LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

Fusion Energy Division

ORNL/FEDC--87/1

DE88 003668

COOLDOWN OF THE COMPACT IGNITION TOKAMAK

D. C. KEETON
Fusion Engineering Design Center
and
Tennessee Valley Authority

Date published: August 1987

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

MASTER

CONTENTS

ABSTRACT.....	v
ACKNOWLEDGMENTS.....	vii
1. SUMMARY.....	1
2. COOLDOWN WITH LIQUID NITROGEN AT ATMOSPHERIC PRESSURE.....	4
2.1 SYSTEM DESCRIPTION.....	4
2.2 PHYSICAL MODEL.....	4
2.3 HEAT TRANSFER MODEL.....	6
2.3.1 Conduction.....	6
2.3.2 Two-Phase Heat Transfer.....	6
2.3.2.1 Nucleate and Transition Boiling.....	7
2.3.2.2 Annular-Flow Film Boiling.....	7
2.3.2.3 Mist-Flow Film Boiling.....	7
2.4 RESULTS WITH LIQUID NITROGEN COOLANT.....	10
2.4.1 TF Coil Inner Leg.....	10
2.4.2 Ohmic-Heating Solenoid.....	14
3. COOLDOWN WITH PRESSURIZED GASEOUS HELIUM.....	15
3.1 SYSTEM DESCRIPTION.....	15
3.2 PHYSICAL MODEL.....	23
3.3 HEAT TRANSFER MODEL.....	26
3.3.1 Conduction.....	26
3.3.2 Forced Convection.....	26
3.4 RESULTS WITH HELIUM COOLANT.....	27
REFERENCES.....	29
APPENDIX A. COMPUTER PROGRAM TO ANALYZE LIQUID NITROGEN COOLING OF THE CIT TF COIL INNER LEG.....	31
APPENDIX B. MIST-FLOW HEAT TRANSFER CORRELATIONS APPLIED TO THE CIT.....	47
APPENDIX C. COMPUTER PROGRAM TO ANALYZE HELIUM COOLING OF THE CIT TF COIL INNER LEG.....	53
APPENDIX D. COMPUTER PROGRAM TO ANALYZE HELIUM COOLING OF THE CIT OH SOLENOID.....	65

ABSTRACT

Cooldown of the Compact Ignition Tokamak (CIT) with the baseline liquid nitrogen cooling system was analyzed. On the basis of this analysis and present knowledge of the two-phase heat transfer, the current baseline CIT can be cooled down in about 1.5 h. An extensive heat transfer test program is recommended to reduce uncertainty in the heat transfer performance and to explore methods for minimizing the cooldown time.

An alternate CIT cooldown system is described which uses a pressurized gaseous helium coolant in a closed-loop system. It is shown analytically that this system will cool down the CIT well within 1 h. Confidence in this analysis is sufficiently high that a heat transfer test program would not be necessary. The added cost of this alternate system is estimated to be about \$5.3 million. This helium cooling system represents a reasonable backup approach to liquid nitrogen cooling of the CIT.

ACKNOWLEDGMENTS

The author wishes to acknowledge the valuable contributions to this work by other members of the Compact Ignition Tokamak (CIT) design team. Credit is given to Ali Dabiri for first suggesting helium as an alternate coolant for the CIT and for performing the early analysis on this to establish its feasibility. Also, the support of John Haines in reviewing the work reported here and in providing technical input and encouragement in two-phase heat transfer is appreciated.

1. SUMMARY

This report describes work performed at the Fusion Engineering Design Center (FEDC) to calculate the cooldown time of the Compact Ignition Tokamak (CIT) between pulses with liquid nitrogen as the baseline coolant and pressurized gaseous helium as a backup coolant.

The conceptual design of the CIT uses inertial cooling to absorb ohmic and neutron heating during pulses. In the interval between pulses, internal components are cooled to near the saturation temperature of liquid nitrogen at atmospheric pressure (77.6 K). Use of a low starting temperature extends the pulse length by reducing the initial ohmic-heating rate and by extending the allowable temperature excursion before copper strength is degraded.

In the baseline conceptual design of the CIT, a large fraction of the heat is removed by liquid nitrogen, flowing upward along the nose of the toroidal field (TF) coil and along the inner and outer radii of the ohmic-heating (OH) solenoid. Heat is removed by extracting latent heat of vaporization from the flowing liquid nitrogen. The project estimates that about 18,000 L, or 14,000 kg, of liquid nitrogen will be required for each full-power pulse. This liquid nitrogen will be delivered to the site in bulk quantity and stored for use in a series of shots.

Calculations to estimate the cooldown time of the TF coil inner leg with liquid nitrogen coolant were performed and are described in detail in Sect. 2. These calculations indicate that achieving the desired 1-h cooldown time between full-power pulses with liquid nitrogen coolant may not be possible. On the basis of present knowledge of nucleate- and film-boiling heat transfer coefficients in this geometry, about 1.5 h appears necessary to cool the TF coil inner leg with the present design. Extrapolation of these results to the OH solenoid indicates that a longer cooldown time would be required for this component, since nucleate boiling of liquid nitrogen would be initiated much later in time because of higher wall temperatures at the coolant inlet. It may be possible to increase

the heat-removal rate, and thereby shorten cooldown time, by altering the TF coil surface by roughening it or by adding fins. In any case, a significant test program would be required to confirm that cooldown can be achieved in the desired time with boiling liquid nitrogen. These tests should duplicate, as nearly as possible, the actual CIT configuration and initial temperature distribution.

An added concern with two-phase flow in the CIT geometry is the effect of flow distribution and flow instability. Results of these analyses show that pressure drop through the TF coil flow channel changes significantly during a cooldown transient. The present design provides a common liquid nitrogen plenum for the OH solenoid and the TF coil inner leg. During cooldown, these two channels will compete for liquid nitrogen flow. A passive flow adjustment providing adequate flow to cool down both components will have to be designed. Separate flow headers to each component may be required to allow each to have individual external adjustment of flow during the transient. Flow instability and nonuniformity in the narrow annulus of the TF coil nose is also a concern. This annulus is equivalent to a flow channel 3 m long, 1 cm wide, and 3 m high. Flow stability of the two-phase mixture may be a problem in this annulus and should be investigated further. In addition, nonuniform flow may be a problem since the hottest magnet segments will experience the largest pressure drop, further reducing coolant flow to that region, causing it to be cooled more slowly. Instability and nonuniform flow may require that liquid nitrogen flow be channeled separately in each TF coil nose. A comprehensive test program using liquid nitrogen coolant should include tests for flow distribution and instability.

Conceptual design work was performed to evaluate an alternate CIT cooling system using pressurized gaseous helium as the working fluid in a closed circulating system. Boiling liquid nitrogen would be used in an external part of the loop to remove heat from the helium. Calculations were performed to evaluate this system with helium pressurized to about 8 atm, and results of the calculations are described in detail in Sect. 3. The conclusion from this work is that an alternate cooling system using pressurized helium as coolant

could be provided for CIT at a cost of about \$5.3 million, excluding any increased building space and any impacts on CIT structural design. This cooling system is calculated to cool down the CIT TF coil inner leg and the OH solenoid in about half an hour, with a reasonable pressure drop. Confidence in these calculations is quite high since heat transfer coefficients for simple gases are well known and do not depend strongly on material surface conditions. As a result, design of CIT TF coils and the OH solenoid could continue in the present configuration with a high assurance that cool-down time would be less than 1 h.

2. COOLDOWN WITH LIQUID NITROGEN AT ATMOSPHERIC PRESSURE

2.1 SYSTEM DESCRIPTION

In the baseline cooldown system for the CIT, heat is removed by liquid nitrogen at near atmospheric pressure, which flows up along the nose of the TF coil and along the inner and outer radii of the OH solenoid. These two components represent the major heat load for the CIT. Other PF coils and the outer leg of the TF coil are also cooled by liquid nitrogen. The vacuum vessel and rf antennas are cooled by cold nitrogen gas. Liquid nitrogen will be delivered to the site in bulk quantity and stored in tanks for use in a series of shots. During operation, it will be piped from the storage tanks to an elevated tank, where gravity will supply the head to force liquid through CIT components. Heat of vaporization (0.2 MJ/kg) of the liquid nitrogen will cool the components. Liquid and vapor will be discharged from the top of the CIT. A separator will extract the liquid, which will be pumped back to the elevated tank for reuse; vapor will be piped to the atmosphere. The project estimates that about 18,000 L of liquid nitrogen will be consumed during each cooldown following a full-power shot. For a more detailed description of this system, refer to Sect. 4.17 of ref. 1.

2.2 PHYSICAL MODEL

A finite element, two-dimensional model of the CIT TF coil inner leg (Fig. 2.1) was developed to perform transient heat transfer calculations. The TF coil inner leg was divided axially into 20 nodes, with conduction between adjacent nodes and insulation at the top and bottom. The coil was divided radially into six nodes near the midplane of the inner leg and eight nodes at the top and bottom. Conduction was allowed between adjacent nodes. A 1-cm annular flow channel was modeled at the TF coil nose, with convective heat transfer between this channel and the TF coil nose. A small amount of heat was removed from the inner leg by conduction to the outer leg. This was simulated by providing a small conductance

Fig. 2.1. Physical model of CIT TF coil inner leg used in cooldown calculations.

between the outer nodes of the coil and the boundary nodes maintained at 80 K. Typically, 10% of the heat in the inner leg was removed by this mechanism.

2.3 HEAT TRANSFER MODEL

The transient heat transfer model that was applied to the physical model consisted of conduction between adjacent solid nodes and convection between fluid coolant nodes and the first solid node. A simple, forward difference procedure was used to calculate time-dependent temperatures. A listing of the computer program used in the analysis, along with typical input data, is provided in Appendix A.

2.3.1 Conduction

Conduction between solid nodes in the TF coil inner leg was based on a homogeneous mixture of copper and Inconel in a 57.5 to 37.5% ratio. Temperature-dependent conductivity and specific heat of this mixture was developed and used in the calculation. The error in using this homogeneous mixture rather than alternate layers of copper and Inconel was evaluated; because of the thin laminates of copper and Inconel and because heat is conducted axially and radially rather than circumferentially, the error is not significant.

2.3.2 Two-Phase Heat Transfer

The major uncertainty in calculating the cooldown time of the CIT with liquid nitrogen is the magnitude of two-phase heat transfer coefficients. The calculational procedure used in this analysis divided the two-phase heat transfer into four regions: (1) nucleate boiling, characterized by quality (weight percent vapor in the flow) less than 10% and a temperature difference between the fluid and the wall of less than 12 K; (2) transition boiling, characterized by quality less than 10% and a temperature difference between 12 and 50 K; (3) annular-flow film boiling, characterized by quality less than 10% and a temperature difference greater than 50 K; and

(4) mist-flow film boiling, characterized by quality greater than 10%. These regions are summarized in Table 2.1.

2.3.2.1 Nucleate and Transition Boiling

For nucleate and transition boiling, the correlations reported in ref. 2 for pool boiling of liquid nitrogen were used. Figure 2.2 shows the ref. 2 data plotted in terms of the heat transfer coefficient. The transition-boiling curve is estimated by a straight line connecting the coefficient at critical heat flux with the coefficient at minimum film boiling.

2.3.2.2 Annular-Flow Film Boiling

For annular-flow film boiling at qualities less than 10%, the pool film-boiling curves reported in ref. 2 for liquid nitrogen were used. This coefficient, practically constant with temperature, has an average value of about $160 \text{ W/m}^2\text{-K}$. Figure 2.2 also shows these data.

2.3.2.3 Mist-Flow Film Boiling

Boiling heat transfer coefficients at qualities greater than about 10% can be characterized as a mist-flow region. The liquid is contained in small-diameter, well-dispersed droplets that are swept along with the vapor. A significant portion of the heat removed from the CIT will be in this regime; therefore, it is important that the model be as accurate as possible. Fortunately, a great deal of theoretical and experimental work on liquid nitrogen in this flow regime was done at the Massachusetts Institute of Technology in the 1968 time frame and is reported in ref. 3. The calculational procedure recommended was modified for use in analyzing the CIT. These modifications are described in Appendix B. On the basis of data in Appendix B for heat fluxes of interest to the CIT, it is concluded that the modified model gives results in good agreement with MIT experimental results.

Table 2.1. Two-phase heat transfer regions used in cooldown analysis of CIT TF coil inner leg with liquid nitrogen

ΔT (K)	Quality (%)			
	<1	1-10	10-100	>100
<10	Greater of liquid convection or pool nucleate boiling	Pool nucleate boiling	Mist-flow boiling	Vapor convection
10-12	Pool nucleate boiling	Pool nucleate boiling	Mist-flow boiling	Vapor convection
12-50	Transition boiling	Transition boiling	Mist-flow boiling	Vapor convection
>50	Pool film boiling	Pool film boiling	Mist-flow boiling	Vapor convection

Fig. 2.2. Pool-boiling heat transfer coefficients based on data of ref. 2.

2.4 RESULTS WITH LIQUID NITROGEN COOLANT

2.4.1 TF Coil Inner Leg

This section describes results obtained from calculating cooldown time of the CIT TF coil inner leg with liquid nitrogen. The physical and heat transfer models previously described were used. Figure 2.3 summarizes the results from the calculations. Cooldown time is the time required to cool the TF coil inner leg to 80 K. Nitrogen flow rate is the flow rate of liquid nitrogen into the bottom of the TF coil annular flow channel.

Figure 2.4 shows temperature vs time for the midplane at the radial center of the TF coil inner leg for liquid nitrogen flowing at 10 kg/s. The long, straight portion of this curve has a slope that is characteristic of pool film boiling because the quality is below 10% and the temperature difference between wall and fluid is greater than 50 K. At about 1.4 h into the transient, nucleate boiling has moved up to this position, and the temperature begins to drop rapidly.

Figure 2.5 is a plot of peak friction, momentum, elevation, and total pressure drop for each flow rate. Note that the sum of friction, momentum, and elevation pressure drop does not equal the peak total pressure drop, because the individual peaks occur at different times during the transient.

In each of the preceding cases, it was assumed that equipment would be available to handle liquid and gaseous nitrogen flowing at the rates used in the calculation. For example, in the 10-kg/s transient, initial flow rates at the top are 8.5 kg/s for liquid and 1.5 kg/s for gas; whereas, at the end of the same transient, flow rates are 9.7 kg/s for liquid and 0.3 kg/s for gas. No evaluation has been performed to determine whether the baseline design of the cryogenic system will adequately perform this function.

Fig. 2.3. Cooldown time of CIT TF coil inner leg with liquid nitrogen coolant.

Fig. 2.4. Temperature at the center of the midplane of CIT TF coil inner leg during cooldown with liquid nitrogen at 10 kg/s.

Fig. 2.5. Peak pressure drop during CIT cooldown with liquid nitrogen coolant.

2.4.2 Ohmic-Heating Solenoid

No calculations were performed on the OH solenoid to determine its cooldown time with liquid nitrogen. However, calculations for helium (shown later) along with the high wall temperature at the inlet to the solenoid indicate that the cooldown time of the solenoid will be longer than that for the TF coil inner leg. Helium calculations described in the next section show that cooldown time for the OH solenoid and the TF coil inner leg are roughly the same for reasonably similar heat transfer coefficients. However, in liquid nitrogen cooling, low wall temperature at the entrance to the TF coil causes nucleate boiling to be initiated immediately and then to progress up the channel by conduction. Because of high wall temperature at the OH solenoid flow entrance, nucleate boiling would not be initiated until late in the transient. The result would be a longer time to cool the OH solenoid than the TF coil.

3. COOLDOWN WITH PRESSURIZED GASEOUS HELIUM

As pointed out previously, calculations show uncertainty as to whether the CIT can be cooled down between shots in the desired time with liquid nitrogen. If a heat transfer test program confirms these calculational results, it may be necessary to change the basic design of the cooling system. This section describes an alternate cooling system using pressurized gaseous helium as the coolant.

3.1 SYSTEM DESCRIPTION

Conceptual design work was performed to evaluate an alternate cooldown system for CIT using pressurized gaseous helium as the working fluid in a closed circulating system. An external, helium-to-liquid-nitrogen heat exchanger is used to cool the helium to near the saturation temperature of liquid nitrogen before it is returned to the CIT. This evaluation is based on a helium pressure of 8 atm. This appears to be a reasonable compromise between a higher pressure (which would reduce pumping power and some of the loop component sizes) and a lower pressure (which might require fewer design accommodations in the CIT structure).

A flow diagram for the helium cooling system is shown in Fig. 3.1, and principal design parameters are listed in Table 3.1. Key features of this system are as follows:

- A regenerative heat exchanger is used to raise the helium temperature at the suction of the circulator to near room temperature. This relatively high temperature allows chilled water to accept a significant part of the heat added to the helium by the circulator. A 5 K temperature difference was used for this heat exchanger. A 10 K temperature difference would result in smaller equipment and a capital cost savings of about \$350,000. However, during operation the additional 5 K temperature difference would cost \$450 per shot for liquid nitrogen. It was concluded, therefore, that the additional capital cost would be justified.

Fig. 3.1. Helium cooling system for CIT (overall system).

Table 3.1. Helium cooling system parameters

Desired cooldown time, h	<1
Helium flow rate, kg/s	12
Maximum operating pressure, atm	8
Design pressure, psig	135
System pressure drop, atm	1
Maximum design temperature, K	320
Helium circulator power, MW	1.25

- The helium circulator is designed to raise system pressure from 7 atm to 8 atm at a flow rate of 12 kg/s. The resulting helium temperature rise is about 20 K, and the pump input power requirement is about 1.25 MW. There are no unusual design requirements for this circulator, and it could be supplied by several vendors.
- A chilled-water-cooled heat exchanger is used to remove about 75% of the heat added to the helium by the circulator. This will save about \$1500 per shot in liquid nitrogen cost. This heat exchanger also "anchors" the inlet temperature at the circulator to a constant, near-room temperature. The constant suction temperature eases the design problem of the circulator. The chilled-water system used to cool the Tokamak Fusion Test Reactor TF coils can be used to supply water to this heat exchanger.
- A helium-to-liquid-nitrogen heat exchanger is used to cool the helium before it is returned to the CIT. This heat exchanger will boil liquid nitrogen to remove heat from the helium. The consumed liquid nitrogen is replaced by bulk delivery to the site.
- A helium surge tank, along with a makeup pump, is used to maintain a constant pressure in the CIT during cooldown and heatup. Representative pressure, temperature, and helium quantities for various operating states are shown in Figs. 3.2 through 3.5. The surge tank is sized to meet two design conditions: (1) 8 atm system pressure at an ambient temperature of 320 K in case all cooling is lost for a long period of time; and (2) a minimum pressure of 1.05 atm in the tank to prevent inleakage of impurities.

A preliminary cost estimate was made for the helium cooling system shown in Fig. 3.1. The total added cost is estimated to be about \$5.3 million, excluding the cost of any additional buildings to house the equipment and the cost of any impacts on CIT structural design resulting from higher pressure. No credit was taken for

Fig. 3.2. Helium cooling system for CIT (system conditions at ambient temperature).

Fig. 3.3. Helium cooling system for CIT (system conditions immediately after a shot).

Fig. 3.4. Helium cooling system for CIT (system conditions near end of cooldown).

Fig. 3.5. Helium cooling system for CIT (system conditions immediately before a shot).

reduction in scope of the present CIT cryogenic system since most of the liquid nitrogen handling and storage equipment would be required for the helium cooling system.

Table 3.2 gives a breakdown of these costs. Major equipment cost is based on vendor quotes or similar equipment for the CIT and the previously used CIT project-related costs.

3.2 PHYSICAL MODEL

The physical model used to calculate cooldown time of the TF coil inner leg with helium was identical to that described previously for liquid nitrogen. Inlet temperature of the helium gas was programmed as a function of exit temperature and total CIT flow rate. This program was based on calculations of heat added by the circulator and heat removed by the heat exchangers.

The physical model of the OK solenoid treated each pancake as two axial nodes, one for the copper-Inconel conductor and one for the copper heat conductor. The nine turns in the radial direction were treated as five nodes to conserve computer run time. The dummy turn was conservatively treated as an active turn. Two additional radial nodes were used for copper heat-removal surfaces, and two were used for coolant flow channels.

The only change made to the baseline OK solenoid design for this heat transfer model was to increase the gas flow area and the surface area available for convective heat transfer. This was accomplished within the baseline design envelope. Figure 3.6 shows these modifications, which consisted in decreasing the relative area of insulation that is in contact with the inner and outer radii of the solenoid, thereby opening more area for gas contact with the wall.

Table 3.2. Estimated cost of a helium cooling system for CIT

Equipment	Approximate size	Vendor cost	Cost source
Regenerative heat exchanger	Two each, 1.5 × 1.5 × 6 m long	\$ 250,000	Altec International, Inc.
Helium circulator	2 × 2.5 × 5 m long	1,000,000	High side of 13 vendor quotes
Heat exchanger (chilled water to helium)	1 m diam × 2 m long	25,000	Earl F. Scott Co. type BPO, Model 31-75
Heat exchanger (helium to liquid nitrogen)	1.2 × 1.2 × 2 m high	45,000	Altec International, Inc.
Helium surge tank	760-m ³ volume	75,000	FEDC cost algorithms
Helium transfer pump system		500,000	Estimate by Oak Ridge National Laboratory, based on similar system for the Large Coil Program
Helium valves		75,000	FEDC cost algorithms
Helium piping		<u>110,000</u>	Similar equipment for the CIT cryogenic system
Total vendor cost		2,116,000	
Procurement cost @15.5%		<u>328,000</u>	CIT percentage
Total hardware cost		2,444,000	
Installation @35% of hardware		860,000	CIT percentage
Engineering @35% of hardware		<u>860,000</u>	CIT percentage
Subtotal		4,164,000	
Contingency @27.5%		<u>1,145,000</u>	CIT percentage
TOTAL COST		\$ 5,309,000

Fig. 3.6. Modifications to OH solenoid electrical insulator to increase heat transfer and flow areas. The outer radial insulator is shown; the inner insulator is also modified.

3.3 HEAT TRANSFER MODEL

3.3.1 Conduction

The heat transfer model used in the helium coolant analysis in the solid region of the TF coil was the same as the liquid nitrogen model previously described. In the solid region of the OH solenoid, conduction in the axial direction is dominated by G-10 insulation. The small conduction effect of Inconel was accounted for by increasing the thickness of the G-10 insulator. Heat transfer in the radial direction is dominated by conductivity of G-10 insulation between turns.

3.3.2 Forced Convection

Correlations for forced-convection heat transfer coefficients for simple gases such as helium are well known. The modified Dittus-Boelter equation, used in this analysis, is expressed as

$$h = 0.023 \frac{G^{0.8}}{D^{0.2}} \frac{C_p^{0.4} k^{0.6}}{\mu^{0.4}} \left(\frac{T_b}{T_f} \right)^{0.8},$$

where

h = convection film conductance (W/m^2-K),

G = gas flow rate per unit area (kg/m^2-s),

D = hydraulic diameter (m),

C_p = specific heat of gas at constant pressure ($J/kg-K$),

k = conductivity of gas ($W/m-K$),

μ = viscosity ($kg/m-s$),

T_b = bulk temperature of gas (K),

T_f = film temperature (K), taken as an average of gas bulk temperature and wall temperature.

Pressure drop was calculated with the Darcy formula by using a friction factor of 0.031. This friction factor is characteristic of smooth pipe with a hydraulic diameter equal to that of the CIT flow channels and a Reynolds number in the range of the cases studied in this report.

A listing of the computer programs used to analyze the TF coil and OH solenoid cooldown, along with typical input data used in the analysis, is shown in Appendixes C and D.

3.4 RESULTS WITH HELIUM COOLANT

Figure 3.7 shows results of cooldown calculations for the helium cooling system described previously. These calculations used a helium system pressurized to 8 atm. Figure 3.7 clearly shows that, for a common pressure drop across the TF coil and OH solenoid in the range of 2 to 4 psi, cooldown time of the two components is comparable and is achieved in about 0.6 h, which is well within the 1-h requirement. Furthermore, the helium flow rate required to achieve this cooldown time is in the range that was assumed in the conceptual design of the closed-loop helium cooling system.

Fig. 3.7. CIT cooldown time with helium coolant at 8 atm pressure.

REFERENCES

1. Compact Ignition Tokamak (CIT) Conceptual Design Report, A-860606-P-01, Princeton Plasma Physics Laboratory, June 6, 1986.
2. E. G. Brentari and R. V. Smith, "Nucleate and Film Pool Boiling Design Correlations for O_2 , N_2 , and He," in Advances in Cryogenic Engineering, ed. K. D. Timmerhaus, Plenum Press, New York, 1965.
3. R. P. Forslund and W. M. Rohsenow, "Dispersed Flow Film Boiling," J. Heat Transfer 90, 399-407 (November 1968).

APPENDIX A

COMPUTER PROGRAM TO ANALYZE LIQUID NITROGEN COOLING OF THE CIT TF COIL INNER LEG

This appendix contains a listing of the computer program, along with sample input data, for calculating the cooldown of the CIT TF coil inner leg with liquid nitrogen. This program was written to run on the National Magnetic Fusion Energy Center computers. Definition of input is provided as comments at the beginning of the program.

```

c Calculate tran temp in TF coils of CIT using liquid nitrogen
c coolant: pool boiling and MIT mist flow-"cittmit"- (metric)
c
c   input variable definition:
c
c   cmts = one line of comments
c
c   rntm = problem maximum run time (s)
c   delt = guess at delta time (s). The code calculates a value
c         based on stability criteria, so input a sufficiently
c         large number
c   pi = print interval (s)
c   coel = interval (s) for recalculating temperature dependent
c         coefficients
c   nx = number of nodes in the x or radial direction. The
c         fluid channel is node 1. An additional node must be
c         provided external to the last solid node. This node
c         may be a boundary condition.
c   ny = number of nodes in the y or z direction. This includes
c         a node at the top and at the bottom external to the
c         solid. These nodes may be used for boundary cond.
c   multip= number of fine mesh elements per coarse mesh element.
c           This must be an odd number. A fine element mesh length
c           of .5 to 1 cm is recommended.
c
c   tsat = saturation temp of coolant (k)
c   wlin = coolant flow rate (kg/s)
c   p = coolant pressure (atm)
c
c   dh = hydraulic diameter (m). 4x(area)/wetted perimeter
c   floa = coolant channel flow area (m2)
c   dadl = heat transfer area per unit length of flow channel
c   haov = heat transfer area per unit volume in coolant channel
c         (m2/m3)
c
c   alen = length of flow channel nodes (m)
c   ff = friction factor in pressure drop equation
c   temptest = minimum change in wall temp that triggers a
c             recalculation of nitrogen heat trans coeffs
c
c the following four sets of input must be read-in as a nx by ny
c (i,j) array, starting at the bottom, or fluid entrance; each line
c of input goes from i=1 to i=nx; line 1 for j=1, line 2 for j=2,
c etc. to j=ny.
c   gh = geometric factor (m2/m) in heat conduction equation.
c         This is the factor in the x or radial direction from
c         the previous node to the present node. the first value
c         (i=1) at each elevation will normally be zero; the
c         second value (i=2) is the heat transfer area between
c         the fluid and wall for the node (m2); the last value
c         at each elevation (i=nx) will normally be zero
c         (indicating an insulated surface) except in a case
c         with an external boundary temperature
c   gv = same as gh, except in the y or z direction; the first
c         two rows (j=1 and 2) will normally be all zeros (there
c         is no conductance from a previous node to the boundary
c         condition nodes external to the solid, and in case of
c         an insulated surface at the bottom, there is no
c         conductance to the first solid nodes); the last row
c         (j=ny) is normally all zeros indicating an insulated
c         surface.
c   wt = weight (kg) for each node; a non-zero value should be

```

```

c          assigned to each node, even though the node is not in
c          a solid region; this prevents a divide-by-zero in the
c          calculation of delta time
c
c          t      = initial temperature (k) for each node
c
c          additional important variables:
c          bigwtd= largest wall temp difference between this pass
c                  and when last nitrogen heat transfer coeffs
c                  were calculated
c          btul  = heat removed (J) through the outer leg (estimated)
c          btur  = heat removed (J) in the fluid channel
c          ch    = conductance(w/k) into the node from the previous node
c                  in the x or radial direction
c          countlp= number of passes thru the solid transient calc
c          counthx= number of passes thru the recalc of nitrogen heat
c                  transfer coeffs
c          cv    = same as ch, except in the y or z direction
c          ownnw = old wall temp minus new wall temp
c          sumflo= integrated liquid nitrogen inlet flow (kg)
c          sumfloo= integrated LN carry over (kg)
c          time  = problem elapsed time (s)
c          tgi   = inlet temperature (k)
c          wcp   = weight times specific heat for each node (J/k)
c
c          variables associated with nodes are indexed as follows:
c          t(i,j), where i is the x or radial index, and j is the
c          y or z index. node (1,1) is the bottom left, or inlet fluid
c          node; node (1,ny) is the exit fluid node at the top left
c
c          CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c
c          dimension t(18,22),ch(18,22),cv(18,22),gh(18,22),gv(18,22),
c          1 wt(18,22),wcp(18,22),cmts(9),x(22)
c          dimension to(22)
c          common t,ch,cv,gh,gv,wt,wcp,x
c          common rntm,delt,pl,coef,tgi,dh,deq,floa,nx,ny
c          common time,btur,btul,btus
c          common tsat,wlin,p,alen,ff,dadl,haov,multip
c          common dpfs,dpms,dpes,deltap,sumflo,sumfloo
c
c          call link('unit5=(ttinnw,open,text),
c          1          unit6=(ttout,create,text),
c          2          unit9=terminal//')
c
c          read data in free format; separate fields with a blank or comma;
c          no decimal is necessary
c
c          read(5,950) (cmts(i),i=1,9)
c          read(5,*) rntm,delt,pl,coef,nx,ny,multip
c          read(5,*) tsat,wlin,p
c          read(5,*) dh,floa,dadl,haov
c          read(5,*) alen,ff,temptest
c          do 4 j=1,ny
c            read(5,*) (gh(i,j),i=1,nx)
c          4 continue
c          do 6 j=1,ny
c            read(5,*) (gv(i,j),i=1,nx)
c          6 continue
c          do 10 j=1,ny
c            read(5,*) (wt(i,j),i=1,nx)

```

```

10 continue
do 20 j=1,ny
  read(5,*) (t(i,j),i=1,nx)
20 continue
950 format(9a8)
c
c set constants and initial values
c
  tgi=tsat
  time=0.
  pt=pi
  coet=coei
  btur=0.
  btul=0.
  btus=0.
  ncoef=0
  sumflo=0.
  sumfloo=0.
  hfg=2.e5
  countlp=0.
  counthx=0.
c
do 41 j=1,ny
do 40 i=1,nx
  ch(i,j)=0.
  cv(i,j)=0.
40 continue
41 continue
c
c write some input values
c
  write(6,951) (cmts(i),i=1,9)
  write(6,952) tsat,wlin
951 format("1",10x,9a8,/)
952 format(" Inlet temp      (K)      ",f10.1,/,
  |      " Inlet flow      (kg/sec)    ",f10.3,/)
c
c calc initial coefficients and print
c
  t(1,1)=tsat
  t(1,ny)=tsat
  t(2,1)=t(2,2)
  t(2,ny)=t(2,ny-1)
  call twoph
  call coefs
  call prnt
c
cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
c write matrix of coefs for debugging
c
do 50 j=1,ny
c  write(6,970) j,(cv(i,j),i=1,nx)
c  write(6,971) j,(ch(i,j),i=1,nx)
c 50 continue
c  write(6,972)
c  do 60 j=1,ny
c  write(6,973) j,(wcp(i,j),i=1,nx)
c973 format(15,10f8.3)
c 60 continue
c  write(6,972)
c970 format(15,4x,10f8.1)

```

```

c971 format(i5,10f8.1)
c972 format(i,i)
c go to 1001
c
cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
c
c start loop to calculate temps
c
c gas stream
c
  65 continue
    t(2,1)=t(2,2)
    t(2,ny)=t(2,ny-1)
    t(1,1)=tsat
c
c calc max change in wall temp since last recalc of nitrogen
c heat transfer parameters; recalc parms if max wall temp change
c exceeds criteria
c
  bigwtd=.
  do 100 j=2,ny-1
    ownnw=abs(to(j)-t(2,j))
    if(bigwtd.lt.ownnw) bigwtd=ownnw
100 continue
    if(bigwtd.gt.temptest) then
      counthx=counthx+1.
      call twoph
      do 105 j=2,ny-1
        to(j)=t(2,j)
105 continue
      end if
c
c calc solid temp changes
c
  do 150 j=2,ny-1
    do 137 i=2,nx-1
      t(i,j)=t(i,j)+delt/wcp(i,j)*(ch(i,j)*(t(i-1,j)-t(i,j))
1      +ch(i+1,j)*(t(i+1,j)-t(i,j))
2      +cv(i,j)*(t(i,j-1)-t(i,j))
3      +cv(i,j+1)*(t(i,j+1)-t(i,j)))
      if(t(i,j).gt.1200..or.t(i,j).lt.0.) go to 900
137 continue
    btul=btul+(ch(10,j)*(t(9,j)-t(10,j)))*delt
150 continue
    sumflo=sumflo+wlin*delt
    sumfloo=sumfloo+wlin*(1.-x(ny))*delt
c
c increment time; decide whether to print and/or recalc coeffs
c
  time=time+delt
  if(t(7,16).le.80.) go to 1000
  countlp=countlp+1.
  if (time.lt.pt) go to 200
  call prnt
  pt=pt+time
200 continue
  if (time.ge.rntm) go to 1000
  if(ncoef.lt.1) then
    call coeffs
    ncoef=ncoef+1
    if(ncoef.eq.1) call prnt

```

```

        go to 65
    end if
    if (time.lt.coet) go to 218
    call coefs
218  continue
    go to 65
c
c  end loop to calc temps
c
c  write error message
c
998  write(6,998) time,t(i,j),i,j
998  format('/'," At time=",f18.4," a temp of",f7.1," occurred at"
1    ", " node",2i6)
1888 continue
    call prnt
    write(6,992) countlp,counthx
992  format('/'," Total number of loops=      ",f12.8,/,
1    " Total two phase calculations= ",f12.8)
1881 stop
    end
c
c  subroutine to calc ht coefs
c
    subroutine coefs
    dimension t(18,22),ch(18,22),cv(18,22),gh(18,22),gv(18,22),
1    wt(18,22),wcp(18,22),cmts(9),x(22)
    common t,ch,cv,gh,gv,wt,wcp,x
    common rntm,delt,pi,coel,tgi,dh,deq,floa,nx,ny
    common time,btur,btul,btus
    common tsat,wlin,p,alen,ff,dadl,haov,multip
    common dpfs,dpms,dpes,deltap,sumfio,sumfloo
c
c  calc temp dependent heat transfer coefficients in solid region
c
    do 51 j=2,ny-1
        do 58 i=3,nx
            ta=(t(i-1,j)+t(i,j))/2.
            ch(i,j)=fkc(ta)*gh(i,j)
58      continue
51      continue
        do 61 j=3,ny-1
            do 68 i=2,nx-1
                ta=(t(i,j-1)+t(i,j))/2.
                cv(i,j)=fkc(ta)*gv(i,j)
68      continue
61      continue
c
c  calc wcp at each node
c
    do 71 j=2,ny-1
        do 78 i=2,nx-1
            wcp(i,j)=wt(i,j)*fcpc(t(i,j))
78      continue
71      continue
c
c  find largest k/wcp at any node to adjust delt for stability
c
    bigsum=8.
    do 81 j=2,ny-1
        do 88 i=2,nx-1

```



```

      sum=(ch(i,j)+ch(i+1,j)+cv(i,j)+cv(i,j+1))/wcp(i,j)
      if(sum.lt.bigsum) go to 88
      bigsum=sum
88  continue
81  continue
82  if(delt*bigsum.gt.8.4) go to 84
      delt=delt*1.1
      go to 82
84  if(delt*bigsum.lt.8.6) go to 85
      delt=delt/1.1
      go to 84
85  continue
      return
      end
      subroutine prnt
c
c  subroutine to print results
c
      dimension t(18,22),ch(18,22),cv(18,22),gh(18,22),gv(18,22),
1  wt(18,22),wcp(18,22),cmts(9),x(22)
      dimension hdg(22),chp(22)
      common t,ch,cv,gh,gv,wt,wcp,x
      common rntm,delt,pi,coef,tgi,dh,deq,floa,nx,ny
      common time,btur,btul,btus
      common tsat,wlin,p,alen,ff,dadl,haov,multip
      common dpfs,dpms,dpes,deltap,sumflo,sumfloo
c
c
      data hdg/" 1"," 2"," 3"," 4"," 5"," 6"," 7"," 8"," 9",
1  " 10"," 11"," 12"," 13"," 14"," 15"/
      data exi/"Exit"/,flo3/"Qual"/
      data hdr2/"pct "/,blk/" "/,flo4/" /188"/,flo5/" h"/
      data hdr3/"q/a "/,hdr4/"w/m2"/,hdr5/"w/m2-k"/
c
      do 12 j=2,ny-1
        chp(j)=ch(2,j)*(t(2,j)-t(1,j))/gh(2,j)/188.
12  continue
      pj=btur/1.e6
      pjr=btul/1.e6
      call delp
c
c  write top info
c
      write(6,1888) time,delt,top,pj,pjr,dpfs,dpms,dpes,deltap,
1  wlin,sumflo,sumfloo
1888 format(///," Time (sec) ",f18.5,/,
1  " Delta time (sec) ",f18.5,/,
2  " Outlet temp (K) ",f18.3,/,
3  " Heat removed (MJ)(gas/leg) ",2f18.1,/,
4  " Press drop (atm) (fr.mom.ele.tot) ",4f18.4,/,
5  " Flow rate (kg/sec) ",f18.3,/,
6  " Integ flow (kg) ",f18.3,/,
7  " Integ liq carry over (kg) ",f18.3,/)
c
c  write header
c
      write(6,1818)
1818 format(48x,"Temperature (K)",/,45x,"i node",/)
      write(6,1828) (blk,i=1,nx-1),exi,hdr3
      write(6,1828) (blk,i=1,nx-1),flo3,flo4,flo5
      write(6,1824) (hdg(i),i=1,nx-1),hdr2,hdr4,hdr5

```

```

        write(6,1025)
1020 format(11x,17a7)
1024 format(11x,17a7)
1025 format(' j node      (Gas)')
        do 21 j=ny-1,2,-1
            write(6,1030) j,(t(i,j),i=1,nx-1),x(j+1)*
            1 1030..chp(j), ch(2,j)/gh(2,j)
        21 continue
1035 format(3x,12,4x,17f7.1)
        return
        end

c
c Calculates total pressure drop from friction,momentum,and elevation
c
        subroutine delp
c
        dimension t(10,22),ch(10,22),cv(10,22),gh(10,22),gv(10,22),
        1 wt(10,22),wcp(10,22),cmts(9),x(22)
        dimension sv(22)
        common t,ch,cv,gh,gv,wt,wcp,x
        common rntm,delt,pl,coef,tgi,dh,deq,floa,nx,ny
        common time,btur,btul,btus
        common tsat,wlin,p,alen,ff,dadl,haov,multip
        common dpfs,dpms,dpes,deltap,sumflo,sumfloo
        data gc/9.75/,sv1/.001237/

c
c function to calc specific vol of nitrogen vapor
c
        fsvv(tin,pin)=tin/341.6/pin

c
c calc a weighted specific volume of mixture
c
        do 10 i=2,ny
            sv(i)=sv1*(1.+x(i)/sv1*(fsvv(t(i,i),p)-sv1))
10        continue
        g=wlin/floa
        gsq=g**2
        dpfs=0.
        dpms=0.
        dpes=0.

c
c calc pressure drop; f=friction, m=momentum, e=elevation
c
        do 20 i=2,ny-1
            dpf=(ff*gsq*(sv(i)+sv(i+1))/2./dh/2./gc)/10353.
            dpm=(gsq*(sv(i+1)-sv(i))/alen/gc)/10353.
            dpe=(2./((sv(i)+sv(i+1))))/10353.
            dpfs=dpfs+dpf
            dpms=dpms+dpm
            dpes=dpes+dpe
20        continue
        dpfs=dpfs*alen
        dpms=dpms*alen
        dpes=dpes*alen
        deltap=dpfs+dpms+dpes
        return
        end

c
c function to calc conductivity of composite at 57.5%/37.5% (metric)
c approximates the curve with straight line segments whose end
c points are given by the t and c matrices

```

```

c      function fkc(tin)
c
c      dimension t(5),c(5)
c      data t/75.,92.,157.,258.,488./
c      data c/538.,268.,288.,238.,238./
c
c      if(tin.lt.t(1)) then
c          fkc=c(1)
c          return
c      else if(tin.ge.t(5)) then
c          fkc=c(5)
c          return
c      end if
c      do 18 i=2,5
c          if(tin.gt.t(i)) go to 18
c          fkc=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)
c      go to 186
c 18 continue
c 186 return
c      end
c
c      function fcpc(tin)
c
c      dimension t(6),c(6)
c      data t/75.,98.,158.,235.,488.,588./
c      data c/178.,238.,325.,398.,428.,448./
c
c      if(tin.lt.t(1)) then
c          fcpc=c(1)
c          return
c      else if(tin.ge.t(6)) then
c          fcpc=c(6)
c          return
c      end if
c      do 18 i=2,6
c          if(tin.gt.t(i)) go to 18
c          fcpc=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)
c      go to 188
c 18 continue
c 188 return
c      end
c
c      Subroutine "twoph" to calc two phase heat transfer. For mist flow
c      region (>18% quality) uses method of Forslund and Rohsenow as
c      reported in J of Heat Transfer, Nov. 1968. For region <18% quality
c      uses pool boiling coefficients. Equations 3-19 below have the same
c      number as in the reference.
c
c      subroutine twoph
c
c      cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
c      c important variables, other than those listed in main program
c
c      c      a3      = part of wall-vapor heat transfer coef
c      c      a4      = part of wall-liquid heat transfer coef
c      c      del     = droplet diam (m)

```



```

coef1=fhn((tw(i)-tsat),p)*(tw(i)-tsat)
coef2=a4*gt**.8*(tw(i)-tsat)
if(coef2.gt.coef1.and.xpe(i).lt..#1.and.(tw(i)-tsat).lt.1#.)
1   coef1=coef2
   qoa(i)=coef1
   tv(i+1)=tv(i)
   dxpeod1=qoa(i)/w/hfg*dad1
   xpe(i+1)=xpe(i)+dlen*dxpeod1
   xa(i+1)=xpe(i+1)
   if(xpe(i+1).gt..#95) then
       nskip1=1
       call incon(tsat,p,xpe(i+1),gt,del#,vv(i+1),vl(i+1))
       del(i+1)=del#
       tv(i+1)=tsat
   end if
   go to 6##

c
c  mist flow calcs
c
58#  rhov=fnrho(tv(i),p)
3   qoavv=a3*gt**.8*fncpkw(tv(i))*(xa(i)+(1.-xa(i))*rhov*vv(i)/8#8.5/
1   vl(i))**.8*(tw(i)-tv(i))
6   hstar=hfg*(1+.##1813*(tw(i)-77.))
   phip=(fnkv(tv(i))**3*hstar*rhov/(tw(i)-77.)/fnmu(tv(i))**.25*
1   9.957
9   qoaw1=.##5572*gt**.6667*((1.-xa(i))/vl(i))**.6667/del(i)**.25*
1   phip*(tw(i)-77.)
   qoawt=qoavv+qoaw1
   qoa(i)=qoawt
8   an3=.##236*gt*(1.-xa(i))/vl(i)/del(i)**3
   tf=(tw(i)+tv(i))/2.
   amu=fnmu(tf)/fnrho(tf,p)
1#  hdelc=fnkv(tf)/del(i)*(2+.55*((vv(i)-vl(i))*del(i)/amu)**.5*
1   fnpr(tf)**.3333)
11  qoade=.318*haov*qoaw1/del(i)**2/an3
12  qoadt=qoade+hdelc*(tv(i)-77.)

c
13  ddod1=-1.237e-#8*qoadt/vl(i)
14  dxod1=-3.*(1.-x#)/del#**3*del(i)**2*ddod1
   dxpeod1=qoawt/w/hfg*dad1
15  dtvod1=(193.*dxpeod1-(116.+tv(i))*dxod1)/xa(i)
   red=gt*(1.-xa(i))*del(i)/.##165
   vsq=(vv(i)-vl(i))**2
   cd1=27./red**.85
   cd2=.384+26.62/red**.8#3
   acc1=.##928*cd1*rhov*vsq/del(i)-9.8
   acc2=.##928*cd2*rhov*vsq/del(i)-9.8
   if(acc1.le.152.) f1=#.
   if(acc1.gt.152..and.acc1.lt.1677.) f1=(acc1-152.)/1677.
   if(acc1.ge.1677.) f1=1.
   if(acc2.le.152.) f2=#.
   if(acc2.gt.152..and.acc2.lt.1677.) f2=(acc2-152.)/1677.
   if(acc2.ge.1677.) f2=1.
   cd=cd2-(cd2-cd1)*(f1+f2)/2.
16  dvod1=(.##9276*cd*rhov*vsq/del(i)-9.8)/vl(i)

c
   del(i+1)=del(i)+dlen*ddod1
   xa(i+1)=xa(i)+ dlen*dxod1
   tv(i+1)=tv(i)+ dlen*dtvod1
   vl(i+1)=vl(i)+ dlen*dvod1
   xpe(i+1)=xpe(i)+dlen*dxpeod1

```

```

18  vv(i+1)=gt*xa(i+1)/fnrho(tv(i+1),p)/(1.-gt*(1.-xa(i+1))/  
1  (888.5*v1(i+1)))  
19  we=fnrho(tv(i+1),p)*(vv(i+1)-v1(i+1))*2*del(i+1)/9.e-3  
    if(we.gt.7.5) then  
        del8=.794*del(i+1)  
        x8=xa(i+1)  
        del(i+1)=del8  
    end if  
    if(xa(i+1).ge.1.) then  
        va(i+1)=1.  
        nskip2=1  
        den8=fnrho(tv(i+1),p)  
        vv8=vv(i+1)  
    end if  
    go to 688  
c  
c  vapor only (no liquid) calcs  
c  
598  qoa(i)=a3*gt**.8*fncpk8(tv(i))*(tw(i)-tv(i))  
      dtvodl=qoa(i)/w/cpg*dadl  
      tv(i+1)=tv(i)+dtvodl*dlen  
      v1(i+1)=8.  
      vv(i+1)=vv8*den8/fnrho(tv(i+1),p)  
      xa(i+1)=1.  
688  continue  
618  continue  
c  
c  translate fine mesh parameters into coarse mesh  
c  
      btur1=8.  
      do 785 j=2,ny-1  
          ttva=8.  
          tqoa=8.  
          k1=(j-2)*multip+1  
          k2=k1+multip  
          do 788 k=k1,k2  
              ttva=ttva+tv(k)  
              tqoa=tqoa+qoa(k)  
788      continue  
          t(1,j)=ttva/(fmul+1.)  
          tqoa=tqoa/(fmul+1.)  
          btur1=btur1+tqoa*gh(2,j)  
          ch(2,j)=tqoa/(t(2,j)-t(1,j))*gh(2,j)  
          x(j)=xa(k1)  
785      continue  
      btur=btur+btur1*(time-oldtime)  
      oldtime=time  
      x(ny)=xa(in)  
      return  
      end  
c  
      subroutine incon(tin,pin,xi,gt,del,vv,v1)  
c  calculates initial conditions for MIT mist flow  
c  
cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
c  External to this program set We=7.5; solve equations 16 and 19  
c  simultaneously for velocity difference, delv, and droplet diam,  
c  del, both as functions of cd, the drag coeff. Calc Re as a  
c  function of cd, then as a function of del from above. Write  
c  the equation of curve for cd, a function of Re in Fig 7.  
c  Then in the code below iteratively solve for a Re and cd that

```



```

dimension t(18,22),tw(588)
data ncount/8/

c
  if(ncount.eq.8) then
    fmul=float(multip)
    multil=int(multip/2)+1
    multp2=multil+1
    multm1=multil-1
    multm2=multil
    ncount=1
  end if

c
  if(multip.gt.1) goto 288
do 168 j=2,ny-1
  tw(j-1)=t(2,j)
168 continue
nf=ny-2
return

c
288 k=8
do 328 j=2,ny-1
  deltm=(t(2,j)-t(2,j-1))/fmul
  do 268 l=1,multil
    k=k+1
    tw(k)=t(2,j-1)+deltm*float(1+multm1)
268 continue
  deltm=(t(2,j+1)-t(2,j))/fmul
  do 318 l=multp2,multip
    k=k+1
    tw(k)=t(2,j)+deltm*float(1-multm2)
318 continue
328 continue
nf=k
return
end

c
c Function to cal the pool boiling heat transfer coef of
c nitrogen (metric); from "Nucleate and Film Boiling Design
c Correlations for O2, N2, H2, and He", by Brentari and Smith,
c CELD National Bureau of Standards, Boulder Colorado.
c "Int. Adv. Cryog. Eng., Proc. 1964 Cryo. Eng Conf.", K. D.
c Timmerhaus, Ed., Plenum Press, New York, 1965
c
  function fhn(tin,p)
c
  if(tin.le.12.) then
    h1=458.*p*tin**1.4
    h2=47.3e6/tin**3.22
    if(h2.lt.h1) h1=h2
    fhn=h1
    return
  end if
  if(tin.le.58.) then
    fhn=47.3e6/tin**3.22
    return
  end if
  fhn=168.
  return
end

```


Baseline design, mit flow model, 16 kg/s flow

9888. 5. 388. 188. 18 22 21

77. 16. 1.

.82 .8375 3.78 188.

.183 .831 .5

8 8 8 8 8 8 8 8 8 8
8 .692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.
8 .692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.
8 .692 17.13 18.28 19.4 28.6 21.72 13.39 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 8.8 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 13.39 8.8 8.
8 .692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.
8 .692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.

8 8 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8
8 .439 .924 .984 1.846 1.184 1.164 2.838 7.379 8.
8 .439 .924 .984 1.846 1.184 1.164 2.838 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 8.8 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 2.838 8.8 8.
8 .439 .924 .984 1.846 1.184 1.164 2.838 7.379 8.

8 8 8 8 8 8 8 8 8 8
8 .128.8 254. 278.4 287.5 383.4 319.9 779.9 2827.7 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 779.9 2827.7 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 779.9 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.
8 .128.8 254. 278.4 287.5 383.4 319.9 1. 1. 8.

#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	1.	1.	#.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	1.	1.	#.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	1.	1.	#.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	1.	1.	#.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	1.	1.	#.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	779.9	1.	#.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	779.9	2#	27.7 #.
#.	12#	8	254.	27#	4	287.5	3#	3.4	319.9	779.9	2#	27.7 #.
#	#	#	#	#	#	#	#	#	#	#	#	#
1	1	1	1	1	1	1	1	1	1	1	1	1
77	82	82	83	84	85	86	88	9#	79			
77	94	95	96	97	97	98	99	123	79			
77	125	125	125	127	135	14#	15#	1	1			
77	22#	22#	22#	22#	225	245	1	1	1			
77	275	275	275	278	28#	278	1	1	1			
77	283	282	282	281	279	285	1	1	1			
77	291	29#	289	288	288	292	1	1	1			
77	298	297	296	296	297	298	1	1	1			
77	3#6	3#5	3#5	3#5	3#9	312	1	1	1			
77	315	315	314	315	322	328	1	1	1			
77	315	315	314	315	322	328	1	1	1			
77	3#6	3#5	3#5	3#5	3#9	312	1	1	1			
77	298	297	296	296	297	298	1	1	1			
77	291	29#	289	288	288	292	1	1	1			
77	283	282	282	281	279	285	1	1	1			
77	275	275	275	278	28#	278	1	1	1			
77	22#	22#	22#	22#	225	245	1	1	1			
77	125	125	125	127	135	14#	15#	1	1			
77	94	95	96	97	97	98	99	123	79			
77	82	82	83	84	85	86	88	9#	79			
1	1	1	1	1	1	1	1	1	1			

APPENDIX B

MIST-FLOW HEAT TRANSFER CORRELATIONS APPLIED TO THE CIT

This appendix describes modifications to published work on mist- or dispersed-flow film-boiling heat transfer to adapt the theoretical methods to CIT calculations. In ref. 1, Forslund and Rohsenow provide test data for dispersed-flow film boiling of nitrogen in various tube diameters and mass velocities that are typical of CIT conditions. They also present a theoretical basis for calculating heat transfer parameters in this flow regime. The test data were obtained from electrically heated tubes with constant heat flux that was uniform along the tube length. The calculational procedure given by these investigators is based on calculating a wall temperature that corresponds to a known heat flux.

To adapt the ref. 1 method to CIT calculations, the procedure was modified to calculate a heat flux that corresponds to a known wall temperature. This change resulted in a less complex procedure wherein known wall temperatures and vapor temperatures are used directly in Eqs. (3) and (9) of ref. 1 to calculate heat flux from wall to vapor and from wall to liquid. The remainder of the ref. 1 procedure was used without modification.

Forslund and Rohsenow obtained an arbitrary set of constants, K_1K_2 , to fit results of their recommended calculational procedure to their test data. To determine whether these same constants are applicable to the calculational procedure used for CIT, several cases were run to find a K_1K_2 that gave results that fit the ref. 1 data best. From the results, a K_1K_2 of 0.4 appears to be an optimum value, in contrast to a K_1K_2 of 0.2 found by Forslund and Rohsenow. In addition to this change, the exponent in Eq. (6) for calculating an effective heat of vaporization was changed to 1, and the constant in Eq. (3) was changed from 0.019 to 0.023. Figures B.1 through B.5 compare ref. 1 data with calculations using the revised constants. They also show the effect of assuming pool-boiling film coefficients in the tube prior to reaching 10% quality.

Fig. B.1. Comparison of ref. 1 test data with calculations with best-fit constants in the analytical solution for a heat flux of $15,750 \text{ W/m}^2$ and a flow rate of 95 kg/s-m^2 .

Fig. B.2. Comparison of ref. 1 test data with calculations with best-fit constants in the analytical solution for a heat flux of $47,250 \text{ W/m}^2$ and a flow rate of 95 kg/s-m^2 .

Fig. B.3. Comparison of ref. 1 test data with calculations with best-fit constants in the analytical solution for a heat flux of $15,750 \text{ W/m}^2$ and a flow rate of 174 kg/s-m^2 .

Fig. B.4. Comparison of ref. 1 test data with calculations with best-fit constants in the analytical solutions for a heat flux of $31,140 \text{ W/m}^2$ and a flow rate of 258 kg/s-m^2 .

Fig. B.5. Comparison of ref. 1 test data with calculations with best-fit constants in the analytical solution for a heat flux of $78,750 \text{ W/m}^2$ and a flow rate of 258 kg/s-m^2 .

As shown in these figures, the analytical results are in good agreement with experimental data, particularly at low heat fluxes that are typical of CIT heat removal. From these results, it is concluded that this method can be used with confidence for CIT calculations.

REFERENCE

1. R. P. Forslund and W. M. Rohsenow, "Dispersed Flow Film Boiling," J. Heat Transfer 90, 399-407 (November 1968).

APPENDIX C

COMPUTER PROGRAM TO ANALYZE HELIUM COOLING OF THE CIT TF COIL INNER LEG

This appendix contains a listing of the computer program, along with sample input data, for calculating the cooldown of the CIT TF coil inner leg with gaseous helium. This program was written to run on the National Magnetic Fusion Energy Center computers. Definition of input is provided as comments at the beginning of the program.


```

c   Program to calculate tran temp in tf coils of CIT
c   with 57.5/37.5 pct composite (metric) helium cooled-"cithb"
c
c   input variable definition:
c
c   cmts   = one line of comments
c
c   rntm   = problem maximum run time (s)
c   delt   = guess at delta time (s). The code calculates a value
c           based on stability criteria, so input a sufficiently
c           large number, say 5 sec
c   pi     = print interval (s)
c   coel   = interval (s) for recalculating temperature dependent
c           coefficients (188 sec is reasonable)
c   nx     = number of nodes in the x or radial direction. The
c           fluid channel is node 1. An additional node must be
c           provided external to the last solid node. This node
c           may be a boundary condition.
c   ny     = number of nodes in the y or z direction. This includes
c           a node at the top and at the bottom external to the
c           solid. These nodes may be used for boundary cond.
c
c   wg     = helium flow rate (kg/s)
c   dh     = hydraulic diameter (m). 4x(area)/wetted perimeter
c   floa   = helium channel flow area (m2)
c   tgi    = helium inlet temperature (k)
c   press  = helium inlet pressure (atm)
c
c   alen   = length of flow channel nodes (m)
c   ff     = friction factor in pressure drop equation
c   whx    = total helium flow (kg/s) to the external hx equipment.
c           This is used to calculate a cit inlet temperature.
c   dtrhx  = fixed temperature rise (k) between cit and the helium-
c           liquid nitrogen hx
c
c   the following four sets of input must be read-in as a nx by ny
c   (i,j) array, starting at the bottom, or fluid entrance; each line
c   of input goes from i=1 to i=nx; line 1 for j=1, line 2 for j=2,
c   etc. to j=ny.
c   gh     = geometric factor (m2/m) in heat conduction equation.
c           This is the factor in the x or radial direction from
c           the previous node to the present node; the first value
c           (i=1) at each elevation will normally be zero; the
c           second value (i=2) is the heat transfer area between
c           the fluid and wall for the node (m2); the last value
c           at each elevation (i=nx) will normally be zero
c           (indicating an insulated surface) except in a case
c           with an external boundary temperature
c   gv     = same as gh, except in the y or z direction; the first
c           two rows (j=1 and 2) will normally be all zeros (there
c           is no conductance from a previous node to the boundary
c           condition nodes external to the solid, and in case of
c           an insulated surface at the bottom, there is no
c           conductance to the first solid nodes); the last row
c           (j=ny) is normally all zeros indicating an insulated
c           surface.
c   wt     = weight (kg) for each node; a non-zero value should be
c           assigned to each node, even though the node is not in
c           a solid region; this prevents a divide-by-zero in the
c           calculation of delta time
c   t      = initial temperature (k) for each node
c

```

```

c      additional important variables:
c      ch      = conductance(w/k) into the node from the previous node
c               in the x or radial direction
c      cv      = same as ch, except in the y or z direction
c      wcp     = weight times specific heat for each node (j/k,
c      btur    = heat removed (j) in the fluid channel
c      btul    = heat removed (j) through the outer leg
c      presp   = pressure (a) at each fluid node
c      rhoh    = density at each fluid node (kg/m3)
c      to      = outlet temperature from the fluid node
c      time    = problem elapsed time (s)
c      xcoef   = collection of constants in the heat transfer equation
c      delpl   = collection of constants in the pressure drop equation
c
c      variables associated with nodes are indexed as follows:
c      t(i,j), where i is the x or radial index, and j is the
c      y or z index; node (1,1) is the bottom left, or inlet fluid
c      node; node (1,ny) is the exit fluid node at the top left
c
c      cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
c
c      dimension t(10,22),ch(10,22),cv(10,22),gh(10,22),gv(10,22),
c      1 wt(10,22),wcp(10,22),cmts(9),presp(22),rhoh(22)
c      common t,ch,cv,gh,gv,wt,wcp,presp,rhoh,vel,btul
c      common rntm,delt,pi,coef,tgi,dh,wg,floa,xcoef,nx,ny
c      common to,time,btur,vg,press,deipl,whx,dtrhx
c
c      call link("unit5=(ttinhb,open,text),
c      1          unit6=(ttout,create,text),
c      2          unit9=terminal//")
c
c      Read data in free format;separate fields with a blank or comma;
c      no decimal is necessary
c
c      read(5,950) (cmts(i),i=1,9)
c      read(5,*) rntm,delt,pi,coef,nx,ny
c      read(5,*) wg,dh,floa,tgi,press
c      read(5,*) alen,ff,whx,dtrhx
c      do 4 j=1,ny
c        read(5,*) (gh(i,j),i=1,nx)
c      4 continue
c      do 6 j=1,ny
c        read(5,*) (gv(i,j),i=1,nx)
c      6 continue
c      do 10 j=1,ny
c        read(5,*) (wt(i,j),i=1,nx)
c      10 continue
c      do 20 j=1,ny
c        read(5,*) (t(i,j),i=1,nx)
c      20 continue
c      950 format(9a8)
c
c      Set constants and initial values
c
c      xcoef=.023/dh**.2*(wg/floa)**.8
c      delpl=1./(2.*9.8)/dh*ff*alen/10350.
c      time=0.
c      to=tgi
c      pt=pi
c      coet=coef
c      btur=0.

```

```

      btul=0.
      ncoef=0
c
      do 41 j=1,ny
      do 40 i=1,nx
        ch(i,j)=0.
        cv(i,j)=0.
      40 continue
      presp(j)=press
      41 continue
c
c   Write some input values
c
      write(6,951) (cms(i),i=1,9)
      vg=wg/floa*.0205*tgl/press
      write(6,952) wg,vg,tgl,press
      951 format('1',10x,9a8,/)
      952 format(' Gas flow rate (kg/sec) ',f10.2,/,
        1      ' Gas velocity (m/sec) ',f10.2,/,
        2      ' Initial inlet temp (K) ',f10.1,/,
        3      ' Gas inlet pressure (atm) ',f10.3,/)
c
c   Calc initial coefficients and print results
c
      call coefs
      call coefs
      call coefs
      call prnt
c   *****
c   Write matrix of coefficients for debugging
c
      do 50 j=1,ny
      write(6,970) j,(cv(i,j),i=1,nx)
      write(6,971) j,(ch(i,j),i=1,nx)
      50 continue
      write(6,972)
      do 60 j=1,ny
      write(6,973) j,(wcp(i,j),i=1,nx)
      973 format(15,10f8.3)
      60 continue
      write(6,972)
      970 format(15,4x,10f8.1)
      971 format(15,10f8.1)
      972 format(//)
      go to 1001
c   *****
c   start loop to calculate temps
c
c   gas stream
c
      65 to=tgl
      do 111 j=2,ny-1
        ch(2,j)=gh(2,j)*xcoef*fcphmh(to)
        ch(2,j)=gh(2,j)*xcoef*fcphmh(to)*(2.*t(1,j)/
          1      (t(1,j)+t(2,j))**.8
        h1=2.*wg*fcph(t(1,j))
        t(1,j)=(ch(2,j)*t(2,j)+h1*to)/(ch(2,j)+h1)
        to1=2.*t(1,j)-to
        if(t(1,j).gt.1200..or.t(1,j).lt.0.) go to 900
        btur=btur+h1/2.*(to1-to)*delt

```

```

      to=toi
c
c  solid temps
c
      do 118 i=2,nx-1
        t(i,j)=t(i,j)+delt/wcp(i,j)*(ch(i,j)*(t(i-1,j)-t(i,j))
1         +ch(i+1,j)*(t(i+1,j)-t(i,j))
2         +cv(i,j)*(t(i,j-1)-t(i,j))
3         +cv(i,j+1)*(t(i,j+1)-t(i,j)))
        if(t(i,j).gt.1288..or.t(i,j).lt.8.) go to 988
118      continue
      btul=btul+(ch(18,j)*(t(9,j)-t(18,j)))*delt
111      continue
c
c  increment time: decide whether to print and/or recalculate coeffs
c
      time=time+delt
      if(t(7,16).le.88.) go to 1888
      if (time.lt.pt) go to 288
      call coefs
      call prnt
      pt=pt+time
288      continue
      if (time.ge.rntm) go to 1888
      if(ncoef.lt.18) then
c
c975      write(6,975) tgi,to
      format(/," Inlet temp = ",f18.1,"      Outlet temp = ",f18.1)
      call coefs
      ncoef=ncoef+1
      if(ncoef.eq.18) call prnt
      go to 65
      end if
      if (time.lt.coet) go to 218
      call coefs
218      continue
      go to 65
c
c  end loop to calc temps
c
c
c  write error message
c
c988      write(6,988) time,t(i,j),i,j
c988      format(/," At time=",f18.4," a temp of",f7.1," occurred at"
1      ," node",2i6)
1888      call coefs
      call prnt
1881      stop
      end
c
c  subroutine to calc ht coefs
c
c      subroutine coefs
c
      dimension t(18,22),ch(18,22),cv(18,22),gh(18,22),gv(18,22),
1      wt(18,22),wcp(18,22),cmts(9),presp(22),rhoh(22)
      common t,ch,cv,gh,gv,wt,wcp,presp,rhoh,vel,btul
      common rntm,delt,pi,coel,tgi,dh,wg,floa,xcoef,nx,ny
      common to,time,btur,vg,press,delpi,whx,dtrhx
c
c  Find new Inlet temperature

```

```

c      trhx=to+dtrhx
      call deltatem(trhx,whx,tgl)
c
c      Calc pressure at each node
c
      presp(1)=press
      gsq=(wg/floa)**2
      do 51 j=2,ny-1
        rhoh(j)=48.78/(2.*t(1,j))*(presp(j-1)+presp(j))
        presp(j)=presp(j-1)-delp1*gsq/rhoh(j)
        if(presp(j).le.0.) then
          write(9,999)
999      format(//," Adjust gas inlet conditions, program terminated")
          stop
        end if
c
c      Calc new temp dependent coefficients
c
      ch(2,j)=gh(2,j)*xcoef*fcphmh(to)
c      ch(2,j)=gh(2,j)*xcoef*fcphmh(to)*(2.*t(1,j)/
c      1      (t(1,j)+t(2,j)))**.8
      do 50 i=3,nx
        ta=(t(i-1,j)+t(i,j))/2.
        ch(i,j)=fkc(ta)*gh(i,j)
50      continue
51      continue
      do 61 j=3,ny-1
        do 60 i=2,nx-1
          ta=(t(i,j-1)+t(i,j))/2.
          cv(i,j)=fkc(ta)*gv(i,j)
60      continue
61      continue
c
c      calc wcp at each node
c
      do 71 j=2,ny-1
        do 70 i=2,nx-1
          wcp(i,j)=wt(i,j)*fcpc(t(i,j))
70      continue
71      continue
c
c      find largest k/wcp at any node to adjust delt for stability
c
      bigsum=0.
      do 81 j=2,ny-1
        do 80 i=2,nx-1
          sum=(ch(i,j)+ch(i+1,j)+cv(i,j)+cv(i,j+1))/wcp(i,j)
          if(sum.lt.bigsum) go to 80
          bigsum=sum
80      continue
81      continue
84      if(delt*bigsum.lt.0.5) go to 85
          delt=delt/1.2
          go to 84
85      continue
      return
      end
      subroutine prnt
c
c      subroutine to print results

```

```

c      dimension t(18,22),ch(18,22),cv(18,22),gh(18,22),gv(18,22),
1      wt(18,22),wcp(18,22),cmts(9),presp(22),rhoH(22)
      common t,ch,cv,gh,gv,wt,wcp,presp,rhoH,vel,btul
      common rntm,delt,p,ccei,tgi,dh,wg,fioa,xcoef,nx,ny
      common to,time,btur,vg,press,delp1,whx,dtrhx

c      delp=presp(1)-presp(ny-1)
      pj=btur*1.e-6
      pjr=btul*1.e-6

c
c      write top info
c
      write(6,1888) time,delt,tgi,to,wg,pj,pjr,delp
1888 format(//," Time (s) ",f18.5,/,
1      " Delta time (s) ",f18.5,/,
2      " Gas inlet temp (K) ",f18.3,/,
3      " Gas outlet temp (K) ",f18.3,/,
4      " Gas flow rate (kg/s) ",f18.3,/,
5      " Heat removed (MJ)(Gas/Leg) ",2f18.1,/,
6      " Press drop (atm) ",f18.3,/)

c      write header
c
      write(6,1818)
1818 format(48x,"Temperature (K)",/,45x,"i node",/)
      write(6,1828) (i,i=1,nx-1)
      write(6,1825)
1825 format(" j node (Gas)")
1828 format(7x,1517)

c      write temperature matrix
c
      do 21 j=ny-1,2,-1
        write(6,1838) j,(t(i,j),i=1,nx-1)
21      continue
1838 format(3x,12,4x,15f7.1)
      return
      end

c      function to calc heat xtf correlation for He (metric)
c      (cp^.4*k^.6/mu^.4)
c
      function fcpkmh(tin)
c
      fcpkmh=616.+422*tin
      return
      end

c      function to calc specific heat of He (j/kg-k)
c
      function fcph(tin)
c
      fcph=5258.
      return
      end

c
c      Function to calc conductivity of composite at 57.5%/37.5% (w/m-k)
c      Approximates the curve with straight line segments whose end
c      points are given by the t and c matrices.
c

```

```

function fkc(tin)
c
dimension t(5),c(5)
data t/75.,90.,150.,250.,400./
data c/530.,360.,280.,230.,230./
c
if(tin.lt.t(1)) then
    fkc=c(1)
    return
else if(tin.ge.t(5)) then
    fkc=c(5)
    return
end if
do 10 i=2,5
    if(tin.gt.t(i)) go to 10
    fkc=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)
go to 100
10 continue
100 return
end

c
c Function to cal specific heat of composite at 57.5X/37.5X (j/kg-k).
c Approximates the curve with straight line segments whose end
c points are given by the t and c matrices
c
function fcpc(tin)
c
dimension t(6),c(6)
data t/75.,90.,150.,235.,400.,500./
data c/170.,230.,325.,390.,420.,440./
c
if(tin.lt.t(1)) then
    fcpc=c(1)
    return
else if(tin.ge.t(6)) then
    fcpc=c(6)
    return
end if
do 10 i=2,6
    if(tin.gt.t(i)) go to 10
    fcpc=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)
go to 100
10 continue
100 return
end

c
c Subroutine to find CIT inlet temp. given outlet temp and
c total CIT helium flow rate. Data is based on other calcs
c
subroutine deltatem(tin,flow,tout)
dimension tbrk(5),c(6,5)
data tbrk/175.,200.,250.,300.,350./
data c/
1 77.2 ,77.2 ,77.2 ,77.2 ,77.2 ,77.2 ,
2 77.2 ,77.2 ,77.5 ,81. ,85. ,94. ,
3 77.2 ,77.5 ,86. ,104. ,140. ,147. ,
4 77.2 ,85. ,109. ,147. ,162. ,175. ,
77.2 ,91. ,144. ,167. ,185. ,200./
c
if(tin.le.175.) then
    tout=77.2
    return

```

```

end if
n=nint(flow)/2-1
do 10 i=2,6
  if(tin.gt.tbrk(i)) go to i8
  tout=(tin-tbrk(i-1))*(c(n,i)-c(n,i-1))/(tbrk(i)-tbrk(i-1))+
1    c(n,i-1)
  go to 100
10  continue
  tout=c(n,5)
100 return
end

```


Baseline design, new temp dist, 4 kg/s flow

8888. 5. 144. 36. 18 22

4. .82 .8375 118. 8.

.183 .82! 12. 12.

.692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.
.692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.
.692 17.13 18.28 19.4 28.6 21.72 13.39 #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 #.# #.# #.
.692 17.13 18.28 19.4 28.6 21.72 13.39 #.# #.
.692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.
.692 17.13 18.28 19.4 28.6 21.72 13.39 5.28 1.

.439 .924 .984 1.846 1.184 1.164 2.838 7.379 #.
.439 .924 .984 1.846 1.184 1.164 2.838 #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 #.# #.# #.
.439 .924 .984 1.846 1.184 1.164 2.838 #.# #.
.439 .924 .984 1.846 1.184 1.164 2.838 7.379 #.

.128.8 254. 278.4 287.5 383.4 319.9 779.9 2827.7 #.
.128.8 254. 278.4 287.5 383.4 319.9 779.9 2827.7 #.
.128.8 254. 278.4 287.5 383.4 319.9 779.9 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.
.128.8 254. 278.4 287.5 383.4 319.9 1. 1. #.

#.	12#	.8	254.	27#	.4	287.5	3#3.4	319.9	1.	1.	#.
#.	12#	.8	254.	27#	.4	287.5	3#3.4	319.9	1.	1.	#.
#.	12#	.8	254.	27#	.4	287.5	3#3.4	319.9	1.	1.	#.
#.	12#	.6	254.	27#	.4	287.5	3#3.4	319.9	1.	1.	#.
#.	12#	.8	254.	27#	.4	287.5	3#3.4	319.9	779.9	1.	#.
#.	12#	.8	254.	27#	.4	287.5	3#3.4	319.9	779.9	2#27.7	#.
#.	12#	.8	254.	27#	.4	287.5	3#3.4	319.9	779.9	2#27.7	#.

#	#	#	#	#	#	#	#	#	#	#	#
1	1	1	1	1	1	1	1	1	1	1	1
77	82	82	83	84	85	86	88	9#	79		
77	94	95	96	97	97	98	99	123	79		
77	125	125	125	127	135	14#	15#	1	1	1	
77	22#	22#	22#	22#	225	245	1	1	1	1	
77	275	275	275	278	28#	278	1	1	1	1	
77	283	282	282	281	279	285	1	1	1	1	
77	291	29#	289	288	288	292	1	1	1	1	
77	298	297	296	296	297	298	1	1	1	1	
77	3#6	3#5	3#5	3#5	3#9	312	1	1	1	1	
77	315	315	314	315	322	328	1	1	1	1	
77	315	315	314	315	322	328	1	1	1	1	
77	3#6	3#5	3#5	3#5	3#9	312	1	1	1	1	
77	298	297	296	296	297	298	1	1	1	1	
77	291	29#	289	288	288	292	1	1	1	1	
77	283	282	282	281	279	285	1	1	1	1	
77	275	275	275	278	28#	278	1	1	1	1	
77	22#	22#	22#	22#	225	245	1	1	1	1	
77	125	125	125	127	135	14#	15#	1	1	1	
77	94	95	96	97	97	98	99	123	79		
77	82	82	83	84	85	86	88	9#	79		
1	1	1	1	1	1	1	1	1	1	1	

APPENDIX D

COMPUTER PROGRAM TO ANALYZE HELIUM COOLING OF THE CIT OH SOLENOID

This appendix contains a listing of the computer program, along with sample input data, for calculating the cooldown of the CIT OH solenoid with gaseous helium. This program was written to run on the National Magnetic Fusion Energy Center computers. Definition of input is provided as comments at the beginning of the program. A listing of a BASIC program to prepare most of the input data for the OH solenoid is also shown.

```

c Calculate tran temp in OH coil of CIT, Helium
c This calculation is based in two helium streams
c flowing along the vertical edges of a cylinder.
c
c      input variable definition:
c
c      cmts = one line of comments
c
c      rntm = problem maximum run time (s)
c      delt = guess at delta time (s). The code calculates a value
c             based on stability criteria, so input a sufficiently
c             large number to start the calc (say 5 sec)
c      pi    = print interval (s)
c      coe1  = interval (s) for recalculating temperature dependent
c             coefficients (100 sec is reasonable)
c      nx    = number of nodes in the x or radial direction. The
c             fluid channel is node 1. An additional node must be
c             provided external to the last solid node. This node
c             may be a boundary condition.
c      ny    = number of nodes in the y or z direction. This includes
c             a node at the top and at the bottom external to the
c             solid. These nodes may be used for boundary cond.
c
c      wlin  = helium flow rate (kg/s)-inner
c      dh    = hydraulic diam (m).  $4 \times (\text{area}) / \text{wetted perimeter}$ -inner
c      floa  = helium channel total flow area (m2)-inner
c      alen1 = length of each flow channel node (m)-pf1
c      npf1  = total number of pancakes in pf1
c
c      wlin2 = helium flow rate (kg/s)-outer
c      dh2   = hydraulic diam (m).  $4 \times (\text{area}) / \text{wetted perimeter}$ -outer
c      floa2 = helium channel total flow area (m2)-outer
c      alen2 = length of each flow channel node (m)-pf2
c      npf2  = number of pancakes in each pf2
c
c      ff    = friction factor in pressure drop equation
c      tgi   = initial inlet He temperature (k)
c      press = helium pressure (atm)
c      whx   = total helium flow (kg/s) to the external hx equipment.
c             This is used to calculate a cit inlet temperature.
c      dtrhx = fixed temperature rise (k) between cit and the helium-
c             liquid nitrogen hx
c
c      vfcu  = volume fraction of copper in the turn
c      vfin  = volume fraction of Inconel in the turn
c the following four sets of input must be read-in as a nx by ny
c (i,j) array, starting at the bottom, or fluid entrance; each line
c of input goes from i=1 to i=nx; line 1 for j=1, line 2 for j=2,
c etc, to j=ny.
c      gh    = geometric factor (m2/m) in heat conduction equation.
c             This is the factor in the x or radial direction from
c             the previous node to the present node; the first value
c             (i=1) at each elevation will normally be zero; the
c             second value (i=2) is the heat transfer area between
c             the fluid and wall for the node (m2); the last value
c             at each elevation (i=nx) will normally be zero
c             (indicating an insulated surface) except in a case
c             with an external boundary temperature
c      gv    = same as gh, except in the y or z direction; the first
c             two rows (j=1 and 2) will normally be all zeros (there
c             is no conductance from a previous node to the boundary
c             condition nodes external to the solid, and in case of

```

```

c          an insulated surface at the bottom, there is no
c          conductance to the first solid nodes); the last row
c          (j=ny) is normally all zeros indicating an insulated
c          surface.
c          wt      = weight (kg) for each node; a non-zero value should be
c                   assigned to each node, even though the node is not in
c                   a solid region; this prevents a divide-by-zero in the
c                   calculation of delta time
c          t        = initial temperature (k) for each node
c
c          additional important variables:
c          ch        = conductance(w/k) into the node from the previous node
c                   in the x or radial direction
c          cv        = same as ch, except in the y or z direction
c          wcp       = weight times specific heat for each node (j/k)
c          btus      = heat removed (j) in the fluid channel-inner
c          btus2     = heat removed (j) in the fluid channel-outer
c          to,to2    = outlet temperature from the fluid node-inner,outer
c          time      = problem elapsed time (s)
c          xcoef     = collection of constants in the HX equation-inner
c          xcoef4    = collection of constants in the HX equation-outer
c          m1        = node number separating lower pf2 from pf1
c          m2        = node number separating pf1 from upper pf2
c          m3        = node number at top of upper pf2
c
c          variables associated with nodes are indexed as follows:
c          t(i,j), where i is the x or radial index, and j is the
c                   y or z index; node (1,1) is the bottom left, or inlet fluid
c                   node; node (1,ny) is the exit fluid node at the top left;
c                   node (1,nx) is the outer inlet fluid node;
c                   node (ax,ny) is the outer exit fluid node;
c                   there are twice the number of axial nodes as pancakes;
c                   j is even for copper conductor rows, and odd for the copper/
c                   inconel turn.
c
cXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c          dimension t(18,488),ch(18,488),cv(18,488),gh(18,488),gv(18,488),
c          1 wt(18,488),wcp(18,488),cmts(9),alen(488)
c
c          common t,ch,cv,gh,gv,wt,wcp,alen
c          common rntm,delt,time,nx,ny
c          common ff,tgi,press,whx,dtrhx
c          common dpfs,dps,dpes,deltap
c          common wlin,dh,floa,to,btus,sumflo
c          common wlin2,dh2,floa2,to2,btus2,sumflo2
c          common /cpc:1/vfcu,vfin,vfci
c
c          call link("unit5=(ttinohh,open,text),
c          1          unit6=(ttout,create,text),
c          2          unit9=terminal//")
c
c          read data in free format; separate fields with blanks
c          or commas; decimals are optional
c
c          read(5,958) (cmts(i),i=1,9)
c          read(5,*) rntm,delt,pi,coef,nx,ny
c          read(5,*) wlin,dh,floa,alen1,npf1
c          read(5,*) wlin2,dh2,floa2,alen2,npf2
c          read(5,*) ff,tgi,press,whx,dtrhx
c          read(5,*) vfcu,vfin

```

```

      do 4 j=1,ny
        read(5,*) (gh(i,j),i=1,nx)
4      continue
      do 6 j=1,ny
        read(5,*) (gv(i,j),i=1,nx)
6      continue
      do 18 j=1,ny
        read(5,*) (wt(i,j),i=1,nx)
18     continue
      do 28 j=1,ny
        read(5,*) (t(i,j),i=1,nx)
28     continue
958    format(9a8)
c
c  set constants and initial values
c
      xcoef = .023/dh **.2*(wlin /f1oa )**.8
      xcoef4 = .023/dh2 **.2*(wlin2/f1oa2) **.8
      time=8.
      to=sgi
      to2=sgi
      pt=pi
      coet=coef
      btus=8.
      btus2=8.
      ncoef=8
      sumflo=8.
      sumflo2=8.
      vfcu=vfcu*8958.
      vfin=vfin*8168.
      vfc1=vfcu+vfin
      m1=2*npf2
      m2=m1+2*npf1
      m3=m2+2*npf2
      do 25 j=2,m1,2
        alen(j)=alen2
25     continue
      do 26 j=m1+2,m2,2
        alen(j)=alen1
26     continue
      do 27 j=m2+2,m3,2
        alen(j)=alen2
27     continue
c
      do 41 j=1,ny
      do 48 i=1,nx
        ch(i,j)=8.
        cv(i,j)=8.
48     continue
41     continue
c
c  write some input values
c
      write(6,951) (cmts(i),i=1,9)
      vgp=wlin/f1oa*.0285*sgi/press
      vgp2=wlin2/f1oa2*.0285*sgi/press
      write(6,952) vgp,vgp2,sgi,wlin,wlin2
951 format('1',10x,9a8,/)
952 format(' Inlet gas velocity (I/O) (m/sec)      ',.2f18.2,/,
1      ' Inlet temp (K)                          ',.f18.1,/,
2      ' Inlet flow (I/O) (kg/sec)                ',.2f18.3,/)

```

69

```

      t(i,j)=t(i,j)+delt/wcp(i,j)*(ch(i,j)*(t(i-1,j)-t(i,j))
1      +ch(i+1,j)*(t(i+1,j)-t(i,j))
2      +cv(i,j)*(t(i,j-1)-t(i,j))
3      +cv(i,j+1)*(t(i,j+1)-t(i,j)))
      if(t(i,j).gt.1288..or.t(i,j).lt.8.) go to 988
139  continue
      t(2,j+1)=t(2,j)
      t(nx-1,j+1)=t(nx-1,j)
c
c  solid temps-copper/inconel turn
      m=j+1
      do 148 i=2,nx-1
        t(i,m)=t(i,m)+delt/wcp(i,m)*(ch(i,m)*(t(i-1,m)-t(i,m))
1        +ch(i+1,m)*(t(i+1,m)-t(i,m))
2        +cv(i,m)*(t(i,m-1)-t(i,m))
3        +cv(i,m+1)*(t(i,m+1)-t(i,m)))
148  continue
      t(2,j)=t(2,j+1)
      t(nx-1,j)=t(nx-1,j+1)
158  continue
      sumflo=sumflo+wlin*delt
      sumflo2=sumflo2+wlin2*delt
c
c  increment time; decide whether to print and/or recalc coefs
c
      time=time+delt
      if(t(5,352).le.82..and.t(5,256).le.82.) go to 1888
      if (time.lt.pt) go to 288
      call prnt
      pt=pi+time
288  continue
      if (time.ge.rntm) go to 1888
      if(ncoef.lt.1) then
c      write(6,975) t(i,j)
c975  format(/," Inlet temp = ",f18.1,"   Outlet temp = ",f18.1)
      call coefs
      ncoef=ncoef+1
      if(ncoef.eq.1) call prnt
      go to 65
      end if
      if (time.lt.coet) go to 218
      call coefs
218  continue
      go to 65
988  write(6,998) time,t(i,j),i,j
998  format(/," At time=",f18.4," a temp of",f7.1," occurred at"
1    , " node",216)
1888  continue
      call prnt
1881  stop
      end
c
c  subroutine to calc ht coefs
c
      subroutine coefs
c
      dimension t(18,488),ch(18,488),cv(18,488),gh(18,488),gv(18,488),
1    wt(18,488),wcp(18,488),cmts(9),alen(488)
c
      common t,ch,cv,gh,gv,wt,wcp,alen

```



```

common rnta,delt,time,nx,ny
common ff,tgi,press,whx,dtrhx
common dpfs,dpas,dpes,deltap
common wlin,dh,floa,to,btus,sumflo
common wlin2,dh2,floa2,to2,btus2,sumflo2
c
c Modify inlet temp based on outlet temp
c
    trhx=(to*wlin+to2*wlin2)/(wlin+wlin2)+dtrhx
    call deltatem(trhx,whx,tgi)
c
c Copper strip horizontal conductance
c
    do 51 j=2,ny-1,2
        do 50 i=3,nx-1
            ta=(t(i-1,j)+t(i,j))/2.
            ch(i,j)=fkc(ta)*gh(i,j)
50    continue
51    continue
c
c Copper/Inconel region horizontal conductance
c
    do 53 j=3,ny-1,2
        do 52 i=3,nx-1
            ta=(t(i-1,j)+t(i,j))/2.
            ch(i,j)=fkg(ta)*gh(i,j)
52    continue
53    continue
c
c Vertical conductance
c
    do 61 j=3,ny-1
        do 60 i=3,nx-2
            ta=(t(i,j-1)+t(i,j))/2.
            cv(i,j)=fkg(ta)*gv(i,j)
60    continue
61    continue
c
c calc wcp at each node
c
    do 71 j=2,ny-1,2
        do 70 i=2,nx-1
            wcp(i,j)=wt(i,j)*fcpc(t(i,j))
            wcp(i,j+1)=wt(i,j+1)*fcpci(t(i,j+1))
70    continue
71    continue
c
c find largest k/wcp at any node to adjust delt for stability
c
    bigsum=0.
    do 81 j=2,ny-1
        do 80 i=2,nx-1
            sum=(ch(i,j)+ch(i+1,j)+cv(i,j)+cv(i,j+1))/wcp(i,j)
            if(sum.lt.bigsum) go to 80
            bigsum=sum
80    continue
81    continue
84    if(delt*bigsum.lt.0.5) go to 85
        delt=delt/2.
        go to 84
85    continue

```

```

        return
    end
    subroutine prnt
c
c  subroutine to print results
c
    dimension chp(400),hdg(17)
    dimension t(10,400),ch(10,400),cv(10,400),gh(10,400),gv(10,400),
    1 wt(10,400),wcp(10,400),cmts(9),alen(400)
c
    common t,ch,cv,gh,gv,wt,wcp,alen
    common rntm,delt,time,nx,ny
    common ff,tgi,press,whx,dtrhx
    common dpfs,dpms,dpes,deltap
    common wlin,dh,floa,to,btus,sumflo
    common wlin2,dh2,floa2,to2,btus2,sumflo2
c
    data hdg/' 1"," 2"," 3"," 4"," 5"," 6"," 7"," 8"," 9",
    1 " 10"," 11"," 12"," 13"," 14"," 15"/
    data blk/' "/,hdr7/"outer"/
    data hdr3/' h"/,hdr4/"w/m2"/,hdr5/"(Gas)"/,hdr6/"inner"/
c
    do 12 j=2,ny-1
        chp(j)=ch(2,j)/gh(2,j)
    12 continue
    pjs=btus/1.e6
    pjs2=btus2/1.e6
    call delp(1)
c
c  write top info
c
    write(6,1000) time,delt,to,to2,pjs,pjs2,wlin,wlin2,sumflo,
    1 sumflo2,dpfs,dpms,dpes,deltap
    1000 format(///," Time (sec) ".f10.5,/,
    1 " Delta time (sec) ".f10.5,/,
    2 " Outlet temp (I/O) (K) ".2f10.3,/,
    3 " Heat removed (I/O)(MJ) ".2f10.1,/,
    5 " Flow rate (I/O) (kg/sec) ".2f10.3,/,
    6 " Integ flow (I/O) (kg) ".2f10.3,/,
    8 " Press drop (I) (atm)(f,m,e,t) ".4f10.4)
    call delp(2)
    write(6,1005) dpfs,dpms,dpes,deltap
    1005 format(" Press drop (O) (atm)(f,m,e,t) ".4f10.4,/)
c
c  write header
c
    write(6,1010)
    1010 format(40x,"Temperature (K)"./,45x,"1 node",/)
    write(6,1020) (blk,i=1,nx),hdr3,hdr3
    write(6,1020) (hdg(i),i=1,nx),hdr4,hdr4
    write(6,1025) (blk,i=1,nx-2),hdr5,hdr6,hdr7
    1020 format(11x,17a7)
    1025 format(" j node (Gas)".2x,15a7)
    do 21 j=ny-1,3,-2
        write(6,1030) j,(t(i,j),i=1,nx),chp(j),ch(nx,j)/gh(nx,j)
        write(6,1031) j-1,(t(i,j-1),i=2,nx-1)
    21 continue
    1030 format(3x,12,4x,17f7.1)
    1031 format(3x,12,11x,15f7.1)
    return
    end

```

```

c
c Calculates total pressure drop from friction,momentum,and elevation
c
c      subroutine delp(k)
c
c      dimension sv(400)
c      dimension t(10,400),ch(10,400),cv(10,400),gh(10,400),gv(10,400),
c      1 wt(10,400),wcp(10,400),cmts(9),alen(400)
c
c      common t,ch,cv,gh,gv,wt,wcp,alen
c      common rntm,delt,time,nx,ny
c      common ff,tgi,press,whx,dtrhx
c      common dpfs,dpms,dpes,deltap
c      common wlin,dh,floa,to,btus,sumflo
c      common wlin2,dh2,floa2,to2,btus2,sumflo2
c      data gc/9.75/
c
c      k=1 for inner flow channel; k=2 for outer flow channel
c
c      calc specific volume
c
c      if(k.lt.2) then
c        g=wlin/floa
c        t(1,ny)=t(1,ny-2)
c        do 10 j=2,ny,2
c          sv(j)=.0285*t(1,j)/press
c10      continue
c        dhx=dh
c      else
c        g=wlin2/floa2
c        t(nx,ny)=t(nx,ny-2)
c        do 15 j=2,ny,2
c          sv(j)=.0285*t(nx,j)/press
c15      continue
c        dhx=dh2
c      end if
c      gsq=g**2
c      dpfs=#.
c      dpms=#.
c      dpes=#.
c
c      calc pressure drop: f=friction, m=momentum, e=elevation
c      do 20 j=2,ny-1,2
c        dpf=(ff*gsq*(sv(j)+sv(j+2)))/2./dhx/2./gc)*alen(j)/10353.
c        dpm=(gsq*(sv(j+2)-sv(j)))/alen(j)/gc)*alen(j)/10353.
c        dpe=(2./(sv(j)+sv(j+2)))*alen(j)/10353.
c        dpfs=dpfs+dpf
c        dpms=dpms+dpm
c        dpes=dpes+dpe
c20      continue
c      deltap=dpfs+dpms+dpes
c      return
c      end
c
c
c      function to calc heat xtf correlation for Helium (metric)
c      (Cp^.4*k^.6/mu^.4)
c
c      function fcpkxn(tin)
c
c      fcpkxn=616.+422*tin

```

```

        return
    end
c
c function to calc specific heat of Helium (j/kg-k)
c
c
c     function fcpn(tin)
c
c         fcpn=525.
c         return
c     end
c
c function to calc conductivity of copper (w/m-k)
c
c     function fkc(tin)
c
c         dimension t(5),c(5)
c         data t/5.83.,111.,15.25./
c         data c/1254.,778.5,58.476.,389./
c
c         if(tin.lt.t(1)) then
c             fkc=c(1)
c             return
c         else if(tin.ge.t(5)) then
c             fkc=c(5)
c             return
c
c         end if
c         do 10 i=2,5
c             if(tin.gt.t(i)) go to 10
c             fkc=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)
c             go to 10
c         10 continue
c     100 return
c     end
c
c function to cal conductivity of G-10 (w/m-k)
c
c     function fkg(tin)
c         fkg=.8274*tin*.54
c         return
c     end
c
c function to cal specific heat of copper (j/kg-k)
c approximates the curve with straight line segments whose end
c points are given by the t and c matrices
c     function fcpc(tin)
c
c         dimension t(6),c(6)
c         data t/5.83.,100.,15.225.,300.,500./
c         data c/113.,251.,322.,368.,391.,409./
c
c         if(tin.lt.t(1)) then
c             fcpc=c(1)
c             return
c         else if(tin.ge.t(6)) then
c             fcpc=c(6)
c             return
c
c         end if
c         do 10 i=2,6
c             if(tin.gt.t(i)) go to 10
c             fcpc=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)

```

```

      go to 100
10  continue
100 return
end

c
c function to cal specific heat of Inconel (j/kg-k)
c approximates the curve with straight line segments whose end
c points are given by the t and c matrices
c
      function fcpl(tin)
c
      dimension t(6),c(6)
      data t/50.,100.,150.,225.,300.,500./
      data c/54.,226.,334.,409.,543.,468./
c
      if(tin.lt.t(1)) then
          fcpl=c(1)
          return
      else if(tin.ge.t(6)) then
          fcpl=c(6)
          return
      end if
      do 10 i=2,6
          if(tin.gt.t(i)) go to 10
          fcpl=(tin-t(i-1))*(c(i)-c(i-1))/(t(i)-t(i-1))+c(i-1)
          go to 100
10  continue
100 return
end

c
c Function to calc specific heat of mixture (j/kg-k)
c
      function fcpci(ti)
      common /cpcal/vfcu,vfin,vfci
      fcpci=(vfcu*fcpci(ti)+vfin*fcpl(ti))/vfci
      return
      end

c
c subroutine to find CIT inlet temp. given outlet temp and
c total CIT helium flow rate. Data is based on other calcs
      subroutine deltatem(tin,flow,tout)
      dimension tbrk(5),c(6,5)
      data tbrk/175.,200.,250.,300.,350./
      data c/
1      77.2 ,77.2 ,77.2 ,77.2 ,77.2 , 77.2 .
2      77.2 ,77.2 ,77.5 ,81. , 85. , 94. .
3      77.2 ,77.5 ,86. ,104. ,140. ,147. .
4      77.2 ,85. ,109. ,147. ,162. ,175. .
c
      if(tin.le.175.) then
          tout=77.2
          return
      end if
      n=nint(flow)/2-1
      do 10 i=2,6
          if(tin.gt.tbrk(i)) go to 10
          tout=(tin-tbrk(i-1))*(c(n,i)-c(n,i-1))/(tbrk(i)-tbrk(i-1))+
              c(n,i-1)
          go to 100
10  continue
      tout=c(n,5)
100 return
      end

```

Baseline, flow =2.4 kg/s
 5000. 5. 600. 100. 9 354
 2. .0144 .0115 .01875 80
 4. .0144 .023 .01542 48
 .031 92. 8. 14. 10.
 .55 .45

0.000	0.020	0.022	0.088	0.090	0.102	0.115	0.190	0.037
0.000	0.018	0.101	0.077	0.090	0.102	0.115	0.190	0.037
0.000	0.018	15.022	17.941	20.859	23.778	26.697	29.615	0.037
0.000	0.018	0.101	0.077	0.090	0.102	0.115	0.190	0.037
0.000	0.018	15.022	17.941	20.859	23.778	26.697	29.615	0.037

Repeat the two-line sequence 45 times

0.000	0.018	0.101	0.077	0.090	0.102	0.115	0.190	0.037
0.000	0.018	15.022	17.941	20.859	23.778	26.697	29.615	0.037
0.000	0.022	0.097	0.077	0.090	0.102	0.115	0.182	0.045
0.000	0.022	19.050	22.751	26.452	30.153	33.854	37.555	0.045
0.000	0.022	0.097	0.077	0.090	0.102	0.115	0.182	0.045
0.000	0.022	19.050	22.751	26.452	30.153	33.854	37.555	0.045

Repeat the two-line sequence 77 times

0.000	0.022	0.097	0.077	0.090	0.102	0.115	0.182	0.045
0.000	0.022	19.050	22.751	26.452	30.153	33.854	37.555	0.045
0.000	0.018	0.101	0.077	0.090	0.102	0.115	0.190	0.037
0.000	0.018	15.022	17.941	20.859	23.778	26.697	29.615	0.037
0.000	0.018	0.101	0.077	0.090	0.102	0.115	0.190	0.037
0.000	0.018	15.022	17.941	20.859	23.778	26.697	29.615	0.037

Repeat the two-line sequence 45 times

0.000	0.018	0.101	0.077	0.090	0.102	0.115	0.190	0.037
0.000	0.018	15.022	17.941	20.859	23.778	26.697	29.615	0.037
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0
0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0

Repeat 91 times

0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0
0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	160.7	189.1	217.6	246.0	274.5	0.0	0.0
0.0	0.0	160.7	189.1	217.6	246.0	274.5	0.0	0.0

Repeat 156 times

0.0	0.0	160.7	189.1	217.6	246.0	274.6	0.0	0.0
0.0	0.0	160.7	189.1	217.6	246.0	274.5	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0
0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0

Repeat 91 times

0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0
0.0	0.0	170.5	200.7	230.9	261.1	291.3	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	0.400	1.999	2.353	2.707	3.061	3.415	0.799	1.000
1.000	0.400	11.609	13.665	15.721	17.777	19.832	0.799	1.000
1.000	0.400	1.999	2.353	2.707	3.061	3.415	0.799	1.000
1.000	0.400	11.609	13.665	15.721	17.777	19.832	0.799	1.000

Repeat two-line sequence 45 times

1.000	0.400	1.999	2.353	2.707	3.061	3.415	0.799	1.000
1.000	0.400	11.609	13.665	15.721	17.777	19.832	0.799	1.000
1.000	0.507	1.999	2.353	2.707	3.061	3.415	1.013	1.000
1.000	0.507	14.714	17.320	19.926	22.531	25.137	1.013	1.000
1.000	0.507	1.999	2.353	2.707	3.061	3.415	1.013	1.000
1.000	0.507	14.714	17.320	19.926	22.531	25.137	1.013	1.000

Repeat two-line sequence 77 times

1.000	0.507	1.999	2.353	2.707	3.061	3.415	1.013	1.000
1.000	0.507	14.714	17.320	19.926	22.531	25.137	1.013	1.000
1.000	0.400	1.999	2.353	2.707	3.061	3.415	0.799	1.000
1.000	0.400	11.609	13.665	15.721	17.777	19.832	0.799	1.000
1.000	0.400	1.999	2.353	2.707	3.061	3.415	0.799	1.000
1.000	0.400	11.609	13.665	15.721	17.777	19.832	0.799	1.000

Repeat two-line sequence 45 times

1.000	0.400	1.999	2.353	2.707	3.061	3.415	0.799	1.000
1.000	0.400	11.609	13.665	15.721	17.777	19.832	0.799	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
167.0	167.0	167.0	167.0	167.0	167.0	167.0	167.0	167.0
167.0	167.0	167.0	167.0	167.0	167.0	167.0	167.0	167.0

Repeat 93 times

167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0
304.0 304.0 304.0 304.0 304.0 304.0 304.0 304.0 304.0
304.0 304.0 304.0 304.0 304.0 304.0 304.0 304.0 304.0

Repeat 157 times

304.0 304.0 304.0 304.0 304.0 304.0 304.0 304.0 304.0
167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0
167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0

Repeat 93 times

167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0 167.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

t


```

10 ' CODE TO CALC THE GH,GV,WT,AND T MATRICES FOR CITTOH (OHDATA)
11 ' SEE END FOR EXPLANATION OF INPUT
20 DIM GH(10),GH1(10),GV(10),WT(10),WT1(10),T(10),TI(2),M1(3),RIM(10)
30 DIM TCU(2),TIW(2),HM(2),TIP(2),TINC(2),NPF(2),TCUM(2),RCUN(10)
35 INPUT "OUTPUT DATA FILE NAME";OUTFIL$
36 OPEN OUTFIL$ FOR OUTPUT AS #1
40 READ NPF(1),NPF(2),NRI,R1,R2,DENCU,DENIN
50 READ TCU(1),TIW(1),HM(1),TCUM(1),TINC(1),TIP(1),FRHX(1),TI(1)
60 READ TCU(2),TIW(2),HM(2),TCUM(2),TINC(2),TIP(2),FRHX(2),TI(2)
70 READ M1(1),M1(2),M1(3)
80 NX=NRI+4 :TPI=2*3.1416 :NY=2+2*NPF(1)+4*NPF(2)
82 G1$="###.### " :W1$="###.## "
84 G$=G1$+G1$+G1$+G1$+G1$+G1$+G1$+G1$+G1$
86 W$=W1$+W1$+W1$+W1$+W1$+W1$+W1$+W1$+W1$
90 ' CALC GH
95 PRINT USING G$;0;0;0;0;0;0;0;0;0;0
96 PRINT #1,USING G$;0;0;0;0;0;0;0;0;0;0
100 FOR N=1 TO 3
105 M=M1(N)
110 GOSUB 2000
120 FOR J=2 TO 2*NPF(M)+1 STEP 2
130 GH(1)=0 :GH1(1)=0
140 GH(2)=((R1+TCU(M)/2)*TPI*(TCU(M)+TIP(M)+HM(M)-TINC(M)/2))*FRHX(M) :GH1(2)=
GH(2)
150 FOR I=3 TO NX-1
160 GH(I)=TPI/LOG(RCUN(I)/RCUN(I-1))*TCU(M)
170 GH1(I)=RIM(I)*TPI*HM(M)/TIW(M)
180 NEXT I
190 GH(NX)=((R2+TCU(M)/2)*TPI*(TCU(M)+TIP(M)+HM(M)-TINC(M)/2))*FRHX(M) :GH1(NX)
)=GH(NX)
200 PRINT USING G$;GH(1);GH(2);GH(3);GH(4);GH(5);GH(6);GH(7);GH(8);GH(9)
201 PRINT #1,USING G$;GH(1);GH(2);GH(3);GH(4);GH(5);GH(6);GH(7);GH(8);GH(9)
210 PRINT USING G$;GH1(1);GH1(2);GH1(3);GH1(4);GH1(5);GH1(6);GH1(7);GH1(8);GH1
(9)
211 PRINT #1,USING G$;GH1(1);GH1(2);GH1(3);GH1(4);GH1(5);GH1(6);GH1(7);GH1(8);
GH1(9)
220 NEXT J
230 NEXT N
235 PRINT USING G$;0;0;0;0;0;0;0;0;0;0
236 PRINT #1,USING G$;0;0;0;0;0;0;0;0;0;0
240 ' CALC GV
250 PRINT USING W$;0;0;0;0;0;0;0;0;0;0 :PRINT USING W$;0;0;0;0;0;0;0;0;0;0
251 PRINT #1,USING W$;0;0;0;0;0;0;0;0;0;0 :PRINT #1,USING W$;0;0;0;0;0;0;0;0;0;0
260 FOR N=1 TO 3
270 M=M1(N) :GOSUB 2000
280 FOR J=3 TO 2*NPF(M)+1
290 FOR I=3 TO NX-2
300 GV(I)=RCUN(I)*TPI*DELM/(TIP(M)+.05*TINC(M))
310 NEXT I
320 PRINT USING W$;GV(1);GV(2);GV(3);GV(4);GV(5);GV(6);GV(7);GV(8);GV(9)
321 PRINT #1,USING W$;GV(1);GV(2);GV(3);GV(4);GV(5);GV(6);GV(7);GV(8);GV(9)
330 NEXT J
340 PRINT USING W$;0;0;0;0;0;0;0;0;0;0
341 PRINT #1,USING W$;0;0;0;0;0;0;0;0;0;0
350 NEXT N
360 ' CALC WEIGHT
370 PRINT USING G$;1;1;1;1;1;1;1;1;1;1
371 PRINT #1,USING G$;1;1;1;1;1;1;1;1;1;1
380 FOR N=1 TO 3
390 M=M1(N) :GOSUB 2000
400 FOR J=2 TO 2*NPF(M)+1 STEP 2
410 WT(1)=1 :WT1(1)=1
420 WT(2)=HM(M)*R1*TPI*TCU(M)*DENCU :WT1(2)=WT(2)

```

```

430   FOR I=3 TO NX-2
440     WT(I)=RCUN(I)*TPI*TCU(M)*DELX*DENCU
450     WT1(I)=RCUN(I)*TPI*DELM*(TCUM(M)*DENCU+TINC(M)*2*DENIN)
460   NEXT I
470   WT(NX-1)=HM(M)*R2*TPI*TCU(M)*DENCU :WT1(NX-1)=WT(NX-1)
475   WT(NX)=1 :WT1(NX)=1
480   PRINT USING G$;WT(1);WT(2);WT(3);WT(4);WT(5);WT(6);WT(7);WT(8);WT(9)
481   PRINT #1,USING G$;WT(1);WT(2);WT(3);WT(4);WT(5);WT(6);WT(7);WT(8);WT(9)
490   PRINT USING G$;WT1(1);WT1(2);WT1(3);WT1(4);WT1(5);WT1(6);WT1(7);WT1(8);WT1
(9)
491   PRINT #1,USING G$;WT1(1);WT1(2);WT1(3);WT1(4);WT1(5);WT1(6);WT1(7);WT1(8);
WT1(9)
495   NEXT J
500   NEXT N
510   PRINT USING G$;1;1;1;1;1;1;1;1;1
511   PRINT #1,USING G$;1;1;1;1;1;1;1;1;1
520   ' INITIAL TEMPS
525   PRINT USING W$;1;1;1;1;1;1;1;1;1
526   PRINT #1,USING W$;1;1;1;1;1;1;1;1;1
530   FOR N=1 TO 3
540     M=M1(N)
545     FOR J=2 TO 2*NPF(M)+1
550       FOR I=1 TO NX
560         T(I)=TI(M)
570       NEXT I
580       PRINT USING W$;T(1);T(2);T(3);T(4);T(5);T(6);T(7);T(8);T(9)
581       PRINT #1,USING W$;T(1);T(2);T(3);T(4);T(5);T(6);T(7);T(8);T(9)
590     NEXT J
600   NEXT N
610   PRINT USING W$;1;1;1;1;1;1;1;1;1
611   PRINT #1,USING W$;1;1;1;1;1;1;1;1;1
620   GOTO 3000
2000   TMEAT=R2-R1-TCU(M)*2
2010   DELX=(TMEAT-TIW(M))/NRI
2020   DELM=DELX-TIW(M)
2030   RCUN(2)=R1-HM(M)/2
2040   RCUN(3)=R1+TCU(M)+TIW(M)+DELM/2
2050   FOR I=4 TO NX-2
2060     RCUN(I)=RCUN(I-1)+DELX
2070   NEXT I
2080   RCUN(NX-1)=R2+HM(M)/2
2090   RIM(3)=R1+TCU(M)+TIW(M)/2
2100   FOR I=4 TO NX-1
2110     RIM(I)=RIM(I-1)+DELX
2120   NEXT I
2130   RETURN
2200   'DATA 50,20,5,.228,.509,8950,.8160. : 'PAGE 55 DATA
2210   'DATA .002,.002,.0134,.0089,.00224,.0005,.5,221. : 'PAGE 55 DATA
2220   'DATA .002,.002,.0264,.0152,.0056,.0005,.5,317. : 'PAGE 55 DATA
2230   'DATA 2,1,2 : 'PAGE 55 DATA
2250   DATA 80,48,5,.286,.572,8950,8160 : '9/10 FAX
2260   DATA .002,.0015,.01575,.008695,.0035275,.0005,.75,304. : '9/10 FAX
2270   DATA .002,.0015,.01242,.006925,.0027475,.0005,.75,167. : '9/10 FAX
2280   DATA 2,1,2 : '9/10 FAX
2500   '
2501   ' THE FOUR LINES OF DATA ARE AS FOLLOWS:
2502   ' LINE 1:
2504   '
2506   ' TOTAL PANCAKES IN PF1
2508   ' PANCAKES IN EACH PF2
2510   ' NUMBER OF RADIAL CU/INCONEL NODES
2512   ' RADIUS TO INSIDE OF COIL (M)
2514   ' RADIUS TO OUTSIDE OF COIL (M)
2516   ' DENSITY OF COPPER (KG/M3)
2518   ' DENSITY OF INCONEL (KG/M3)
2520   '

```

2522 ' LINE 2 (FOR PF1):
 2524 '
 2526 ' THICKNESS OF COPPER HEAT CONDUCTOR (M)
 2528 ' THICKNESS OF G10 INSULATION BETWEEN TURNS (M)
 2530 ' TOTAL HEIGHT OF CU/INCONEL IN CONDUCTOR (M)
 2532 ' HEIGHT OF CU IN THE CONDUCTOR (M)
 2534 ' HEIGHT OF INCONEL IN THE CONDUCTOR (ONE SIDE) (M)
 2536 ' THICKNESS OF G10 BETWEEN HEAT CONDUCTOR AND TURNS (M)
 2538 ' FRACTION OF WALL SURFACE AVAILABLE FOR HEAT TRANSFER
 2540 ' INITIAL TEMPERATURE (K)
 2542 '
 2544 ' LINE 3 : SAME AS LINE 2 EXCEPT FOR PF2
 2546 '
 2548 ' LINE 4:
 2550 '
 2552 ' SEQUENCE OF DIGITS THAT IDENTIFIES BOTTOM, MIDDLE, TOP
 2554 ' AS PF2, PF1, PF2 BY INPUTTING 2,1,2
 3000 END

INTERNAL DISTRIBUTION

1. J. R. Haines	10-11. Laboratory Records Department
2. P. N. Haubenreich	12. Laboratory Records, ORNL-RC
3. D. R. Hicks	13. Document Reference Section
4. S. S. Kalsi	14. Central Research Library
5. D. W. Langenberg	15. Fusion Energy Division Library
6. T. L. Ryan	16-17. Fusion Energy Division
7. T. Shannon	Publications Office
8. J. Sheffield	18. ORNL Patent Office
9. M. L. Givens	

EXTERNAL DISTRIBUTION

19. M. A. Abdou, Boelter Hall, University of California, Los Angeles, CA 90024
20. C. A. Anderson, Westinghouse Electric Corporation, Advanced Energy Systems Division, P.O. Box 158, Madison, PA 15663
21. J. L. Anderson, CMB-3, Mail Stop 348, Los Alamos National Laboratory, P.O. Box 1633, Los Alamos, NM 87545
22. C. C. Baker, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
23. D. S. Beard, Office of Fusion Energy, Office of Energy Research, ER-531 Germantown, U.S. Department of Energy, Washington, DC 20545
24. K. L. Elack, Department E452, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
25. R. Botwin, C47-05, Grumman Aerospace Corporation, P.O. Box 31, Bethpage, NY 11714
26. W. B. Briggs, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
27. J. N. Brooks, FPP/207, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
28. S. C. Burnett, GA Technologies, Inc., P.O. Box 81608, San Diego, CA 92138
29. J. D. Callen, Department of Nuclear Engineering, University of Wisconsin, Madison, WI 53706
30. R. N. Cherdack, Burns and Roe, Inc., 800 Kinderkamack Road, Oradell, NJ 07649
31. J. F. Clarke, Director for Fusion Energy, Office of Energy Research, ER-50 Germantown, U.S. Department of Energy, Washington, DC 20545

32. D. R. Cohn, 167 Albany Street, NW 16-140, Massachusetts Institute of Technology, Cambridge, MA 02139
33. R. W. Conn, Department of Chemical, Nuclear, and Thermal Engineering, Boelter Hall, University of California, Los Angeles, CA 90024
34. J. G. Crocker, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401
35. A. E. Dabiri, Science Applications International Corp., 11526 Sorento Valley Road, Suite A, San Diego, CA 92121
36. G. R. Dalton, Department of Nuclear Engineering Science, Nuclear Science Center, University of Florida, Gainesville, FL 32611
37. R. C. Davidson, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139
38. N. A. Davies, Office of the Associate Director, Office of Fusion Energy, Office of Energy Research, ER-51 Germantown, U.S. Department of Energy, Washington, DC 20545
39. S. O. Dean, Director, Fusion Power Associates, Inc., 2 Professional Drive, Suite 249, Gaithersburg, MD 20760
40. D. DeFreece, E451, Building 81/1/B7, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
41. J. N. Doggett, L-441, Lawrence Livermore National Laboratory, P.O. Box 5511, Livermore, CA 94550
42. H. Dreicer, Division Leader, CRT, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
43. F. Farfaletti-Casali, Engineering Division, Joint Research Center, Ispra Establishment, 21020 Ispra (Varese), Italy
44. P. A. Finn, Fusion Power Program, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
45. H. K. Forsen, Bechtel Group, Inc., Research & Engineering, P.O. Box 3965, San Francisco, CA 94119
46. J. S. Foster, Jr., Building R4-2004, TRW Defense and Space Systems, 1 Space Park, Redondo Beach, CA 90278
47. T. K. Fowler, Associate Director for Magnetic Fusion Energy, L-436, Lawrence Livermore National Laboratory, P.O. Box 5511, Livermore, CA 94550
48. J. W. French, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
49. H. P. Furth, Director, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
50. J. G. Gavin, Jr., President, A01-11, Grumman Aerospace Corporation, P.O. Box 31, Bethpage, NY 11714
51. G. Gibson, Westinghouse Electric Corporation, Waltz Mill Site, P.O. Box 158, Madison, PA 15663
52. J. R. Gilleland, GA Technologies, Inc., Fusion and Advanced Technology, P.O. Box 81608, San Diego, CA 92138

53. M. Y. Gohar, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
54. R. W. Gould, Department of Applied Physics, California Institute of Technology, Pasadena, CA 91125
55. R. A. Gross, Plasma Research Laboratory, Columbia University, New York, NY 10027
56. J. R. Haines, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
57. C. D. Henning, Lawrence Livermore National Laboratory, P.O. Box 5511, Livermore, CA 94550
58. J. J. Holmes, Westinghouse-Hanford Engineering Development Laboratory, P.O. Box 1970, Richland, WA 99352
59. J. B. Joyce, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
- 60-64. D. C. Keeton, 464 Robertsville Road, Oak Ridge, TN 37830
65. R. A. Krakowski, CTR-12, Mail Stop 641, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
66. G. L. Kulcinski, University of Wisconsin, Department of Nuclear Engineering, Engineering Research Building, Room 439, 1500 Johnson Drive, Madison, WI 53706
67. D. L. Kummer, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
68. W. Marton, Office of Fusion Energy, Office of Energy Research, ER-55 Germantown, U.S. Department of Energy, Washington, DC 20545
69. L. G. Masson, EG&G Idaho, Idaho National Engineering Laboratory, P.O. Box 1625, Idaho Falls, ID 83401
70. D. M. Meade, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
71. A. T. Mense, Building 107, Post B2, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
72. R. W. Moir, Lawrence Livermore National Laboratory, P.O. Box 5511, Livermore, CA 94550
73. D. B. Montgomery, MIT Plasma Fusion Center, 167 Albany Street, Cambridge, MA 02139
74. A. E. Munier, Grumman Aerospace Corporation, P.O. Box 31, Bethpage, NY 11714
75. R. E. Nygren, FPP/207, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
76. T. Ohkawa, GA Technologies, Inc., P.O. Box 81608, San Diego, CA 92138
77. J. A. O'Toole, Grumman Corporation, B29-25, Bethpage, NY 11714
78. R. R. Parker, Francis Bitter National Magnet Laboratory, 170 Albany Street, Cambridge, MA 02139

79. B. Pease, Culham Laboratory, Abingdon, Oxfordshire OX14 3DB, United Kingdom
80. M. Pelovitz, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
81. F. W. Perkins, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
82. R. Pillsbury, Plasma Fusion Center, Massachusetts Institute of Technology, 175 Albany Street (NW17-219), Cambridge, MA 02139
83. M. Porkolab, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139
84. D. E. Post, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
85. R. E. Price, Office of Fusion Energy, Office of Energy Research, ER-55 Germantown, U.S. Department of Energy, Washington, DC 20545
86. F. A. Puhn, GA Technologies, Inc., P.O. Box 81608, San Diego, CA 92138
87. R. V. Pyle, University of California, Lawrence Berkeley Laboratory, Berkeley, CA 94720
88. J. M. Rawls, GA Technologies, Inc., P.O. Box 81608, San Diego, CA 92138
89. M. Roberts, International Program, Office of Fusion Energy, Office of Energy Research, ER-52 Germantown, U.S. Department of Energy, Washington, DC 20545
90. G. Robinson, Plasma Fusion Center, Massachusetts Institute of Technology, 175 Albany Street (NW17-219), Cambridge, MA 02139
91. J. D. Rogers, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
92. M. L. Rogers, Monsanto Research Corporation, Mound Laboratory Facility, P.O. Box 32, Miamisburg, OH 45342
93. M. N. Rosenbluth, RLM 11.218, Institute for Fusion Studies, University of Texas, Austin, TX 78712
94. P. H. Rutherford, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
95. J. A. Schmidt, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
96. J. Schultz, MIT Plasma Fusion Center, 167 Albany Street, Cambridge, MA 02139
97. F. R. Scott, Electric Power Research Institute, P.O. Box 10412, Palo Alto, CA 94304
98. G. Sheffield, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
99. D. Smith, Materials Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
100. W. M. Stacey, Jr., Georgia Institute of Technology, School of Nuclear Engineering and Health Physics, Atlanta, GA 30332

101. D. Steiner, Rensselaer Polytechnic Institute, Nuclear Engineering Department, NES Building, Tibbets Avenue, Troy, NY 12181
102. E. Stern, Grumman Aerospace Corporation, CN-59, Forrestal Campus, Princeton, NJ 08544
103. P. M. Stone, Fusion Systems Design Branch, Division of Development and Technology, Office of Fusion Energy, Office of Energy Research, ER-532 Germantown, U.S. Department of Energy, Washington, DC 20545
104. I. N. Sviatoslavsky, Room 33, Engineering Research Building, 1500 Johnson Drive, University of Wisconsin, Madison, WI 53706
105. R. E. Tatro, Manager, Energy Systems, M.Z. 16-1070, General Dynamics-Convair Division, P.O. Box 80847, San Diego, CA 92138
106. F. Thomas, B-20-5, Grumman Aerospace Corporation, Bethpage, NY 11714
107. K. I. Thomassen, Lawrence Livermore National Laboratory, P.O. Box 5511, Livermore, CA 94550
108. R. J. Thome, Francis Bitter National Magnet Laboratory, 170 Albany Street, Cambridge, MA 02139
109. C. Trachsel, McDonnell Douglas Astronautics Company, P.O. Box 516, St. Louis, MO 63166
110. A. W. Trivelpiece, Office of Energy Research, U.S. Department of Energy, Washington, DC 20545
111. L. R. Turner, Fusion Power Program, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
112. E. H. Valeo, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
113. R. Varma, Physical Research Laboratory, Navrangpura, Ahmedabad 380009, India
114. J. C. Wesley, GA Technologies, Inc., P.O. Box 81608, San Diego, CA 92138
115. J. E. C. Williams, Francis Bitter National Magnet Laboratory, 170 Albany Street, Cambridge, MA 02139
116. H. H. Yoshikawa, W/A-62, Hanford Engineering Development Laboratory, P.O. Box 1970, Richland, WA 99352
117. K. M. Young, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
118. N. E. Young, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08544
119. Bibliothek, Max-Planck Institut fur Plasmaphysik, D-8046 Garching, Federal Republic of Germany
120. Bibliothek, Institut fur Plasmaphysik, KFA, Postfach 1913, D-5170 Julich, Federal Republic of Germany
121. F. Prevot, CEN/Cadarache, Departement de Recherches sur la Fusion Controlee, 13108 Saint-Paul-lez-Durance, France

122. Documentation S.I.G.N., Departement de la Physique du Plasma et de la Fusion Controlee, Association EURATOM-CEA, Centre d'Etudes Nucleaires, B.P. 85, Centre du Tri, 38041 Grenoble Cedex, France
123. Library, Centre de Recherches en Physique des Plasmas, 21 Avenue des Bains, 1007 Lausanne, Switzerland
124. Library, Culham Laboratory, UKAEA, Abingdon, Oxfordshire, OX14 3DB, England
125. Library, FOM Instituut voor Plasmafysica, Rijnhuizen, Edisonbaan 14, 3439 MN Nieuwegein, The Netherlands
126. J. Shang-Xian, Graduate School, Academia Sinica, P.O. Box 3908, Beijing, China (PRC)
127. Library, Institute for Plasma Physics, Nagoya University, Nagoya 464, Japan
128. Library, International Centre for Theoretical Physics, Trieste, Italy
129. Library, JET Joint Undertaking, Abingdon, Oxfordshire, OX14 3EA, England
130. Library, Laboratorio Gas Ionizzati, CP 65, 00044 Frascati, Rome, Italy
131. Plasma Research Laboratory, Australian National Laboratory, P.O. Box 4, Canberra, ACT 2000, Australia
132. Thermonuclear Library, Japan Atomic Energy Research Institute, Tokai Research Establishment, Tokai, Ibaraki 319-11, Japan
133. Library, Plasma Physics Laboratory, Kyoto University, Gokasho, Uji, Kyoto 611, Japan
134. Office of the Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge Operations, P.O. Box E, Oak Ridge, TN 37831
- 135-275. Given distribution as shown in TIC-4500, Magnetic Fusion Energy (Distribution Category UC-20c,d: Reactor Materials and Fusion Systems)