONTINUE
        ENDIF
        INDXR(I)=IROW
        INDXC(I)=ICOL
        IF (A(ICOL,ICOL).EQ.0.) GOTO 9999   ! PAUSE 'SINGULAR MATRIX.'
        PIVINV=1./A(ICOL,ICOL)
        A(ICOL,ICOL)=1.
        DO 16 L=1,N
        A(ICOL,L)=A(ICOL,L)*PIVINV
16      CONTINUE
        DO 17 L=1,M
        B(ICOL,L)=B(ICOL,L)*PIVINV
17      CONTINUE
        DO 21 LL=1,N
        IF(LL.NE.ICOL)THEN
        DUM=A(LL,ICOL)
        A(LL,ICOL)=0.
        DO 18 L=1,N
        A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
18      CONTINUE
        DO 19 L=1,M
        B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19      CONTINUE
        ENDIF
21      CONTINUE
22      CONTINUE
        DO 24 L=N,1,-1
        IF(INDXR(L).NE.INDXC(L))THEN
        DO 23 K=1,N
        DUM=A(K,INDXR(L))
        A(K,INDXR(L))=A(K,INDXC(L))
        A(K,INDXC(L))=DUM
23      CONTINUE
        ENDIF
24      CONTINUE
        RETURN
9999    IF(G_M.EQ.1) CALL MW$TURNOFF(1,1)
        WRITE(*,*)"SINGULAR MATRIX IN GAUSSJ"
        STOP
        END

C       START OF MAKEP SUBROUTINE
C       8/30/87 FORTRAN VERSION
C       OUTPUTS PMN(M,L) AS LEGENDRE FUNCTION, P (SUPER M) (SUB L) AS FUNCTION
C       OF Z, FOR L=0 THRU N
        SUBROUTINE MAKEP(N,Z,PMN)
        DIMENSION PMN(0:1,0:50)
        PMN(0,0)=1
```

```
      PMN(1,0)=0
      PMN(0,1)=Z
      PMN(1,1)=(1-Z*Z)**.5
      PMN(0,2)=(3.*Z*Z-1)/2
      PMN(1,2)=3*Z*(1-Z*Z)**.5
      IF( N.GT.2) THEN
      DO 1 K=3,N
      DO 2 MM=0,1
      X=K-MM
      PMN(MM,K)=((2*K-1)*Z*PMN(MM,K-1)-(K+MM-1)*PMN(MM,K-2))/X
2     CONTINUE
1     CONTINUE
      END IF
      RETURN
      END

C     START OF SUBROUTINE GFNC
C     8/13/87 FORTRAN VERSION.
C     RETURNS G AS VALUE OF FUNCTION DEFINED IN NOTES 1/7/87 FOR USE
C     IN COMPUTATION OF FIELD PRODUCED BY CYLINDER SOURCE
      FUNCTION GFNC(M,N,Z,PMN,R)
      DIMENSION PMN(0:1,0:50)
      RR_G=R*R
      G=0
      IF (N.GE.1) THEN
      IF ((M.EQ.2) .OR. (N.GT.1)) THEN
      IF (M.EQ.1) THEN
      G=-((((1-Z*Z)*RR_G)**(N/2.))/R)*PMN(1,N-1)/FLOAT(N-1)
      ELSE
      G=((((1-Z*Z)*RR_G)**((-N-1.)/2.))/R)*PMN(1,N+1)/FLOAT(N+2)
      END IF
      ELSE
      G=Z
      END IF
      END IF
      GFNC=G
      RETURN
      END

C     START OF SUBROUTINE CYLSRC
C     8/13/87  FORTRAN VERSION
C     OUTPUTS BR, BTH  WHICH ARE PROPORTIONAL TO THE R, THETA COMPONENTS
C     OF MAGNETIC FIELD, BR,BTH AT R,TH, FOR CYLINDER SOURCE.  MULTIPLICATVE
C     CONSTANT NEEDED IS MU0*I/2, WHERE MU0=4*PI*10**(-7), I=CURRENT/LENGTH
C     IN AMP/METERS.  HERE COORDINATES ARE SCALED SO RADIUS OF CYLINDER
C     IS ONE, AND IN THESE UNITS, LENGTH IS H.
C     ALSO REQUIRES OTHER ARGUMENTS, INCLUDING LEGENDRE POLYNOMIALS TO HAVE
C     BEEN CALCULATED ALREADY
      SUBROUTINE CYLSRC(NT,R,BR,BTH,Z0,Z1,S1,PMN,QMN,RMN)
      DIMENSION PMN(0:1,0:50),QMN(0:1,0:50),RMN(0:1,0:50)
      BR=0
      BTH=0
      IF (R.NE.0) THEN
      DO 1 N=1,NT,2
      IF (R.LE.1) THEN
      G1N= GFNC(1,N,Z0,QMN,R)
      BR=BR+2*G1N*PMN(0,N)
      BTH=BTH-(2./N)*G1N*PMN(1,N)
      ELSE
      IF (R.LE.S1) THEN
```

```
      G1N0=GFNC(1,N,Z0,QMN,R)
      G1N1=GFNC(1,N,Z1,RMN,R)
      G2N1=GFNC(2,N,Z1,RMN,R)
      BR=BR+2*(G1N0-G1N1+G2N1)*PMN(0,N)
      BTH=BTH-((2./N)*(G1N0-G1N1)-(2./(N+1.))*G2N1)*PMN(1,N)
      ELSE
      G2N=GFNC(2,N,Z0,QMN,R)
      BR=BR+2*G2N*PMN(0,N)
      BTH=BTH-(-2./(N+1.))*G2N*PMN(1,N)
      END IF
      END IF
1     CONTINUE
      END IF
      RETURN
      END

C     FOLLOWING DIRECTLY FROM NUMERICAL RECIPES BOOK, PAGE 114
C     MODIFIED TO INCLUDE SWITCH, G_M, TO HANDLE ERROR TRAPS FOR
C     GRAPH MODE ON/OFF
      SUBROUTINE QROMB(FUNC,A,B,SS)
      EXTERNAL FUNC
      PARAMETER(EPS=1.E-6,JMAX=20,JMAXP=JMAX+1,K=5,KM=4)
      DIMENSION S(JMAXP),H(JMAXP)
      COMMON/GMODE/G_M                    !  GRAPH OFF/ON
      H(1)=1.
      DO 11 J=1,JMAX
      CALL TRAPZD(FUNC,A,B,S(J),J)
      IF (J.GE.K) THEN
      L=J-KM
      CALL POLINT(H(L),S(L),K,0.,SS,DSS)
      IF (ABS(DSS).LT.EPS*ABS(SS)) RETURN
      ENDIF
      S(J+1)=S(J)
      H(J+1)=0.25*H(J)
11    CONTINUE
      IF(G_M.EQ.1) CALL MW$TURNOFF(1,1)
      PAUSE 'TOO MANY STEPS.'
      END

C     POLINT FROM NUMERICAL RECIPES BOOK, PAGE 82
      SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
      PARAMETER (NMAX=10)
      DIMENSION XA(N),YA(N),C(NMAX),D(NMAX)
      NS=1
      DIF=ABS(X-XA(1))
      DO 11 I=1,N
      DIFT=ABS(X-XA(I))
      IF (DIFT.LT.DIF) THEN
      NS=I
      DIF=DIFT
      ENDIF
      C(I)=YA(I)
      D(I)=YA(I)
11    CONTINUE
      Y=YA(NS)
      NS=NS-1
      DO 13 M=1,N-1
      DO 12 I=1,N-M
      HO=XA(I)-X
      HP=XA(I+M)-X
```

```
      W=C(I+1)-D(I)
      DEN=HO-HP
      IF(DEN.EQ.0.)PAUSE
      DEN=W/DEN
      D(I)=HP*DEN
      C(I)=HO*DEN
12    CONTINUE
      IF (2*NS.LT.N-M)THEN
      DY=C(NS+1)
      ELSE
      DY=D(NS)
      NS=NS-1
      ENDIF
      Y=Y+DY
13    CONTINUE
      RETURN
      END

C     TRAPZD FROM NUMERICAL RECIPES BOOK, PAGE 111
      SUBROUTINE TRAPZD(FUNC,A,B,S,N)
      EXTERNAL FUNC
      IF (N.EQ.1) THEN
      S=0.5*(B-A)*(FUNC(A)+FUNC(B))
      IT=1
      ELSE
      TNM=IT
      DEL=(B-A)/TNM
      X=A+0.5*DEL
      SUM=0.
      DO 11 J=1,IT
      SUM=SUM+FUNC(X)
      X=X+DEL
11    CONTINUE
      S=0.5*(S+(B-A)*SUM/TNM)
      IT=2*IT
      ENDIF
      RETURN
      END

C     FOR STRAIGHT LINE SEGMENTS
C     THE FOLLOWIN SET OF FUNCTIONS PROVIDE THE RADIUS AS A
C     FUNCTION OF THETA FOR INNER SURFACE, SRA, AND FOR OUTER, SRB
C     NORMALS ALONG R,THETA UNIT VECTORS, ARE NRA,NTHA (INNER) AND
C     NRB, NTHB (OUTER).  ALSO, MU(TH) PROVIDES MU AS A FUNCTION OF
C     THETA CORRESPONDING TO SHIELD GAP. THESE ARE ALL REAL.
      FUNCTION SRA(TH)
      DIMENSION FA(0:19),FB(0:19)
      REAL MU_0,MU_1,MU
      COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
      REAL NRA,NTHA,NRB,NTHB,ITH_S,MU,MU_0,MU_1
C     XA,YA ARE INNER POINTS ENTERED
C     XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C     RA,THA,RB,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
C     A(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),A(20)
      REAL RA(20),RB(20)
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,A,RA,RB,NP
C     THA(1)=THB(1) IS GAP ANGLE.  IF TH<THA(1) FILL OUT SHIELD WITH
C     LINES PARALLEL TO FIRST ONE, ANGLE A(1).  SOME SHAPE IS NEEDED
```

```
C     FOR MATCHING ACROSS SHIELD SURFACE IN GAP ANGLE, WHERE ABSENCE
C     OF SHIELD IS ACCOUNTED FOR BY SET MU=1 THERE IN FUNCTION MU(TH)
      DO 1 I=1,NP
      IF(TH.LT.THA(I)) GOTO 2
1     CONTINUE
C     IF TH>LAST ENTERED ANGLE, USE A(NP), WHICH HAS BEEN
C     SET=PI TO MAKE VERTICAL INTERSECTION WITH X-AXIS
      I=NP+1
2     IF(I.GT.1)I=I-1
      X_S=RA(I)*SIN(A(I)-THA(I))/SIN(A(I)-TH)
9     SRA=X_S
      RETURN
      ENTRY SRB(TH)
      DO 11 I=1,NP
      IF(TH.LT.THB(I)) GOTO 21
11    CONTINUE
      I=NP+1
21    IF(I.GT.1)I=I-1
      X_S=RB(I)*SIN(A(I)-THB(I))/SIN(A(I)-TH)
91    SRB=X_S
      RETURN
      ENTRY NRA(TH)
      DO 13 I=1,NP
      IF(TH.LT.THA(I)) GOTO 23
13    CONTINUE
      I=NP+1
23    IF(I.GT.1)I=I-1
      X_S=RA(I)*SIN(A(I)-THA(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=(1.+(DRA/X_S)**2)**(-.5)
93    NRA=X_S
      RETURN
      ENTRY NTHA(TH)
      DO 14 I=1,NP
      IF(TH.LT.THA(I)) GOTO 24
14    CONTINUE
      I=NP+1
24    IF(I.GT.1)I=I-1
      X_S=RA(I)*SIN(A(I)-THA(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=-(DRA/X_S)*(1.+(DRA/X_S)**2)**(-.5)
94    NTHA=X_S
      RETURN
      ENTRY NRB(TH)
      DO 15 I=1,NP
      IF(TH.LT.THB(I)) GOTO 25
15    CONTINUE
      I=NP+1
25    IF(I.GT.1)I=I-1
      X_S=RB(I)*SIN(A(I)-THB(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=(1.+(DRA/X_S)**2)**(-.5)
95    NRB=X_S
      RETURN
      ENTRY NTHB(TH)
      DO 16 I=1,NP
      IF(TH.LT.THB(I)) GOTO 26
16    CONTINUE
      I=NP+1
26    IF(I.GT.1)I=I-1
```

```
      X_S=RB(I)*SIN(A(I)-THB(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=-(DRA/X_S)*(1.+(DRA/X_S)**2)**(-.5)
96    NTHB=X_S
      RETURN
      ENTRY MU(TH)
      IF(TH.LE.TH_MU0) THEN
      MU=MU_0
      ELSE
      MU=MU_1
      END IF
      RETURN
      END

      SUBROUTINE OUTER(IERR)
C     FORTRAN VERSION 11/29/87
C     FOR STRAIGHTLINE SEGMENT SHIELDS
C     INPUT IS XA,YA,RA,THA,A AND OUTPUT IS XB,YB,RB,THB
C     OUTER STRAIGHT LINE SEGMENTS ARE AT DISTANCE T FROM
C     INNER ONES.
C     XA,YA ARE INNER POINTS ENTERED
C     XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C     RA,THA,RB,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
C     A(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),A(20)
      REAL RA(20),RB(20),MU_0,MU_1
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,A,RA,RB,NP
      DIMENSION FA(0:19),FB(0:19)
      COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
      IERR=0    ! ZERO ERROR
C     FIRST IS SPECIAL
      I=1
      THB(I)=THA(I)
      SAT=SIN(A(I)-THA(I))
      TT=THICK/SAT
      RB(I)=RA(I)+TT
      YB(I)=YA(I)+TT*COS(THA(I))
      XB(I)=XA(I)+TT*SIN(THA(I))
      DO 2 I=1,NP-1
      SAT=SIN(A(I)-A(I+1))
      IF(SAT.EQ.0) THEN
      DX=-THICK*COS(A(I))
      DY=THICK*SIN(A(I))
      ELSE
      DX=THICK*(SIN(A(I+1))-SIN(A(I)))/SAT
      DY=THICK*(COS(A(I+1))-COS(A(I)))/SAT
      ENDIF
      XB(I+1)=XA(I+1)+DX
      YB(I+1)=YA(I+1)+DY
      RB(I+1)=(XB(I+1)**2+YB(I+1)**2)**.5
2     THB(I+1)=F_ATAN(XB(I+1),YB(I+1))
C     CHECK FOR ERROR
      DO 4 I=1,NP
      IF(XB(I).LT.0.OR.YB(I).LT.0) GO TO 913
      IF(I.GT.1) THEN
      IF(THB(I).LT.THB(I-1)) GOTO 913
      ENDIF
4     CONTINUE
      RETURN
```

```
913   IERR=1
      RETURN
      END

C     FORTRAN VERSION 8/13/87
C     CARL H. BRANS    12/31/86
C     FUNC IS THE FUNCTION USED BY THE INTEGRATION ROUTINE, QROMB,
C     WHICH IS CALLED TO INTEGRATE ALONG THE SPHERICAL ANGLE THETA
C     TO PRODUCE THE VOLUME OF THE SHIELD BETWEEN THE RADII SRA(TH)
C     AND SRB(TH)
      FUNCTION FUNC(X)
      X_F=SRA(X)
      Y_F=SRB(X)
      FUNC=(Y_F**3-X_F**3)*SIN(X)
      RETURN
      END

      SUBROUTINE GRAPH(S_G,H_G)
C     GRAPH PREPARES GRAPHICS:  AXES, SCALES, ETC.
C     LAHEY FORTRAN VERSION, 8/14/87
      CHARACTER*37 MSG,C1*2
      ASP=1.
      CALL MW$TURNON(1,ASP)
      CALL MW$AXES(-1.,12.,-1.,9.,1.,1.)
      DO 1 I=1,11
      X=I
      CALL MW$LOCAT(X-.2,-.8,0)
      WRITE(C1,100)I
100   FORMAT(I2)
1     CALL DRAWSTRING(C1)
      IF(H_G.GT.0) THEN
      DO 2 X=.98,1.02,.01
      CALL MW$LOCAT(X,0.,0)
2     CALL MW$LOCAT(X,H_G/2,1)
      ENDIF
      WRITE(MSG,101)S_G
101   FORMAT('NOTE: EACH AXES UNIT IS ',F5.3,' METERS')
      CALL MW$LOCAT(1.,8.5,0)
      CALL DRAWSTRING(MSG)
      RETURN
      END

      SUBROUTINE GETXY(X,Y)
C     FOR SHIELDIN  8/29/87
C     GETS MOUSE POINT WHEN LEFT BUTTON IS DEPRESSED AND DRAWS
C     CIRCLE AROUND POINT, SHOWS UP AS SQUARE, BECAUSE OF
C     DISCRETE ROUNDOFF
      INTEGER*2 MX,MY,BUTTONS,SR(4)
      REAL OTHER(9)
      COMMON/MWGR/SR,OTHER
      CALL MW$GETCS(MX,MY,X,Y)
      CALL MOVECURSOR(MX,MY)!LOCATE TO LAST POINT, SO ON FIRST CALL SET X,Y
      CALL SHOWCURSOR
1     CALL READMOUSE(MX,MY,BUTTONS)
      IF(BUTTONS.LT.0) GOTO 1!NO CHANGE
      CALL MOVECURSOR(MX,MY)
      IF(BUTTONS.NE.4) THEN
      CALL MOVETO(MX,MY)
      GOTO 1
      ENDIF
```

```
C     BUTTONS=4, LEFT BUTTON
      CALL MOVETO(MX,MY)
      CALL HIDECURSOR
C     DRAW SMALL CIRCLE
      PI=4.*ATAN(1.)
      CALL MW$GETXY(MX,MY,X,Y)!CONVERTS TO REAL X,Y
      IF=0
      DO 2 TH=0,2*PI,.1
      XX=X+.05*COS(TH)
      YY=Y+.05*SIN(TH)
      CALL MW$LOCAT(XX,YY,IF)
2     IF=1
      CALL MW$LOCAT(X,Y,0)
      RETURN
      END
```

```
      PROGRAM FLUX
C
C     FLUX.FOR
C     5/25/88
C     FINAL ALTERATIONS BY CHRIS MILTENBERGER
C     COMPILE AND LINK WITH FMG2.BAT
C
C     ------------------------------------------------------------------
C
C     DELIVERED TO ACE 12/7/87
C     VERSION TO GO WITH OUTPUT FROM SHIELDN2, STRAIGHT LINE
C     SEGMENT SHIELDS.
C     CHECKS INPUT FILE FOR FIRST FIELD='SHIELDN2' TO CONFIRM
C     IT WAS OUTPUT BY SHIELDN2
C     COMPILE AND LINK WITH FMG2.BAT
C
C     FOLLOWING NOTES PRECEDE 12/5/87
C     CORRECTED 11/18/87 TO RESCALE THICK WITH S_SCALE
C     DELIVERED TO ACE 9/13/87
C
C     9/13/87 CORRECTED APPARENT INCONSISTENCIES WITH TBASIC VERSION
C     FOR SHIELDIN.  NO MAJOR CHANGES IN THIS, EXCEPT FOR INTRODUCTORY
C     GRAPH-PRINT OPTION MESSAGE
C
C     DELIVERY TO ACE 9/3/87, WITH PROBLEMS IN SHIELDIN: INCONSISTENT
C     WITH TBASIC VERSION
C
C     FORTRAN F77L VERSION 8/31/87
C
C     DELIVERED TO ACE 7/24/87
C
C     THIS IS COMPANION PROGRAM TO SHIELDIN AND WAS
C     MODIFIED ON 7/24/87 WHEN SHIELDIN WAS
C
C     MAJOR MODIFICATION TO THIS IS TO ALLOW USER OPTION
C     TO PRINT UNSHIELDED FLUX LINES.  ALSO USER IS ASKED
C     TO HIT MOUSE BUTTON/RETURN AFTER GRAPHICAL DISPLAY IN
C     ORDER TO HAVE TIME TO VIEW DISPLAY AND/OR "GRAB" IT
C     FOR DRHALO
C
C     -->  FOLLOWING MODIFIED 7/24/87
C     MODIFIED 3/25/87 LIBRARY "NUMRCP" TO "NUMRCP.TRC"
C     AS NEEDED FOR RUN VERSION
C
C     CARL H. BRANS    1/1/87
C     READS DATA FROM FILE FIN$, OUTPUTTED BY SHIELDIN, AND
C     PLOTS FLUX LINES, WITH R INCREMENTS OF DEL, STARTING FROM
C     CIRCLE OF RADIUS R0, OUT TO R1, FOR NTH THETA INCREMENTS
C
C     THIS PART DEFINES DATA, GETS INPUT CHOICES
C     FOR DATA DEFINITIONS AND FURTHER DOCUMENTATION SEE SHIELDIN.FOR
C
C     XA,YA ARE INNER POINTS ENTERED
C     XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C     R_A,THA,R_B,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
C     ALP(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),ALP(20)
      REAL R_A(20),R_B(20)
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,ALP,R_A,R_B,NP
```

```
      LOGICAL OKI_PRINT!NOT NEEDED AT ACE
      DIMENSION C21(50),C12(50),C22(50),C13(50)
      DIMENSION PMN(0:1,0:50),QMN(0:1,0:50),RMN(0:1,0:50)
      REAL MUI,MU0,MU,MU_0,MU_1,NRA,NTHA,NRB,NTHB,NRI,NTHI
      DIMENSION FA(0:19),FB(0:19)
      CHARACTER*11 S_U$,FIN$*20,S$*70,ODATE$*8,OTIME$*8,CH*1,X$*8
      COMMON/GMODE/G_M
      COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
      OPTION BREAK(*9999)
      OKI_PRINT=.FALSE.
      CALL GETCL(X$)
      IF(X$.EQ.' OKI') OKI_PRINT=.TRUE.
      OPEN(1,'CON',ACCESS='TRANSPARENT')
      OPEN(2,'LPT1')
      PI=4.*ATAN(1.)
1     CONTINUE
C     DO!THIS IS THE PROGRAM MASTER LOOP, NOT ENDED TILL END OF FP3
      CLOSE(7)!INPUT FILE
C     CLEAR
      WRITE(*,*)' FLUXPLT2 F77L VERSION 12/5/87 '
      WRITE(*,*)' PLOTS FLUX LINES FOR DATA PREPARED BY SHIELDN2'
      WRITE(*,*)' '
      WRITE(*,*)' AFTER GRAPH DISPLAY, PROGRAM PAUSES TO ALLOW USER'
      WRITE(*,*)' TO VIEW GRAPH AND DUMP TO PRINTER IF WANTED. HIT'
      WRITE(*,*)' ANY KEY TO CONTINUE AFTER SUCH PAUSE.'
      IF(OKI_PRINT) WRITE(*,*) ' >>> THIS IS SET TO DUMP GRAPHICS',
     X ' TO OKIDATA 192 PRINTER.'
      WRITE(*,*)' '
      G_M=0
1314  WRITE(*,*)"ENTER NAME OF INPUT FILE: "
      READ(*,*)FIN$
      WRITE(*,*)"ENTER S FOR SHIELDED FLUX LINES, U FOR UNSHIELDED."
C     S_U$="X"
C     DO WHILE S_U$ <> "U" AND S_U$ <> "S"
2     WRITE(*,*) "ENTER S OR U: "
      READ(*,*)S_U$
      IF(S_U$.NE."U".AND.S_U$.NE."S") GO TO 2
      OPEN(7,FILE=FIN$,STATUS='OLD',FORM='UNFORMATTED',ERR=1314)
C     7 IS DATA FILE
      READ(7)X$
      IF(X$.NE.'SHIELDN2') THEN
      WRITE(*,*) 'DATA FILE NOT PREPARED BY THIS SYSTEM.'
      GOTO 1
      ENDIF
      READ(7)ODATE$,OTIME$,NP,CRNT,H,NT,TH_MU0,MU_1,SA,SB,
     X S_SCALE,THICK
      X=S_SCALE
      DO 719 I=1,NP
      READ(7)XA(I),YA(I),R_A(I),THA(I),XB(I),
     X YB(I),R_B(I),THB(I),ALP(I)
      XA(I)=XA(I)/X
      YA(I)=YA(I)/X
      XB(I)=XB(I)/X
      YB(I)=YB(I)/X
      R_A(I)=R_A(I)/X
719   R_B(I)=R_B(I)/X
      TH_MU0=THA(1)
      DO 3 I=1,NT,2
      II=((I+1)/2)
3     READ(7)C21(II),C12(II),C22(II),C13(II)
```

```
      CLOSE(7)
      THICK=THICK/S_SCALE
      H=H/S_SCALE
      SA=SA/S_SCALE
      SB=SB/S_SCALE
      MU0=4*PI*10.**(-7)!STANDARD VACUUM VALUE IN NEWTON/AMP**2
      CRNTF=CRNT*MU0/2
      S1=(1+H*H/4)**.5
      Z0=H/(2*S1)
      CALL MAKEP(NT+1,Z0,QMN)
      WRITE(*,*) "FLUX LINES WILL BE PLOTTED "//
     X "STARTING AT R0 THRU R1, STEP DELR"
      WRITE(*,*) "FOR THETA FROM TH0 THRU TH1, "//
     X "STEP DELTH (ANGLES IN DEGREES)"
      WRITE(*,*) "NOTE: ACCURACY OF FLUX LINE PLOTS "//
     X "INCREASES WITH DECREASING DELR"
      WRITE(*,*) "ENTER LENGTHS IN METERS"
      WRITE(*,*) "ENTER R0,R1,DELR,TH0,TH1,DELTH: "
      READ(*,*)R0,R1,DEL,TH0P,TH1P,DELTH
      R0=R0/S_SCALE
      R1=R1/S_SCALE
      DEL=DEL/S_SCALE
      TH0P=TH0P*PI/180.
      TH1P=TH1P*PI/180.
      DELTH=DELTH*PI/180.
      CALL GRAPH(S_SCALE,H)
      G_M=1
      IF (S_U$.EQ."S") THEN
      S_U$=" SHIELDED"
      ELSE
      S_U$=" UNSHIELDED"
      MU_1=1
      END IF
      S$="FLUXPLT2, INPUT FILE: "//CHARNB(FIN$)
     X //", "//ODATE$//", "//OTIME$
      CALL MW$LOCAT(2.,8.,0)
      CALL DRAWSTRING(S$)
      WRITE(S$,101)NP,H*S_SCALE,MU0*CRNT,NT
101   FORMAT("NP=",I2,", H=",F5.2,", CENTRAL FIELD=",
     X E10.4,", NT=",I2)
      CALL MW$LOCAT(2.,7.5,0)
      CALL DRAWSTRING(S$)
      WRITE(S$,102)TH_MU0*180./PI,MU_1,DEL*S_SCALE,S_U$
102   FORMAT("TH_MU0=",F4.1,", MU=",I5,', STEP SIZE=',F7.5,A)
      CALL MW$LOCAT(2.,7.,0)
      CALL DRAWSTRING(S$)
      DO 4 TH=TH_MU0,PI/2,.01
      RR1=SRA(TH)
      RR2=SRB(TH)
      STH=SIN(TH)
      CTH=COS(TH)
      IF(TH.EQ.TH_MU0) THEN
      CALL MW$LOCAT(RR1*STH,RR1*CTH,0)
      CALL MW$LOCAT(RR2*STH,RR2*CTH,1)
      ELSE
      CALL MW$LOCAT(RR1*STH,RR1*CTH,0)
      CALL MW$LOCAT(RR1*STH,RR1*CTH,1)!MAKE POINT
      CALL MW$LOCAT(RR2*STH,RR2*CTH,0)
      CALL MW$LOCAT(RR2*STH,RR2*CTH,1)!MAKE POINT
      END IF
```

```
4      CONTINUE!PLOT OF SHIELD SURFACE
       CALL MW$LOCAT(RR2*STH,RR2*CTH,0)
C      THIS PART PLOTS LINES OF FLUX FROM NORMALIZED FIELD VECTORS
       DO 5 THP=TH0P,TH1P,DELTH
       TH=THP
       R=R0
       X=R*SIN(TH)
       Y=R*COS(TH)
       R=0
       COUNT=0
       LIN=0
60     CONTINUE
C      DO WHILE R<R1 AND X>=0 AND Y>=0 AND COUNT<12/DEL
       IF(R.GE.R1.OR.X.LE..01.OR.Y.LE..01.OR.COUNT.GE.12/DEL)
      X GOTO 5
       COUNT=COUNT+1
       R=(X*X+Y*Y)**.5
       IF(Y.NE.0) THEN
       TH=ATAN(X/Y)
       ELSE
       TH=PI/2
       END IF
       STH=SIN(TH)
       CTH=COS(TH)
       CALL MAKEP(NT+1,CTH,PMN)
       BR=0
       BTH=0
       Z1=0
       IF(R.GT.1) Z1=(1-1/(R*R))**.5
       IF(S_U$.EQ." UNSHIELDED") THEN
       CALL MAKEP(NT+1,Z1,RMN)
       CALL CYLSRC(NT,R,BR,BTH,Z0,Z1,S1,PMN,QMN,RMN)
       BR=BR*CRNTF
       BTH=BTH*CRNTF
       ELSE
       IF(R.LT.SRA(TH)) THEN
       CALL MAKEP(NT+1,Z1,RMN)
       CALL CYLSRC(NT,R,BR,BTH,Z0,Z1,S1,PMN,QMN,RMN)
       BR=BR*CRNTF
       BTH=BTH*CRNTF
       END IF
       DO 7 I=1,NT,2
       II=((I+1)/2)
       IF(R.LT.SRA(TH)) THEN
       RR=(R/SA)**(I-1)
       BR=BR+C21(II)*RR*PMN(0,I)
       BTH=BTH-C21(II)*RR*PMN(1,I)/FLOAT(I)
       ELSE
       IF(R.LT.SRB(TH)) THEN
       RRA=(R/SA)**(-I-2)
       RRB=(R/SB)**(I-1)
       BR=BR+(C12(II)*RRA+C22(II)*RRB)*PMN(0,I)
       BTH=BTH+(C12(II)*RRA/FLOAT(I+1)-C22(II)*RRB/FLOAT(I))*
      X PMN(1,I)
       ELSE
       RRB=(R/SB)**(-I-2)
       BR=BR+C13(II)*RRB*PMN(0,I)
       BTH=BTH+C13(II)*RRB*PMN(1,I)/FLOAT(I+1)
       END IF
       END IF
```

```
C       NEXT I
7       CONTINUE
        END IF
        BX=(BR*STH+BTH*CTH)
        BY=(BR*CTH-BTH*STH)
        BB=(BX*BX+BY*BY)**.5
        X=X+(BX/BB)*DEL
        Y=Y+(BY/BB)*DEL
        R=(X*X+Y*Y)**.5
        CALL MW$LOCAT(X,Y,LIN)
        LIN=1
C       LOOP
        GOTO 60
C       NEXT THP
5       CONTINUE
        READ(1,105)CH!ALLOW TIME TO GRAB
105     FORMAT(A)
        IF(CH.EQ.'P'.AND.OKI_PRINT) CALL MW$PRINT(2)!ONLY FOR OKIDATA
        CALL MW$TURNOFF(1,1)
        G_M=0
        GOTO 1
C       LOOP! END OF MASTER LOOP STARTED IN FP3
9999    IF(G_M.NE.0) CALL MW$TURNOFF(1,1)
        STOP
        END

        FUNCTION F_ATAN(DX,DY)
C       COMPUTES ATAN(DX/DY) WITH VALUE IN RANGE 0 TO PI
C       ALSO WORKS FOR DY=0
        PI=ATAN(1.)*4
        IF(DY.EQ.0) THEN
           TH=PI/2
        ELSE
           TH=ATAN(DX/DY)
        ENDIF
        IF(TH.LT.0)TH=TH+PI
        IF(DX.LT.0.AND.DY.LT.0) TH=TH+PI
        F_ATAN=TH
        RETURN
        END
```

```
      PROGRAM ANGLE
C     FILE IS ANGLE.FOR
C     5/25/88
C     MODIFIED BY CHRIS MILTENBERGER
C     THIS PROGRAM MAKES THE SAME FIELD COMPUTATIONS AS SHIELD.FOR, BUT THE
C     CALCULATIONS ARE FOR A SINGLE ANGLE ENTERED BY THE USER.
C
C     TO LINK, TYPE =>   C:\F77L>LINK ANGLE+MWSUBS,SHIELD,NUL,WNDOWF77
C
C     ----------------------------------------------------------------------
C
C     DELIVERED TO ACE 12/7/87
C     COMPILE AND LINK WITH FMG2, WHICH LINKS WITH NUMRCP2 AND MWSUBS
C
C     THIS IS THE VERSION FOR STRAIGHT LINE SEGMENTS FOR
C     SHIELD SHAPES.
C
C     IN THIS VERSION, USER ENTERS SHIELD SHAPES IN TERMS OF
C     STRAIGHT LINE SEGMENTS, DEFINED BY UP TO 20 POINTS.
C     ENTRY CAN BE EITHER BY POINTS DEFINED BY MOUSE ON SCREEN
C     OR THRU DIRECT ENTRY OF X,Y COORDINATES.  IN EITHER CASE
C     USER ENTERS POINTS DEFINING INNER SHIELD, STORED AS CARTESIAN
C     COORDINATES IN XA,YA, FROM WHICH RADIUS AND ANGLE, R_A, THA
C     ARE COMPUTED.  THE ANGLE THE ITH LINE MAKES WITH RESPECT TO
C     THE Y-AXIS IS STORED AS ALP(I).
C     THE FOLLOWING CONDITIONS MUST BE MET DURING DATA ENTRY,
C     OR DATA IS REJECTED:
C             EACH X>=0, Y>=0, THA(I+1)>=THA(I), ALP(I)<=PI
C     THUS, INFORMALLY, DATA MUST BE ENTERED IN CLOCKWISE MANNER MAKING
C     CONVEX SET WITH NO RE-ENTRIES.
C     THE FIRST POINT ENTERED IS TAKEN TO DETERMINE THE GAP ANGLE.
C     NOTE: THIS DIFFERS FROM PREVIOUS, SHIELDIN, WHERE GAP ANGLE
C     WAS ENTERED SEPARATELY.
C
C     THE OUTER SHIELD POINTS, XB,YB, ARE DEFINED TO BE ON
C     INTERSECTION OF LINES PARALLEL TO INNER ONES, BUT SEPARATED
C     BY DISTANCE EQUAL TO USER ENTERED THICK.
C     SEE HANDWRITTEN NOTES:  FMAGS GEOMETRY, 11/29/87.
C
C     NOTE: FIRST POINT ON OUTER SURFACE IS ON RADIUS THRU FIRST POINT
C     ON INNER SURFACE, I.E., THB(1)=THA(1)=GAP ANGLE
C
C     -->  FOLLOWING PRECEDE 12/5/87
C     ALL LENGTHS ARE STORED INTERNALLY(FOR GRAPHICS DISPLAY)
C     IN UNITS OF CURRENT SOURCE CYLINDER RADIUS, S_SCALE
C
C     DELIVERED TO ACE 9/13/87
C     PREVIOUS DISAGREEMENTS WITH TBASIC VERSION WERE FOUND TO
C     DUE TO EVALUATION OF MU(TH) EXACTLY AT SHIELD EDGE.
C     TBASIC VS F77L PRODUCE DIFFERENT VALUES BECAUSE THEY HANDLE
C     ROUNDOFFS AND VALUE OF PI DIFFERENTLY.  THIS INCONSISTENCIES
C     IS ELIMINATED BY INCREASING NUMBER OF AVERAGES UP TO 10, WHICH
C     IS CURRENT RECOMMENDED VALUE.
C
C     -->  FOLLOWING REPLACE 9/13/87
C     TEST VERSION MAILED TO ACE 9/3/87.  DISAGREES WITH TBASIC
C     VERSION IN VALUES OF FIELD!!!!  EITHER ERROR IN CONVERSION
C     OR DISAGREEMENT IN NUMERICAL CALCULATION !!!!
C
C     CONVERSION FROM .TRU VERSION 9/3/87
```

```
C     AS OF 9/3/87, AFTER GRAPH DISPLAY, WILL PAUSE, AWAITNG
C     CHARACTER IN UNIT 10 (CONSOLE, TRANSPARENT).
C     ENTRY OF P WILL CAUSE A CALL TO MW$PRINT(1), UNIT1=PRINT
C     ONLY IF -1313 HAS BEEN ENTERED FOR FIRST NUMBER(SHIELD RADIUS).
C     THIS IS CODE ACTIVATTING OKIDATA PRINT ROUTINE. SWITCH IS
C     LOGICAL OKI_PRINT.
C     IN DELIVERY TO ACE, THIS IS NOT NEEDED, BECAUSE THEY
C     HAVE SATISFACTORY DUMP-TO-PRINT PROCEDURE OF THEIR OWN.
C
C     -->  MODIFIED 7/24/87 AS FOLLOWS:
C     <<<<<<  NOTE 12/5/87  >>>>>>
C     THIS PARAGRAPH 1 IS OBSOLETE FOR SHIELDN2.  SEE ABOVE
C     1) IF USER ENTERS SHIELD SHAPE GRAPHICALLY USING MOUSE,
C        HE ONLY ENTERS INNER SHIELD SURFACE AND THICKNESS.
C        SYSTEM THEN COMPUTES FOURIER SERIES TO APPROXIMATE
C        OUTER SURFACE AT NORMAL DISTANCE=THICKNESS.  WARNING:
C        IF INNER SURFACE IS TOO CURVED, OUTER SURFACE SO
C        GENERATED MAY BE AT A NORMAL DISTANCE SIGNIFICANTLY
C        DIFFERENT FROM THE ENTERED THICKNESS IN CERTAIN REGIONS.
C        THIS INVOLVES NEW SUBROUTINE OUTER BASED ON HANDWRITTEN
C        NOTES: CONSTANT NORMAL THICKNESS REGION 7/20/87
C
C     2) USER ENTERS CURRENT PER UNIT LENGTH IN "CENTRAL FIELD"
C        UNITS (TESLAS), EQUIVALENT TO MU0*CURRENT/LENGTH.
C
C     3) PRINTED OUTPUT INCLUDES UNSHIELDED FIELD STRENGTH.
C
C     4) FOLLOWING GRAPHICAL DISPLAYS, USER IS ASKED TO HIT
C        MOUSE BUTTON/RETURN TO CONTINUE.  THIS GIVES HIM TIME
C        LOOK AT DISPLAY AND, IF DESIRED, USE ALTPRTSCRN TO
C        "GRAB" IMAGE FOR DRHALO.  MOUSE POINTER CAN BE MOVED
C        OFF SCREEN IF DESIRED.
C
C     -->  FOLLOWING REPLACED 7/24/87
C     THESE WERE NOTES FOR TBASIC VERSION
C     MODIFIED 3/26/87 TO GET .EXE VERSION, BY CHANGING
C     LIBRARY "NUMRCP" TO LIBRARY "NUMRCP.TRC"
C     CARL H. BRANS    1/8/87
C     THIS IS THE FINAL DELIVERY VERSION TO ACE OF PROGRAM
C     WHICH PRODUCES AN ESTIMATE TO THE CONSTANTS INVOLVED IN
C     THE LEGENDRE EXPANSION IN SPHERICAL COORDINATES OF THE
C     MAGNETIC FIELD DUE TO A CYLINDRICAL SOURCE, SURROUNDED
C     BY A SHIELD WHOSE SURFACES ARE INPUT BY USER.  THE SHIELD
C     MAY ALSO HAVE A VACUUM GAP DEFINED BY A USER ENTERED ANGLE
C     WITH RESPECT TO THE Z-AXIS.  THE PROBLEM IS ASSUMED TO HAVE
C     FULL SYMMETRY UNDER ROTATIONS ABOUT THE Z-AXIS AS WELL AS
C     INVERSION ABOUT THE Z-AXIX, I.E., Z --> -Z.  THE APPROACH
C     USED IN THIS PROGRAM MAKES USE OF THE LEGENDRE EXPANSION
C     WITH DIFFERENT COEFFICIENTS IN EACH OF THE THREE REGIONS:
C     R<SRA(TH), SRA(TH)<=R<SRB(TH), AND R>SRB(TH), WHERE THE
C     INNER AND OUTER SHIELD SURFACES ARE DEFINED BY THE FUNCTIONS
C     SRA(TH), SRB(TH).  THE USUAL MAGNETOSTATIC CONTINUITY
C     AND CONDITIONS AT INFINITY THEN IMPOSE SUFFICIENT EQUATIONS
C     TO DETERMINE THESE COEFFICIENTS.  HOWEVER, IF THE SHIELD
C     SURFACES ARE NOT SPHERICAL, THESE CONDITIONS CANNOT BE
C     SOLVED EXPLICITLY IN CLOSED FORM.  THE APPROXIMATION OF
C     THIS PROGRAM CONSISTS IN KEEPING ONLY A FINITE NUMBER OF
C     TERMS IN THE LEGENDRE EXPANSION AND THEN IMPOSING THE
C     MATCHING CONTINUITY CONDITIONS AND ONLY A DISCRETE NUMBER
C     OF POINTS, SUFFICIENT TO FULLY DETERMINE THE NOW FINITE
```

```
C     NUMBER OF LEGENDRE COEFFICIENTS.  THIS IS EXPLAINED IN DETAIL
C     IN THE NOTES: "ACE MAGNETIC SHIELDING PROGRAM",1/7/87 AND
C     "DISCRETE SHIELDING, SYMMETRIC", 12/10/86.
C
C     THIS PROGRAM MAKES USE OF SUBROUTINES, CONTAINED IN COMPILED
C     LIBRARY, NUMRCP.  THEY INCLUDE THE FOLLOWING.
C
C     GAUSSJ   FROM NUMERICAL RECIPES FOR SOLVING LINEAR EQUATION SET
C     MAKEP    MAKES LEGENDRE FUNCTIONS
C     GFNC     FUNCTION USED CYLSRC
C     CYLSRC   SUBROUTINE TO COMPUTE FREE (UNSHIELDED) FIELD FROM
C              CYLINDRICAL SOURCE
C     QROMB, TRAPZD, POLINT, FROM NUMERICAL RECIPES, FOR DOING NUMERICAL
C              VOLUME INTEGRAL FOR SHIELD
C     SRA,SRB,NRA,NTHA,NRB,NTHB FUNCTIONS GIVING RADIUS AND NORMAL
C              COMPONENTS OF SHIELD
C     MU       FUNCTION GIVING (REAL) MU AS FUNCTION OF THETA.  VALUE IS
C              MU_0 IN GAP, ELSE MU_1. THIS IS RELATIVE MU
C     OUTER    FINDS OUTER SHIELD SHAPE FROM INNER ONE AND USER ENTERED
C              THICKNESS
C     FUNC     FUNCTION USED IN NUMERICAL INTEGRATION, QROMB, FOR THE
C              SHIELD VOLUME.
C     GRAPH    SETS UP GRAPHICS MODE
C     GETXY    GETS MOUSE X,Y COORDINATES
C
C     IN ADDITION, GRAPHICS NEEDS META WINDOW SUPPORT, INCLUDING
C     F77L SUBROUTINES IN MWSUBS.  THUS, AFTER F77L COMPILE,
C
C     LINK MWSUBS+SHIELDIN,SHIELDIN,NUL,WNDOWF77
C
C     THIS SECTION IS CONCERNED WITH BACKGROUND DATA DEFINITION,
C     GETTING THE SHAPE OF THE SHIELD SURFACES.  IT ALSO STARTS THE
C     PROGRAM'S MASTER LOOP, SO THAT PROGRAM CAN BE RE-RUN.
C
C     FOR MATHEMATICAL ANALYSIS AND FURTHER INFORMATION ON NOTATION
C     SEE HANDWRITTEN NOTES:
C
C        SPHERICAL MAG SHIELD   8/20/86
C        DISCRETE SHIELDING, SYMMETRIC 12/10/86
C        MAGNETIC SHIELDING PROGRAM 1/7/87
C        SPHERICAL SHIELDING FACTOR 1/12/87
C        FMAGS GEOMETRY 11/29/87
C
C     XA,YA ARE INNER POINTS ENTERED
C     XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C     R_A,THA,R_B,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
C     ALP(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),ALP(20)
      REAL R_A(20),R_B(20)
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,ALP,R_A,R_B,NP
      LOGICAL OKI_PRINT!CONTROLS CALL TO MW$PRINT NOT USED AT ACE
      CHARACTER*1 FF_P$,EMP$*2,STEMP$*2,FN$*20,FOUT$*20
      DIMENSION A(100,100),B(100),C21(50),C12(50),C22(50),C13(50)
      DIMENSION PMN(0:1,0:50),QMN(0:1,0:50),RMN(0:1,0:50)
C     PMN,QMN,RMN HOLD THE LEGENDRE POLYNOMIALS EVALUATED AT VARIOUS
C     ANGLES
      DIMENSION FA(0:19),FB(0:19)
      REAL MUI,MU0,MU,MU_0,MU_1,NRA,NTHA,NRB,NTHB,NRI,NTHI
      COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
```

```
      CHARACTER*70 TITL$,MES$,SP$*1,DATEX*8,TIMEX*8,TIMEY$*8,XP$*1
      CHARACTER*8 X$,Y$,MES2$*70,MES3$*70
      CHARACTER*70 CH$,MSG*50
      COMMON/GMODE/G_M!G_M=0/1 FOR GRAPH OFF/ON
      EXTERNAL FUNC
      OPTION BREAK(*9999)
      OKI_PRINT=.FALSE.
      CALL GETCL(X$)
      IF(X$.EQ.' OKI') OKI_PRINT=.TRUE.
      FF_P$=CHAR(12)!PRINTER FORM FEED
      EMP$=CHAR(27)//CHAR(69)!PRINTER EMPHASIZES
      STEMP$=CHAR(27)//CHAR(70)!STOP PRINTER EMPHASIS
C     A AND B ARE THE MATRICES DEFINING THE LINEAR EQUATION
C
C        AX = B
C
C     TO BE SOLVED FOR X BY SUBROUTINE GAUSSJ.
C     THE MATRICES C21,C12,C22,C13 ARE THE COEFFICIENTS IN THE
C     LEGENDRE EXPANSION OF THE FIELD IN THE THREE REGIONS, AS
C     EXPLAINED IN THE NOTES: "DISCRETE SHIELDING, SYMMETRIC", 12/10/86
      TITL$="ALABAMA CRYOGENIC ENGINEERING  MAGNETIC SHIELDING PROGRAM"
      CALL DATE(DATEX)
      CALL TIME(TIMEX)
      MES$="SHIELDN2 F77L VERSION 12/5/87, RUN AT: "//DATEX//
     X ", "//TIMEX
      MES2$="THIS VERSION USES STRAIGHT LINE SEGMENTS TO"
      MES3$="DEFINE SHIELD SURFACES"
      SP$=" "
      PI=4.*ATAN(1.)
      OPEN(UNIT=10,FILE='CON',ACCESS='TRANSPARENT')
C     10=CONSOLE FOR USER CONTINUATION AFTER GRAPH DISPLAY PAUSE
      OPEN(UNIT=1,FILE='LPT1')
C     1=PRINTER
1     CONTINUE
      G_M=0!GRAPH MODE OFF
C     THIS IS THE BEGINNING OF THE PROGRAMS'S MASTER LOOP
      CLOSE(2)
      CLOSE(7)
      WRITE(*,*) " "
      WRITE(*,*) "    ",TITL$
      WRITE(*,*) "    ",MES$
      WRITE(*,*) "    ",CHARNB(MES2$)//" "//MES3$
      WRITE(*,*) " "
      WRITE(*,*)' '
      WRITE(*,*)' AFTER GRAPH DISPLAY, PROGRAM PAUSES TO ALLOW USER'
      WRITE(*,*)' TO VIEW GRAPH AND DUMP TO PRINTER IF WANTED. HIT'
      WRITE(*,*)' ANY KEY TO CONTINUE AFTER SUCH PAUSE.'
      IF(OKI_PRINT) WRITE(*,*)' >>>  THIS IS SET TO DUMP GRAPHICS',
     X ' TO OKIDATA 192 PRINTER.'
      WRITE(*,*)' '
C     ENTER SOURCE CYLINDER PARAMETER
      WRITE(*,*) "ENTER SOURCE CYLINDER RADIUS IN METERS: "
1001  READ(*,*)S_SCALE
C     S_SCALE IS USED INTERNALLY FOR ALL LENGTHS, SO THAT THEY ARE
C     IMMEDIATELY AVAILABLE FOR GRAPH.  HOWEVER, WHEN DATA IS
C     OUTPUT IN NUMERIC FORM, EACH LENGHT MUST BE MULTIPLIED
C     BY S_SCALE
C     S_SCALE WILL SET THE SCALE OF PLOTS AND INTERNAL CALCULATIONS
C     SEE NOTES: "MAGNETIC SHIELDING PROGRAM 1/7/87"
      WRITE(*,*) "ENTER SOURCE CYLINDER TOTAL LENGTH IN METERS: "
```

```
      READ(*,*)H
      H=H/S_SCALE
C     GET SHAPES OF SHIELD SURFACES
      WRITE(*,*)
      WRITE(*,*) "SHIELD SURFACES WILL NOW BE DEFINED.  ENTER 1 TO "//
     X "ENTER NEW SHAPES, OR"
      WRITE(*,*) "2 TO USE THE SHAPES DEFINED BY A PREVIOUS RUN AND "//
     X "CONTAINED IN A KNOWN FILE"
      WRITE(*,*) " "
      WRITE(*,*) "1. NEW SHAPE"
      WRITE(*,*) "2. OLD SHAPE"
      N=0
2     WRITE(*,*)  "ENTER CHOICE (1 OR 2): "
      READ(*,*)N
      IF(N.NE.1.AND.N.NE.2) GOTO 2
      FN$=" "                               ! FN$=FILE NAME
      IF(N.EQ.2) THEN
      WRITE(*,*) "ENTER FILE NAME: "
      READ(*,100)FN$
100   FORMAT(A)
      END IF
      IF(FN$.EQ." ") THEN!                              --- IF FN$ START
      NP=0!NP=NUMBER OF USER ENTERED POINTS
      WRITE(*,*) "SHIELD SURFACES CAN BE DEFINED BY DIRECT "//
     X "ENTRY OF COORDINATES,"
      WRITE(*,*) "OR BY MOVING CURSOR TO A NUMBER OF POINTS, "//
     X "WHICH WILL THEN"
      WRITE(*,*) "DEFINE STRAIGHT LINE SEGMENTS.  CHOOSE 1 TO "//
     X "ENTER COORDINATES DIRECTLY,"
      WRITE(*,*) "OR 2 TO ENTER POINTS"
      WRITE(*,*) " "
      WRITE(*,*) "1. ENTER COEFFICIENTS"
      WRITE(*,*) "2. ENTER POINTS ON GRAPH"
      NC=0
4     WRITE(*,*) "ENTER YOUR CHOICE (1 OR 2): "
      READ(*,*)NC
      IF(NC.LT.1.OR.NC.GT.2) GOTO 4
6     WRITE(*,*)'ENTER NUMBER OF POINTS(LESS THAN 21): '
      READ(*,*)NP
      IF(NP.GT.20) GO TO 6
      WRITE(*,*)"ENTER THICKNESS IN CENTIMETERS: "
      READ(*,*)THICK
      THICK=THICK/S_SCALE
      THICK=THICK/100                      ! CARRY IT IN METERS INTERNALLY
      IF(NC.EQ.2) GO TO 500
103   DO 401 I=1,NP
601   WRITE(*,*)'X,Y: '
      READ(*,*)X,Y
      XA(I)=X
      YA(I)=Y
      IF(X.LT.0.OR.Y.LT.0) THEN
      WRITE(*,*)' CANNOT BE NEGATIVE. RE-ENTER'
      GOTO 601
      ENDIF
      IF(I.GT.1) ALP(I-1)=F_ATAN(XA(I)-XA(I-1),YA(I)-YA(I-1))
      THA(I)=F_ATAN(XA(I),YA(I))
      IF(I.GT.1) THEN
      IF(THA(I).LT.THA(I-1).OR.ALP(I-1).GT.PI)THEN
      WRITE(*,*)' INVALID. MUST BE CONVEX'
      GOTO 601
```

```
      ENDIF
      ENDIF
401   R_A(I)=(XA(I)**2+YA(I)**2)**.5
      ALP(NP)=PI
      TH_MU0=THA(1)
      X=TH_MU0*180/PI
      CALL OUTER(IERR)
      IF(IERR.NE.0) THEN
      WRITE(*,*)' ERROR: OUTER SHIELD POINTS ILLEGAL'
      WRITE(*,*)' EITHER NOT CONVEX, OR NEGATIVE X OR Y'
      WRITE(*,*)' '
      GOTO 103
      ENDIF
      CALL GRAPH(S_SCALE,H)
      G_M=1
      CALL MW$LOCAT(1.,8.,0)
      CALL DRAWSTRING(MSG)
      CALL MW$LOCAT(1.,7.5,0)
      WRITE(MSG,300)X
      CALL DRAWSTRING(MSG)
      GOTO 700
500   CALL GRAPH(S_SCALE,H)
      G_M=1
      CALL MW$LOCAT(1.,8.,0)
      CALL DRAWSTRING(MSG)
      TH=0
      X=0
      Y=0
      DO 701 I=1,NP
      CALL GETXY(X,Y)
101   XA(I)=X
      YA(I)=Y
      IF(X.LT.0.OR.Y.LT.0) THEN
      CALL ERRS(X,Y)
      GOTO 101
      ENDIF
      IF(I.GT.1) ALP(I-1)=F_ATAN(XA(I)-XA(I-1),YA(I)-YA(I-1))
      THA(I)=F_ATAN(XA(I),YA(I))
      IF(I.GT.1) THEN
      IF(THA(I).LT.THA(I-1).OR.ALP(I-1).GT.PI)THEN
      CALL ERRS(X,Y)
      GOTO 101
      ENDIF
      ENDIF
701   R_A(I)=(XA(I)**2+YA(I)**2)**.5
      ALP(NP)=PI
      TH_MU0=THA(1)
      X=TH_MU0*180/PI
      WRITE(MSG,300)X
300   FORMAT(' GAP ANGLE = ',F6.2,' DEGREES')
      CALL MW$LOCAT(1.,7.5,0)
      CALL DRAWSTRING(MSG)
      CALL OUTER(IERR)
      IF(IERR.NE.0) THEN
      CALL MW$TURNOFF(1,1)
      G_M=0
      WRITE(*,*)' ERROR: OUTER SHIELD POINTS ILLEGAL'
      WRITE(*,*)' EITHER NOT CONVEX, OR NEGATIVE X OR Y'
      WRITE(*,*)' '
      GOTO 103
```

```
      ENDIF
      ELSE!DATA FROM FILE                                        ----- FN$ ELSE
      OPEN(2,FILE=FN$,STATUS='OLD',FORM='UNFORMATTED')
C     2 IS DATA FILE
      READ(2)X$
      IF(X$.NE.'SHIELDN2') THEN
      WRITE(*,*) 'DATA FILE NOT PREPARED BY THIS SYSTEM.'
      GOTO 1
      ENDIF
      READ(2)X$,Y$,NP,X1,X2,X3,X5,X6,SA,SB,X7,THICK
      SA=SA/S_SCALE
      SB=SB/S_SCALE
      THICK=THICK/S_SCALE
      X=S_SCALE
      DO 719 I=1,NP
      READ(2)XA(I),YA(I),R_A(I),THA(I),XB(I),
     X YB(I),R_B(I),THB(I),ALP(I)
      XA(I)=XA(I)/X
      YA(I)=YA(I)/X
      XB(I)=XB(I)/X
      YB(I)=YB(I)/X
      R_A(I)=R_A(I)/X
719   R_B(I)=R_B(I)/X
      TH_MU0=THA(1)
      CLOSE(2)
      CALL GRAPH(S_SCALE,H)
      G_M=1!GRAPH MODE ON
      CALL MW$LOCAT(2.,8.,0)
      CH$="FROM FILE "//FN$
      CALL DRAWSTRING(CH$)
      END IF!END OF NEW OR OLD SHAPE IF                         ------  FN$ ENDIF
C     NOW PLOT CURVES FROM COEFFICIENTS
700   L=0
      DO 15 TH=TH_MU0,PI/2,.01
      SRS=SRA(TH)
      XS=SRS*SIN(TH)
      YS=SRS*COS(TH)
      CALL MW$LOCAT(XS,YS,L)
15    L=1
      L=0
      DO 16 TH=TH_MU0,PI/2,.01
      SRS=SRB(TH)
      XS=SRS*SIN(TH)
      YS=SRS*COS(TH)
      CALL MW$LOCAT(XS,YS,L)
16    L=1
      READ(10,116)XP$  ! GIVE USER CHANCE TO SEE OR GRAB
      IF(XP$.EQ."P".AND.OKI_PRINT)CALL MW$PRINT(1)    ! ONLY FOR OKIDATA
116   FORMAT(A1)
      CALL MW$TURNOFF(1,1)
      G_M=0!GRAPHMODE OFF
C     THIS PART CONTINUES ENTRY OF PARAMETERS.
      WRITE(*,'(A,F6.2)') " ANGLE OF GAP IN DEGREES: ",TH_MU0*180/PI
      WRITE(*,*) "ENTER RELATIVE MU "
      READ(*,*)MU_1
      WRITE(*,*) "COMPUTING VOLUME, PLEASE WAIT . . ."
      CALL QROMB(FUNC,TH_MU0,PI/2,SS)
C     THIS IS NUMERICAL INTEGRATION, SEE PART 3
      VOL=(S_SCALE**3)*SS*4*PI/3
      CALL GRAPH(S_SCALE,H)
```

```
      G_M=1!GRAPH MODE ON
      SA=0
      SB=0
C     SA,SB ARE AVERAGE VALUES OF RADIUS ON INNER AND OUTER SURFACES
C     USED IN SCALING POWERS OF R IN LEGENDRE EXPANSION
      N=0
      DO 17 TH=TH_MU0,PI/2,.01!SHOW VOLUME OF SHIELD
      R1=SRA(TH)
      R2=SRB(TH)
      STH=SIN(TH)
      CTH=COS(TH)
      CALL MW$LOCAT(R1*STH,R1*CTH,N)
      CALL MW$LOCAT(R2*STH,R2*CTH,1)
      SA=SA+R1
      SB=SB+R2
17    N=N+1
      SA=SA/N
      SB=SB/N
      WRITE(CH$,117)VOL
117   FORMAT("VOLUME=",E10.4," CUBIC METERS")
      CALL MW$LOCAT(2.,8.,0)
      CALL DRAWSTRING(CH$)
      READ(10,116)XP$          !   LET USER SEE
      IF(XP$.EQ.'P'.AND.OKI_PRINT) CALL MW$PRINT(1)   ! ONLY FOR OKIDATA
      CALL MW$TURNOFF(1,1)
      G_M=0!GRAPH OFF
      WRITE(*,*) "DATA OUTPUT FILE: "
      READ(*,*)FOUT$
      OPEN(7,FOUT$,FORM='UNFORMATTED')
C     7=OUTPUT FILE
18    WRITE(*,*) "SEND RESULT TO SCREEN(S)/PRINTER(P): "
      READ(*,*)SP$
      IF(SP$.NE.'P'.AND.SP$.NE.'S')GOTO 18
      MU_0=1    !  THIS IS MU VALUE IN GAP. USED BY FUNCTION MU(TH)
      WRITE(*,*) "NUMBER OF AVERAGES (SUGGEST 10) "
      READ(*,*)NAVE
C     THE PROCESS OF MATCHING THE FIELD AT A DISCRETE NUMBER OF POINTS
C     ACROSS SHIELD BOUNDARIES, THEN USING THE RESUTING EQUATIONS TO
C     DETERMINE THE EXPANSION COEFFICIENTS WILL BE REPEATED NAVE TIMES
C     AND THE RESULT AVERAGED.  THIS HAS THE ADVANTAGE OF SAMPLING THE
C     BOUNDARIES AT MORE POINTS, WITHOUT INCREASING THE NUMER OF UNKOWNS
C     TO BE SOLVED FOR.  EXECUTION TIME IS THUS LINEAR IN NAVE, BUT
C     APPROXIMATELY QUADRATIC IN NT
      WRITE(*,*)"NUMBER OF EXPANSION TERMS (MUST BE ODD, SUGGEST 9) "
      READ(*,*)NT
      NT=9
      WRITE(*,*) "NEXT, ENTER CENTRAL FIELD WHICH "//
     X "IS MU0 * CURRENT/LENGTH"
      WRITE(*,*) "ENTER CENTRAL FIELD(TESLAS): "
      READ(*,*)CF
      MU0=4*PI*10.**(-7)    ! STANDARD VALUE IN NEWTONS/AMP**2
      CRNT=CF/MU0
      CRNTF=CRNT*MU0/2   ! THIS MULTIPLIES OUTPUT OF CYLSRC
      WRITE(*,*) "PRINTED OUTPUT WILL BE ALONG RADII, FROM "//
     X "R1 TO R2, STEP DELTA"
      WRITE(*,*) "ENTER R1, R2, AND DELTA IN METERS: "
      READ(*,*)R1,R2,DELTA
      R1=R1/S_SCALE
      R2=R2/S_SCALE
      DELTA=DELTA/S_SCALE
```

```
C      WRITE(*,*)"RADII WILL BE FOR N DIVISIONS OF PI/2. ENTER N "
      WRITE(*,*)"ENTER THE ANGLE ALONG WHICH THE FIELD "//
     X "WILL BE CALCULATED"
      READ(*,*)ANGLE
C      READ(*,*)NRTH
      NT0=((NT+1)/2)
      IF(NT.EQ.NT0*2) THEN
      NT=NT+1
      NT0=((NT+1)/2)
      WRITE(*,*) "NT RAISED TO ODD NUMBER", NT
      END IF
      HS=H*S_SCALE
      SAS=SA*S_SCALE
      SBS=SB*S_SCALE
      THICK_S=THICK*S_SCALE
      X$='SHIELDN2'
      WRITE(7)X$
      WRITE(7)DATEX,TIMEX,NP,CRNT,HS,NT,TH_MU0,MU_1,SAS,SBS,
     X S_SCALE,THICK_S
      X=S_SCALE
      DO 19 I=1,NP
19    WRITE(7)XA(I)*X,YA(I)*X,R_A(I)*X,THA(I),XB(I)*X,
     X YB(I)*X,R_B(I)*X,THB(I),ALP(I)
      IF(SP$.EQ."P") THEN
      WRITE(1,119)FF_P$//EMP$//"     "//TITL$
119   FORMAT(1X,A)
      WRITE(1,119)" "
      WRITE(1,119)MES$
      WRITE(1,120)"CENTRAL FIELD =",CF," TESLAS"
120   FORMAT(1X,A,E10.4,A)
      WRITE(1,121)"RADIUS AND LENGTH OF CURRENT CYLINDER = ",S_SCALE,
     X " AND ",H*S_SCALE," METERS"
121   FORMAT(1X,A,F6.3,A,F6.3,A)
      WRITE(1,122)"AVERAGE RADIUS OF SHIELD INNER SURFACE = ",
     X SA*S_SCALE," METERS"
122   FORMAT(1X,A,F8.5,A)
      WRITE(1,122)"AVERAGE RADIUS OF SHIELD OUTER SURFACE = ",
     X SB*S_SCALE," METERS"
      WRITE(1,123)"SHIELD THICKNESS = ",THICK*100*S_SCALE,
     X " CENTIMETERS"
123   FORMAT(1X,A,F10.5,A)
      WRITE(1,1230)"TOTAL VOLUME OF SHIELD = ",VOL," CUBIC METERS"
1230  FORMAT(1X,A,E10.4,A)
125   FORMAT(1X,A,I3)
      WRITE(1,125)"NUMBER OF LEGENDRE TERMS = ",NT
126   FORMAT(1X,A,I2,A)
      WRITE(1,126)" THIS AVERAGES ",NAVE," TIMES BEFORE SOLVING"
127   FORMAT(1X,A,F5.2,A,F8.2)
      WRITE(1,127)"    THIS IS FOR GAP(DEGREES)= ",TH_MU0*180/PI,
     X ", RELATIVE MU= ",MU_1
      WRITE(1,119)"OUTPUT FILE = "//FOUT$//
     X "     MAGNITUDE OF B IS IN TESLAS"
      WRITE(1,119)STEMP$//" "
      END IF
C     THIS IS THE MAIN COMPUTATIONAL PART.  THE MATRICES A AND B
C     ARE FILLED IN WITH THE COEFFICIENTS REPRESENTING THE MATCHING
C     OF THE NORMAL AND (1/MU)*TANGENTIAL B COMPONENTS ACROSS THE
C     SHIELD SURFACES.  THIS PROCEDURE IS DONE JAVE TIMES AND THE
C     RESULTING COEFFICENT VALUES ARE AVERAGED
      CALL TIME(TIMEY$)
```

```
      IF(SP$.EQ."P") WRITE(1,119)"START PREP AT "//TIMEY$
      DO 302 I=1,4*NT0
      B(I)=0
      DO 302 J=1,4*NT0
302   A(I,J)=0
      S1=(1+H*H/4)**.5
      Z0=H/(2*S1)
      CALL MAKEP(NT+1,Z0,QMN)!MAKEP MAKES THE LEGENDRE FUNCTIONS FOR COS(TH)
C     EQUAL SECOND ARGUMENT, AND STORES RESULT IN LAST ARGUMENT ARRAY
      DO 30 JAVE =0,NAVE-1
      DO 30 I=1,NT,2
      II=(I+1)/2
      XIAVE=II-FLOAT(JAVE)/FLOAT(NAVE)
      TH=PI*XIAVE/(2.*NT0+2.)
      R=SRA(TH)
C     THIS R, TH DESCRIBE THE "DISCRETE" POINT ON THE INNER
C     SHIELD SURFACE
      STH=SIN(TH)
      CTH=COS(TH)
      Z1=0
      IF(R.GT.1) Z1=(1-(1/(R*R)))**.5
      CALL MAKEP(NT+1,CTH,PMN)
      CALL MAKEP(NT+1,Z1,RMN)
      CALL CYLSRC(NT,R,BR0,BTH0,Z0,Z1,S1,PMN,QMN,RMN)
C     CYLSRC MAKES FIELD DUE TO CURRENT CYLINDER, WITH 2PI*I=1
      BR0=BR0*CRNTF  ! CORRECT TO CURRENT ENTERED
      BTH0=BTH0*CRNTF
      NRI=NRA(TH)
      NTHI=NTHA(TH)
      B(II)=-NRI*BR0-NTHI*BTH0+B(II)
      B(II+NT0)=-NRI*BTH0+NTHI*BR0+B(II+NT0)
      B(II+2*NT0)=0
      B(II+3*NT0)=0
C     THESE ENTER THE INHOMOGENEOUS PART OF FIELDS, DUE
C     TO CURRENT SOURCE
      MUI=1/MU(TH)
      DO 30 J=1,NT,2
      JJ=(J+1)/2
      R=SRA(TH)
      NRI=NRA(TH)
      NTHI=NTHA(TH)
      PJI=PMN(0,J)
      P1JI=PMN(1,J)
      RAUP=(R/SA)**(J-1)
      RADN=(R/SA)**(-J-2)
      RB=(R/SB)**(J-1)
C     NOW MATCHING ACROSS INNER SURFACE
      A(II,JJ)=RAUP*(PJI*NRI-P1JI*NTHI/FLOAT(J))+A(II,JJ)
      A(II,JJ+NT0)=RADN*(-PJI*NRI-P1JI*NTHI/FLOAT(J+1))+A(II,JJ+NT0)
      A(II,JJ+2*NT0)=RB*(-PJI*NRI+P1JI*NTHI/FLOAT(J))+A(II,JJ+2*NT0)
      A(II,JJ+3*NT0)=0
      A(II+NT0,JJ)=RAUP*(-PJI*NTHI-P1JI*NRI/FLOAT(J))+A(II+NT0,JJ)
      A(II+NT0,JJ+NT0)=MUI*RADN*(PJI*NTHI-P1JI*NRI/FLOAT(J+1))+
     X A(II+NT0,JJ+NT0)
      A(II+NT0,JJ+2*NT0)=MUI*RB*(PJI*NTHI+P1JI*NRI/FLOAT(J))+
     X A(II+NT0,JJ+2*NT0)
      A(II+NT0,JJ+3*NT0)=0
C     NOW ON OUTER SURFACE
      R=SRB(TH)
      NRI=NRB(TH)
```

```
      NTHI=NTHB(TH)
      RA=(R/SA)**(-J-2)
      RBUP=(R/SB)**(J-1)
      RBDN=(R/SB)**(-J-2)
      A(II+2*NT0,JJ)=0
      A(II+2*NT0,JJ+NT0)=RA*(PJI*NRI+P1JI*NTHI/FLOAT(J+1))+
     X A(II+2*NT0,JJ+NT0)
      A(II+2*NT0,JJ+2*NT0)=RBUP*(PJI*NRI-P1JI*NTHI/FLOAT(J))+
     X A(II+2*NT0,JJ+2*NT0)
      A(II+2*NT0,JJ+3*NT0)=RBDN*(-PJI*NRI-P1JI*NTHI/FLOAT(J+1))+
     X A(II+2*NT0,JJ+3*NT0)
      A(II+3*NT0,JJ)=0
      A(II+3*NT0,JJ+NT0)=RA*MUI*(-PJI*NTHI+P1JI*NRI/FLOAT(J+1))+
     X A(II+3*NT0,JJ+NT0)
      A(II+3*NT0,JJ+2*NT0)=RBUP*MUI*(-PJI*NTHI-P1JI*NRI/FLOAT(J))+
     X A(II+3*NT0,JJ+2*NT0)
      A(II+3*NT0,JJ+3*NT0)=RBDN*(PJI*NTHI-P1JI*NRI/FLOAT(J+1))+
     X A(II+3*NT0,JJ+3*NT0)
30    CONTINUE
      N=4*NT0
1313  CALL TIME(TIMEY$)
      WRITE(*,*) "START GAUSSJ "//TIMEY$
      IF(SP$.EQ."P") WRITE(1,119)"START GAUSSJ AT "//TIMEY$
      CALL GAUSSJ(A,N,100,B,1,1)
C     THIS IS THE LINEAR EQUATION SOLVER FROM NUMRCP
      CALL TIME(TIMEY$)
      WRITE(*,*) "END GAUSSJ "//TIMEY$
      IF(SP$.EQ."P") WRITE(1,119)"END GAUSSJ AT "//TIMEY$
C     CARL H. BRANS   1/13/87
C     THIS PART PUTS SOLVED COEFFICENTS INTO C21, ETC, THEN PRINTS
C     RESULT. AS NOTED BELOW, IT COULD BE INCORPORATED INTO A
C     PROGRAM WHICH READS DATA FROM FILE, THEN PRINTS RESULTS
      DO 31 I=1,NT,2
      II=(I+1)/2
      C21(II)=B(II)
      C12(II)=B(II+NT0)
      C22(II)=B(II+2*NT0)
      C13(II)=B(II+3*NT0)
31    WRITE(7)C21(II),C12(II),C22(II),C13(II)
C     THIS TRANSFERS FROM SOLVED B TO C MATRICES
      IF(SP$.EQ."P") THEN
      NFILE=1                    ! PRINTER
      ELSE
      NFILE=6                    ! TERMINAL
      ENDIF
      WRITE(NFILE,131)
131   FORMAT(" R(METERS)",T12,"THETA",T20,"MAGNITUDE OF B",
     X T36,"UNSHIELDED B",T50,"SHIELDING FACTOR")
C     THE FOLLOWING COULD BE INCORPORATED INTO A PROGRAM TO PRINT
C     FIELD FROM PRE-COMPUTED DATA IN FILE 7.  HOWEVER, CARE MUST BE
C     TAKEN TO GET ALL NECESSARY INFO FROM FILE:H,SA,SB,NT,CRNT
C     ALSO, H,R1,R2,DELTA, ETC ARE ASSUMED TO BE IN SCALE, I.E.,
C     METERS/S_SCALE
C     THE NEXT FOUR LINES RECOMPUTE DATA TO ALLOW THIS PART TO BE
C     STANDALONE
      MU0=4*PI*10.**(-7)        ! STANDARD VALUE IN NEWTON/AMP**2
      CRNTF=CRNT*MU0/2
      S1=(1+H*H/4)**.5
      Z0=H/(2*S1)
      CALL MAKEP(NT+1,Z0,QMN)
```

```
C     DO 32 NTH=1,NRTH!PRINT RESULTS
C      TH=NTH*PI/(2.*NRTH+2.)
      TH=ANGLE*PI/180
      STH=SIN(TH)
      CTH=COS(TH)
      CALL MAKEP(NT+1,CTH,PMN)
      DO 32 R=R1,R2,DELTA
      BR=0
      BTH=0
      Z1=0
      IF(R.GT.1) Z1=(1-1/(R*R))**.5
      CALL MAKEP(NT+1,Z1,RMN)
      CALL CYLSRC(NT,R,BR0,BTH0,Z0,Z1,S1,PMN,QMN,RMN)
      BR0=BR0*CRNTF
      BTH0=BTH0*CRNTF
      B0=(BR0*BR0+BTH0*BTH0)**.5          ! SAVE FOR SHIELD FACTOR
      IF(R.LT.SRA(TH)) THEN
      BR=BR0
      BTH=BTH0
      END IF
      DO 33 I=1,NT,2
      II=(I+1)/2
      IF(R.LT.SRA(TH)) THEN
      RR=(R/SA)**(I-1)
      BR=BR+C21(II)*RR*PMN(0,I)
      BTH=BTH-C21(II)*RR*PMN(1,I)/FLOAT(I)
      ELSE
      IF(R.LT.SRB(TH)) THEN
      RRA=(R/SA)**(-I-2)
      RRB=(R/SB)**(I-1)
      BR=BR+(C12(II)*RRA+C22(II)*RRB)*PMN(0,I)
      BTH=BTH+(C12(II)*RRA/FLOAT(I+1)-C22(II)*RRB/FLOAT(I))*PMN(1,I)
      ELSE
      RRB=(R/SB)**(-I-2)
      BR=BR+C13(II)*RRB*PMN(0,I)
      BTH=BTH+C13(II)*RRB*PMN(1,I)/FLOAT(I+1)
      END IF
      END IF
33    CONTINUE
      BB=(BR*BR+BTH*BTH)**.5
      WRITE(NFILE,133)R*S_SCALE,TH*180/PI,BB,B0,BB/B0
133   FORMAT(1X,F7.3,T10,F6.3,T23,E9.3,T38,E9.3,T52,E9.3)
32    CONTINUE
      CLOSE(7)
      GOTO 1                  ! MAIN LOOP
9999  IF(G_M.NE.0)CALL MW$TURNOFF(1,1)
      STOP
      END

      SUBROUTINE ERRS(X,Y)
C     WRITES ERROR MESSAGE,
C     GETS NEW X,Y, THEN BLANKS ERROR MESSAGE AND OLD SQUARE
C     AT X,Y
      CHARACTER*40 C
      INTEGER*2 MX,MY,XR(4)
      X0=X
      Y0=Y
      CALL MW$LOCAT(1.,7.,0)
      C='UNACCEPTABLE POINT. RE-ENTER'
      CALL DRAWSTRING(C)
```

```
      XX=X
      YY=Y
      CALL GETXY(XX,YY)
      X=XX
      Y=YY
      X0=X0-.05
      Y0=Y0-.05
      CALL MW$GETCS(MX,MY,X0,Y0)
      XR(1)=MX
      XR(2)=MY
      X0=X0+.1
      Y0=Y0+.1
      CALL MW$GETCS(MX,MY,X0,Y0)
      XR(3)=MX
      XR(4)=MY
      CALL ERASERECT(XR)
      CALL MW$LOCAT(1.,7.,0)
      C='                                    '
      CALL DRAWSTRING(C)
      CALL MW$LOCAT(XX,YY,0)
      RETURN
      END

      FUNCTION F_ATAN(DX,DY)
C     COMPUTES ATAN(DX/DY) WITH VALUE IN RANGE 0 TO PI
C     ALSO WORKS FOR DY=0
      PI=ATAN(1.)*4
      IF(DY.EQ.0) THEN
      TH=PI/2
      ELSE
      TH=ATAN(DX/DY)
      ENDIF
      IF(TH.LT.0)TH=TH+PI
      IF(DX.LT.0.AND.DY.LT.0) TH=TH+PI
      F_ATAN=TH
      RETURN
      END

C     FORTRAN VERSION OF TBASIC SUBS FOR SHIELDIN, FLUXPLOT
C     GAUSSJ IS TAKEN DIRECTRLY FROM BOOK, NUMERICAL RECIPES, BY
C     PRESS ET AL, SEE PAGE 28 FOR DOCUMENTATION AND DISCUSSION
C     CHANGED HERE TO SET NMAX=100 AND HANDLE INTERRUPTS DEPENDING
C     ON GRAPH MODE ON/OFF THRU SWITCH, G_M
      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
      PARAMETER (NMAX=100)
      DIMENSION A(NP,NP),B(NP,MP),IPIV(NMAX),INDXR(NMAX),INDXC(NMAX)
      COMMON/GMODE/G_M!FOR GRAPH MODE ON/OFF
      DO 11 J=1,N
      IPIV(J)=0
11    CONTINUE
      DO 22 I=1,N
      BIG=0.
      DO 13 J=1,N
      IF(IPIV(J).NE.1)THEN
      DO 12 K=1,N
      IF (IPIV(K).EQ.0) THEN
      IF (ABS(A(J,K)).GE.BIG)THEN
      BIG=ABS(A(J,K))
      IROW=J
      ICOL=K
```

```
      ENDIF
      ELSE IF (IPIV(K).GT.1) THEN
C     PAUSE 'SINGULAR MATRIX'
      GOTO 9999
      ENDIF
12    CONTINUE
      ENDIF
13    CONTINUE
      IPIV(ICOL)=IPIV(ICOL)+1
      IF (IROW.NE.ICOL) THEN
      DO 14 L=1,N
      DUM=A(IROW,L)
      A(IROW,L)=A(ICOL,L)
      A(ICOL,L)=DUM
14    CONTINUE
      DO 15 L=1,M
      DUM=B(IROW,L)
      B(IROW,L)=B(ICOL,L)
      B(ICOL,L)=DUM
15    CONTINUE
      ENDIF
      INDXR(I)=IROW
      INDXC(I)=ICOL
      IF (A(ICOL,ICOL).EQ.0.) GOTO 9999    ! PAUSE 'SINGULAR MATRIX.'
      PIVINV=1./A(ICOL,ICOL)
      A(ICOL,ICOL)=1.
      DO 16 L=1,N
      A(ICOL,L)=A(ICOL,L)*PIVINV
16    CONTINUE
      DO 17 L=1,M
      B(ICOL,L)=B(ICOL,L)*PIVINV
17    CONTINUE
      DO 21 LL=1,N
      IF(LL.NE.ICOL)THEN
      DUM=A(LL,ICOL)
      A(LL,ICOL)=0.
      DO 18 L=1,N
      A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
18    CONTINUE
      DO 19 L=1,M
      B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19    CONTINUE
      ENDIF
21    CONTINUE
22    CONTINUE
      DO 24 L=N,1,-1
      IF(INDXR(L).NE.INDXC(L))THEN
      DO 23 K=1,N
      DUM=A(K,INDXR(L))
      A(K,INDXR(L))=A(K,INDXC(L))
      A(K,INDXC(L))=DUM
23    CONTINUE
      ENDIF
24    CONTINUE
      RETURN
9999  IF(G_M.EQ.1) CALL MW$TURNOFF(1,1)
      WRITE(*,*)"SINGULAR MATRIX IN GAUSSJ"
      STOP
      END
```

```
C       START OF MAKEP SUBROUTINE
C       8/30/87 FORTRAN VERSION
C       OUTPUTS PMN(M,L) AS LEGENDRE FUNCTION, P (SUPER M) (SUB L) AS FUNCTION
C       OF Z, FOR L=0 THRU N
        SUBROUTINE MAKEP(N,Z,PMN)
        DIMENSION PMN(0:1,0:50)
        PMN(0,0)=1
        PMN(1,0)=0
        PMN(0,1)=Z
        PMN(1,1)=(1-Z*Z)**.5
        PMN(0,2)=(3.*Z*Z-1)/2
        PMN(1,2)=3*Z*(1-Z*Z)**.5
        IF( N.GT.2) THEN
        DO 1 K=3,N
        DO 2 MM=0,1
        X=K-MM
        PMN(MM,K)=((2*K-1)*Z*PMN(MM,K-1)-(K+MM-1)*PMN(MM,K-2))/X
2       CONTINUE
1       CONTINUE
        END IF
        RETURN
        END

C       START OF SUBROUTINE GFNC
C       8/13/87 FORTRAN VERSION.
C       RETURNS G AS VALUE OF FUNCTION DEFINED IN NOTES 1/7/87 FOR USE
C       IN COMPUTATION OF FIELD PRODUCED BY CYLINDER SOURCE
        FUNCTION GFNC(M,N,Z,PMN,R)
        DIMENSION PMN(0:1,0:50)
        RR_G=R*R
        G=0
        IF (N.GE.1) THEN
        IF ((M.EQ.2) .OR. (N.GT.1)) THEN
        IF (M.EQ.1) THEN
        G=-((((1-Z*Z)*RR_G)**(N/2.))/R)*PMN(1,N-1)/FLOAT(N-1)
        ELSE
        G=((((1-Z*Z)*RR_G)**((-N-1.)/2.))/R)*PMN(1,N+1)/FLOAT(N+2)
        END IF
        ELSE
        G=Z
        END IF
        END IF
        GFNC=G
        RETURN
        END

C       START OF SUBROUTINE CYLSRC
C       8/13/87  FORTRAN VERSION
C       OUTPUTS BR, BTH  WHICH ARE PROPORTIONAL TO THE R, THETA COMPONENTS
C       OF MAGNETIC FIELD, BR,BTH AT R,TH, FOR CYLINDER SOURCE.  MULTIPLICATVE
C       CONSTANT NEEDED IS MU0*I/2, WHERE MU0=4*PI*10**(-7), I=CURRENT/LENGTH
C       IN AMP/METERS.  HERE COORDINATES ARE SCALED SO RADIUS OF CYLINDER
C       IS ONE, AND IN THESE UNITS, LENGTH IS H.
C       ALSO REQUIRES OTHER ARGUMENTS, INCLUDING LEGENDRE POLYNOMIALS TO HAVE
C       BEEN CALCULATED ALREADY
        SUBROUTINE CYLSRC(NT,R,BR,BTH,Z0,Z1,S1,PMN,QMN,RMN)
        DIMENSION PMN(0:1,0:50),QMN(0:1,0:50),RMN(0:1,0:50)
        BR=0
        BTH=0
        IF (R.NE.0) THEN
```

```
      DO 1 N=1,NT,2
      IF (R.LE.1) THEN
      G1N= GFNC(1,N,Z0,QMN,R)
      BR=BR+2*G1N*PMN(0,N)
      BTH=BTH-(2./N)*G1N*PMN(1,N)
      ELSE
      IF (R.LE.S1) THEN
      G1N0=GFNC(1,N,Z0,QMN,R)
      G1N1=GFNC(1,N,Z1,RMN,R)
      G2N1=GFNC(2,N,Z1,RMN,R)
      BR=BR+2*(G1N0-G1N1+G2N1)*PMN(0,N)
      BTH=BTH-((2./N)*(G1N0-G1N1)-(2./(N+1.))*G2N1)*PMN(1,N)
      ELSE
      G2N=GFNC(2,N,Z0,QMN,R)
      BR=BR+2*G2N*PMN(0,N)
      BTH=BTH-(-2./(N+1.))*G2N*PMN(1,N)
      END IF
      END IF
1     CONTINUE
      END IF
      RETURN
      END

C     FOLLOWING DIRECTLY FROM NUMERICAL RECIPES BOOK, PAGE 114
C     MODIFIED TO INCLUDE SWITCH, G_M, TO HANDLE ERROR TRAPS FOR
C     GRAPH MODE ON/OFF
      SUBROUTINE QROMB(FUNC,A,B,SS)
      EXTERNAL FUNC
      PARAMETER(EPS=1.E-6,JMAX=20,JMAXP=JMAX+1,K=5,KM=4)
      DIMENSION S(JMAXP),H(JMAXP)
      COMMON/GMODE/G_M                   !  GRAPH OFF/ON
      H(1)=1.
      DO 11 J=1,JMAX
      CALL TRAPZD(FUNC,A,B,S(J),J)
      IF (J.GE.K) THEN
      L=J-KM
      CALL POLINT(H(L),S(L),K,0.,SS,DSS)
      IF (ABS(DSS).LT.EPS*ABS(SS)) RETURN
      ENDIF
      S(J+1)=S(J)
      H(J+1)=0.25*H(J)
11    CONTINUE
      IF(G_M.EQ.1) CALL MW$TURNOFF(1,1)
      PAUSE 'TOO MANY STEPS.'
      END

C     POLINT FROM NUMERICAL RECIPES BOOK, PAGE 82
      SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
      PARAMETER (NMAX=10)
      DIMENSION XA(N),YA(N),C(NMAX),D(NMAX)
      NS=1
      DIF=ABS(X-XA(1))
      DO 11 I=1,N
      DIFT=ABS(X-XA(I))
      IF (DIFT.LT.DIF) THEN
      NS=I
      DIF=DIFT
      ENDIF
      C(I)=YA(I)
      D(I)=YA(I)
```

```
11      CONTINUE
        Y=YA(NS)
        NS=NS-1
        DO 13 M=1,N-1
        DO 12 I=1,N-M
        HO=XA(I)-X
        HP=XA(I+M)-X
        W=C(I+1)-D(I)
        DEN=HO-HP
        IF(DEN.EQ.0.)PAUSE
        DEN=W/DEN
        D(I)=HP*DEN
        C(I)=HO*DEN
12      CONTINUE
        IF (2*NS.LT.N-M)THEN
        DY=C(NS+1)
        ELSE
        DY=D(NS)
        NS=NS-1
        ENDIF
        Y=Y+DY
13      CONTINUE
        RETURN
        END

C       TRAPZD FROM NUMERICAL RECIPES BOOK, PAGE 111
        SUBROUTINE TRAPZD(FUNC,A,B,S,N)
        EXTERNAL FUNC
        IF (N.EQ.1) THEN
        S=0.5*(B-A)*(FUNC(A)+FUNC(B))
        IT=1
        ELSE
        TNM=IT
        DEL=(B-A)/TNM
        X=A+0.5*DEL
        SUM=0.
        DO 11 J=1,IT
        SUM=SUM+FUNC(X)
        X=X+DEL
11      CONTINUE
        S=0.5*(S+(B-A)*SUM/TNM)
        IT=2*IT
        ENDIF
        RETURN
        END

C       FOR STRAIGHT LINE SEGMENTS
C       THE FOLLOWIN SET OF FUNCTIONS PROVIDE THE RADIUS AS A
C       FUNCTION OF THETA FOR INNER SURFACE, SRA, AND FOR OUTER, SRB
C       NORMALS ALONG R,THETA UNIT VECTORS, ARE NRA,NTHA (INNER) AND
C       NRB, NTHB (OUTER).  ALSO, MU(TH) PROVIDES MU AS A FUNCTION OF
C       THETA CORRESPONDING TO SHIELD GAP. THESE ARE ALL REAL.
        FUNCTION SRA(TH)
        DIMENSION FA(0:19),FB(0:19)
        REAL MU_0,MU_1,MU
        COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
        REAL NRA,NTHA,NRB,NTHB,MU,MU_0,MU_1
C       XA,YA ARE INNER POINTS ENTERED
C       XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C       RA,THA,RB,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
```

```
C     A(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),A(20)
      REAL RA(20),RB(20)
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,A,RA,RB,NP
C     THA(1)=THB(1) IS GAP ANGLE.  IF TH<THA(1) FILL OUT SHIELD WITH
C     LINES PARALLEL TO FIRST ONE, ANGLE A(1).  SOME SHAPE IS NEEDED
C     FOR MATCHING ACROSS SHIELD SURFACE IN GAP ANGLE, WHERE ABSENCE
C     OF SHIELD IS ACCOUNTED FOR BY SET MU=1 THERE IN FUNCTION MU(TH)
      DO 1 I=1,NP
      IF(TH.LT.THA(I)) GOTO 2
1     CONTINUE
C     IF TH>LAST ENTERED ANGLE, USE A(NP), WHICH HAS BEEN
C     SET=PI TO MAKE VERTICAL INTERSECTION WITH X-AXIS
      I=NP+1
2     IF(I.GT.1)I=I-1
      X_S=RA(I)*SIN(A(I)-THA(I))/SIN(A(I)-TH)
9     SRA=X_S
      RETURN
      ENTRY SRB(TH)
      DO 11 I=1,NP
      IF(TH.LT.THB(I)) GOTO 21
11    CONTINUE
      I=NP+1
21    IF(I.GT.1)I=I-1
      X_S=RB(I)*SIN(A(I)-THB(I))/SIN(A(I)-TH)
91    SRB=X_S
      RETURN
      ENTRY NRA(TH)
      DO 13 I=1,NP
      IF(TH.LT.THA(I)) GOTO 23
13    CONTINUE
      I=NP+1
23    IF(I.GT.1)I=I-1
      X_S=RA(I)*SIN(A(I)-THA(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=(1.+(DRA/X_S)**2)**(-.5)
93    NRA=X_S
      RETURN
      ENTRY NTHA(TH)
      DO 14 I=1,NP
      IF(TH.LT.THA(I)) GOTO 24
14    CONTINUE
      I=NP+1
24    IF(I.GT.1)I=I-1
      X_S=RA(I)*SIN(A(I)-THA(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=-(DRA/X_S)*(1.+(DRA/X_S)**2)**(-.5)
94    NTHA=X_S
      RETURN
      ENTRY NRB(TH)
      DO 15 I=1,NP
      IF(TH.LT.THB(I)) GOTO 25
15    CONTINUE
      I=NP+1
25    IF(I.GT.1)I=I-1
      X_S=RB(I)*SIN(A(I)-THB(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=(1.+(DRA/X_S)**2)**(-.5)
95    NRB=X_S
```

```
      RETURN
      ENTRY NTHB(TH)
      DO 16 I=1,NP
      IF(TH.LT.THB(I)) GOTO 26
16    CONTINUE
      I=NP+1
26    IF(I.GT.1)I=I-1
      X_S=RB(I)*SIN(A(I)-THB(I))/SIN(A(I)-TH)
      DRA=X_S*COS(A(I)-TH)/SIN(A(I)-TH)
      X_S=-(DRA/X_S)*(1.+(DRA/X_S)**2)**(-.5)
96    NTHB=X_S
      RETURN
      ENTRY MU(TH)
      IF(TH.LE.TH_MU0) THEN
      MU=MU_0
      ELSE
      MU=MU_1
      END IF
      RETURN
      END

      SUBROUTINE OUTER(IERR)
C     FORTRAN VERSION 11/29/87
C     FOR STRAIGHTLINE SEGMENT SHIELDS
C     INPUT IS XA,YA,RA,THA,A AND OUTPUT IS XB,YB,RB,THB
C     OUTER STRAIGHT LINE SEGMENTS ARE AT DISTANCE T FROM
C     INNER ONES.
C     XA,YA ARE INNER POINTS ENTERED
C     XB,YB ARE OUTER POINTS ENTERED OR COMPUTED
C     RA,THA,RB,THB ARE RADIUS AND ANGLE COMPUTED FROM XA,YA,XB,YB
C     A(K) IS ANGLE OF KTH LINE
C     NP=NUMBER OF POINTS
      REAL XA(20),YA(20),XB(20),YB(20),THA(20),THB(20),A(20)
      REAL RA(20),RB(20),MU_0,MU_1
      COMMON/STRAIGHT/XA,YA,XB,YB,THA,THB,A,RA,RB,NP
      DIMENSION FA(0:19),FB(0:19)
      COMMON N_S,FA,FB,TH_MU0,MU_0,MU_1,THICK
      IERR=0    ! ZERO ERROR
C     FIRST IS SPECIAL
      I=1
      THB(I)=THA(I)
      SAT=SIN(A(I)-THA(I))
      TT=THICK/SAT
      RB(I)=RA(I)+TT
      YB(I)=YA(I)+TT*COS(THA(I))
      XB(I)=XA(I)+TT*SIN(THA(I))
      DO 2 I=1,NP-1
      SAT=SIN(A(I)-A(I+1))
      IF(SAT.EQ.0) THEN
      DX=-THICK*COS(A(I))
      DY=THICK*SIN(A(I))
      ELSE
      DX=THICK*(SIN(A(I+1))-SIN(A(I)))/SAT
      DY=THICK*(COS(A(I+1))-COS(A(I)))/SAT
      ENDIF
      XB(I+1)=XA(I+1)+DX
      YB(I+1)=YA(I+1)+DY
      RB(I+1)=(XB(I+1)**2+YB(I+1)**2)**.5
2     THB(I+1)=F_ATAN(XB(I+1),YB(I+1))
C     CHECK FOR ERROR
```

```
      DO 4 I=1,NP
      IF(XB(I).LT.0.OR.YB(I).LT.0) GO TO 913
      IF(I.GT.1) THEN
      IF(THB(I).LT.THB(I-1)) GOTO 913
      ENDIF
4     CONTINUE
      RETURN
913   IERR=1
      RETURN
      END

C     FORTRAN VERSION 8/13/87
C     CARL H. BRANS    12/31/86
C     FUNC IS THE FUNCTION USED BY THE INTEGRATION ROUTINE, QROMB,
C     WHICH IS CALLED TO INTEGRATE ALONG THE SPHERICAL ANGLE THETA
C     TO PRODUCE THE VOLUME OF THE SHIELD BETWEEN THE RADII SRA(TH)
C     AND SRB(TH)
      FUNCTION FUNC(X)
      X_F=SRA(X)
      Y_F=SRB(X)
      FUNC=(Y_F**3-X_F**3)*SIN(X)
      RETURN
      END

      SUBROUTINE GRAPH(S_G,H_G)
C     GRAPH PREPARES GRAPHICS:  AXES, SCALES, ETC.
C     LAHEY FORTRAN VERSION, 8/14/87
      CHARACTER*37 MSG,C1*2
      ASP=1.
      CALL MW$TURNON(1,ASP)
      CALL MW$AXES(-1.,12.,-1.,9.,1.,1.)
      DO 1 I=1,11
      X=I
      CALL MW$LOCAT(X-.2,-.8,0)
      WRITE(C1,100)I
100   FORMAT(I2)
1     CALL DRAWSTRING(C1)
      IF(H_G.GT.0) THEN
      DO 2 X=.98,1.02,.01
      CALL MW$LOCAT(X,0.,0)
2     CALL MW$LOCAT(X,H_G/2,1)
      ENDIF
      WRITE(MSG,101)S_G
101   FORMAT('NOTE: EACH AXES UNIT IS ',F5.3,' METERS')
      CALL MW$LOCAT(1.,8.5,0)
      CALL DRAWSTRING(MSG)
      RETURN
      END

      SUBROUTINE GETXY(X,Y)
C     FOR SHIELDIN  8/29/87
C     GETS MOUSE POINT WHEN LEFT BUTTON IS DEPRESSED AND DRAWS
C     CIRCLE AROUND POINT, SHOWS UP AS SQUARE, BECAUSE OF
C     DISCRETE ROUNDOFF
      INTEGER*2 MX,MY,BUTTONS,SR(4)
      REAL OTHER(9)
      COMMON/MWGR/SR,OTHER
      CALL MW$GETCS(MX,MY,X,Y)
      CALL MOVECURSOR(MX,MY)!LOCATE TO LAST POINT, SO ON FIRST CALL SET X,Y
      CALL SHOWCURSOR
```

```
1       CALL READMOUSE(MX,MY,BUTTONS)
        IF(BUTTONS.LT.0) GOTO 1!NO CHANGE
        CALL MOVECURSOR(MX,MY)
        IF(BUTTONS.NE.4) THEN
        CALL MOVETO(MX,MY)
        GOTO 1
        ENDIF
C       BUTTONS=4, LEFT BUTTON
        CALL MOVETO(MX,MY)
        CALL HIDECURSOR
C       DRAW SMALL CIRCLE
        PI=4.*ATAN(1.)
        CALL MW$GETXY(MX,MY,X,Y)!CONVERTS TO REAL X,Y
        IF=0
        DO 2 TH=0,2*PI,.1
        XX=X+.05*COS(TH)
        YY=Y+.05*SIN(TH)
        CALL MW$LOCAT(XX,YY,IF)
2       IF=1
        CALL MW$LOCAT(X,Y,0)
        RETURN
        END
```

0.500	22.500	0.223E-03	0.186E-02	0.120E+00
0.550	22.500	0.168E-03	0.140E-02	0.120E+00
0.600	22.500	0.130E-03	0.108E-02	0.121E+00
0.650	22.500	0.103E-03	0.848E-03	0.121E+00
0.700	22.500	0.824E-04	0.679E-03	0.121E+00
0.750	22.500	0.671E-04	0.552E-03	0.122E+00
0.800	22.500	0.554E-04	0.455E-03	0.122E+00
0.850	22.500	0.462E-04	0.379E-03	0.122E+00
0.900	22.500	0.390E-04	0.319E-03	0.122E+00
0.950	22.500	0.332E-04	0.272E-03	0.122E+00
0.100	45.000	0.511E+00	0.211E+00	0.242E+01
0.150	45.000	0.806E-02	0.587E-01	0.137E+00
0.200	45.000	0.306E-02	0.245E-01	0.125E+00
0.250	45.000	0.154E-02	0.125E-01	0.123E+00
0.300	45.000	0.891E-03	0.724E-02	0.123E+00
0.350	45.000	0.560E-03	0.456E-02	0.123E+00
0.400	45.000	0.375E-03	0.305E-02	0.123E+00
0.450	45.000	0.264E-03	0.214E-02	0.123E+00
0.500	45.000	0.192E-03	0.156E-02	0.123E+00
0.550	45.000	0.144E-03	0.117E-02	0.123E+00
0.600	45.000	0.111E-03	0.904E-03	0.123E+00
0.650	45.000	0.875E-04	0.711E-03	0.123E+00
0.700	45.000	0.700E-04	0.569E-03	0.123E+00
0.750	45.000	0.569E-04	0.463E-03	0.123E+00
0.800	45.000	0.469E-04	0.381E-03	0.123E+00
0.850	45.000	0.391E-04	0.318E-03	0.123E+00
0.900	45.000	0.330E-04	0.268E-03	0.123E+00
0.950	45.000	0.280E-04	0.228E-03	0.123E+00
0.100	67.500	0.599E+00	0.150E+00	0.399E+01
0.150	67.500	0.652E-02	0.443E-01	0.147E+00
0.200	67.500	0.261E-02	0.186E-01	0.140E+00
0.250	67.500	0.128E-02	0.952E-02	0.134E+00
0.300	67.500	0.719E-03	0.550E-02	0.131E+00
0.350	67.500	0.446E-03	0.346E-02	0.129E+00
0.400	67.500	0.295E-03	0.232E-02	0.127E+00
0.450	67.500	0.206E-03	0.163E-02	0.126E+00
0.500	67.500	0.149E-03	0.119E-02	0.126E+00
0.550	67.500	0.112E-03	0.891E-03	0.125E+00
0.600	67.500	0.858E-04	0.686E-03	0.125E+00
0.650	67.500	0.673E-04	0.540E-03	0.125E+00
0.700	67.500	0.538E-04	0.432E-03	0.124E+00
0.750	67.500	0.437E-04	0.351E-03	0.124E+00
0.800	67.500	0.359E-04	0.289E-03	0.124E+00
0.850	67.500	0.299E-04	0.241E-03	0.124E+00
0.900	67.500	0.252E-04	0.203E-03	0.124E+00
0.950	67.500	0.214E-04	0.173E-03	0.124E+00