SSCL-N-852

Analysis of Crossflow Cooling in 50-mm CDM with Orificed Channels

A. Yücel

Superconducting Super Collider Laboratory*
2550 Beckleymeade Ave.
Dallas, TX 75237 USA

April 1994

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

1.0 INTRODUCTION

The crossflow cooling scheme proposed by General Dynamics[1] employs yoke laminations with smaller holes (diameter < 0.03175 m) at the lead and return ends of the collider dipole magnet (CDM). One restricted yoke lamination is placed at the inlet for the upper two channels and another is placed at the exit for the lower two channels. The smaller holes—orifices—limit the flows entering the upper channels and those exiting the lower channels, thus creating a pressure differential between the lower and upper channels. This pressure differential can divert flow from the lower half to the upper half via slotted yokes used periodically along the length of the magnet. The infusion of colder coolant from the lower channels into the annular channel and the discharge of warmer coolant out of the annular channel into the upper channels provide the crossflow cooling effect.

This note presents a parametric analysis of the above crossflow cooling scheme. There are two important parameters in assessing the effectiveness of crossflow cooling for a given design and operating conditions. These parameters are: (1) the orifice loss coefficient, which determines the pressure drop across the orifice plates and adds to the total pressure drop for the magnet; and (2) the crossflow conductance, which determines the transverse flow rate between the coolant channels and the annular channel.

2.0 ANALYSIS

Figure 1 shows the incompressible flow network used in the analysis. The 15-m-long dipole magnet is divided into 25 sections. The slotted yokes are assumed to be placed every 60 cm, accommodating transverse flow from the lower channels into the annular channel and from the annular channel into the upper channels. Since gaps between collar packs occur every 25 cm, there will be more than one crossflow path between the collar tooling slots and the annular channel for a given section. For simplicity, these branches are lumped together in the analysis. Hence, for each section, there is assumed to be a single but tortuous crossflow passage linking the lower two channels to the annular channel, and another one linking the annular channel to the upper channels. The conductance of the crossflow passages, Cx, is taken as a free parameter and varied between 1.0 E-6 m^2 and 5.0 E-4 m^2. This wide range of crossflow conductance values covers the whole spectrum of crossflow regimes that will be discussed later. The geometry of the crossflow passages in a given magnet design will determine the actual crossflow conductance value for the passages. The results presented here for a wide range of crossflow conductance values can be used to determine the relative crossflow efficiency for a particular design and to determine the sensitivity of the predictions to the uncertainties in estimating the crossflow conductance for the given design.

The orifice loss coefficient, Ko, is varied primarily between 10 and 40 to represent different orifices; larger Ko values reaching up to fully blocked channels are also considered. The orifice size or diameter corresponding to a certain Ko value depends on the actual shape (contour) of the orifice and can be determined from experimental data.[2] Although the loss coefficient for inlet to a pipe through an orifice is somewhat lower than that for exit from a pipe through an orifice, the inlet and exit loss coefficients in this analysis are taken to be the same.

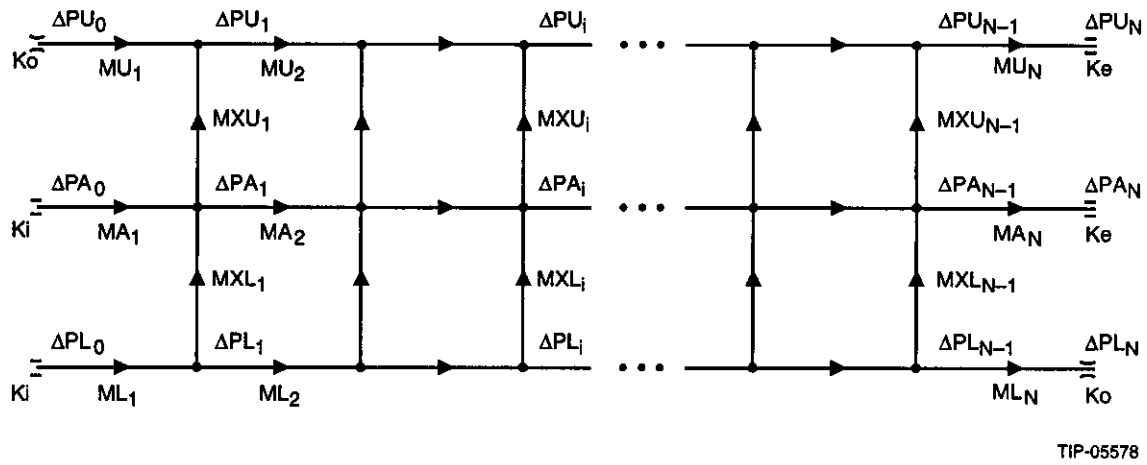

Figure 1. Flow Network.

Unless otherwise noted, the values used for the remaining baseline parameters are shown in Table 1.

Table 1. Baseline Parameters.

Inlet and exit loss coefficients for a non-orificed channel	Ki = Ke = 1
Friction factor, all channels	f = 0.02
Cooling channel diameter	D = 0.03175 m
Beam tube diameter	D_{bt} = 0.037 m
Helium density at 4.2 K, 0.4 MPa	ρ = 138 kg/m^3
Helium specific heat at 4.2 K, 0.4 MPa	Cp = 3.73E3 J/kg-K
Total mass flow rate	MT = 0.1 kg/s
Effective heat transfer conductance for the cold mass	U = 1.75 W/m-K
Linear heat input	Q = 1 W/m

The linear heat input of 1 W/m used in the calculations is a nominal rate. The annular channel temperature rise values presented in this study can be regarded as being per W/m of linear heating and can be scaled to obtain the temperature rise for other linear heat rates.

2.1 Flow Analysis

The equations governing pressure drops along the channels are

$$\Delta PL_i = \Delta PL_{i-1} + KL_i\, ML_i\, |ML_i|$$

$$\Delta PU_i = \Delta PU_{i-1} + KU_i\, MU_i\, |MU_i|$$

$$\Delta PA_i = \Delta PA_{i-1} + KA_i\, MA_i\, |MA_i|$$

$$= \Delta PA_{i-1} + KA_i\, (MT - ML_i - MU_i)\, |MT - ML_i - MU_i|,$$

where

$$KL_i = (Ki\ \delta_{i1} + f\ \Delta x/D + Ko\ \delta_{iN})/(2\rho A^2)$$

$$KU_i = (Ko\ \delta_{i1} + f\ \Delta x/D + Ke\ \delta_{iN})/(2\rho A^2)$$

$$KA_i = (Ki\ \delta_{i1} + f\ \Delta x/D_a + Ke\ \delta_{iN})/(2\rho A_a^2)$$

for i = 1, 2,..., N. The mass flow rates are governed by the mass balance equations for the mixing nodes:

$$ML_{i+1} = ML_i - MXL_i = ML_i - Cx\ \mathrm{sign}(\Delta PA_i - \Delta PL_i)\ [\rho\ |\Delta PA_i - \Delta PL_i|\ ]^{1/2}$$

$$MU_{i+1} = MU_i + MXU_i = MU_i - Cx\ \mathrm{sign}(\Delta PU_i - \Delta PA_i)\ [\rho\ |\Delta PU_i - \Delta PA_i|\ ]^{1/2}$$

for i = 1, 2,..., N–1. The boundary conditions on the system are:

$$\Delta PL_0 = \Delta PU_0 = \Delta PA_0 = 0$$

$$\Delta PL_N = \Delta PU_N = \Delta PA_N.$$

A nonlinear equation system solver is used to obtain the solutions for the pressure drops at each node and the mass flow rates along each section.

2.2 Temperature Analysis

The differential equations governing the helium temperature along a given section in the cooling channels and in the annular channel are given by

$$ML_i\ Cp\ d(TL_i)/dx = 1/2\ U\ (TA_i - TL_i)$$

$$MU_i\ Cp\ d(TU_i)/dx = 1/2\ U\ (TA_i - TU_i)$$

$$MA_i\ Cp\ d(TA_i)/dx = Q - 1/2\ U\ (TA_i - TL_i) - 1/2\ U\ (TA_i - TU_i)$$

for $x_{i-1} \le x \le x_i$, i = 1, 2,..., N. Here, U represents the total heat transfer conductance for the cold mass between the annular channel and the cooling channels.[3,4] The static heat load on the cold mass and the back diffusion of heat from the upper half of the cold mass to the lower half are neglected. The initial conditions for each section are determined assuming perfect mixing at the nodes:

$$ML_i\ TL_i(x_i) = (ML_{i+1} + \langle MXL_i,0\rangle)\ TL_{i+1}(x_i) - \langle -MXL_i,0\rangle\ TA_{i+1}(x_i)$$

$$MU_i\ TU_i(x_i) = (MU_{i+1} + \langle -MXU_i,0\rangle)\ TU_{i+1}(x_i) - \langle MXU_i,0\rangle\ TA_{i+1}(x_i)$$

$$MA_i\ TA_i(x_i) = (MA_{i+1} + \langle -MXL_i,0\rangle + \langle MXU_i,0\rangle)\ TA_{i+1}(x_i)$$
$$- \langle MXL_i,0\rangle\ TL_{i+1}(x_i) - \langle -MXU_i,0\rangle\ TU_{i+1}(x_i)$$

for i = 1, 2,..., N–1. In the above equations, the brackets are used to denote the greater of the two variables inside the brackets, i.e., $\langle a,b\rangle = \max(a,b)$. The initial conditions for the first section are

$$TL_1(0) = TU_1(0) = TA_1(0) = T_0.$$

Once the mass flow rates are determined, the above system of ordinary differential equations is solved using a suitable ODE integrator to obtain the temperature distributions in the channels.

3.0 RESULTS

3.1 Axial Variations of Pressure Drop, Mass Flow Rate, and Annular Channel Temperature

In order to illustrate the effects of crossflow, we present in this section the pressure drops, mass flow rates, and annular temperature profiles for an orificed magnet with Ko = 20.

Figure 2 shows the upper, lower, and annular channel pressure drops along a dipole magnet; Figure 3 shows the corresponding mass flow rates. The upper channel, which is orificed at the inlet, has a large pressure drop across the orifice, while the lower channel experiences a large pressure drop across its orifice at the exit. The pressure and mass flow rate profiles are strongly influenced by the crossflow conductance between the channels:

- For low crossflow conductance values, there is little communication (i.e., crossflow) between the three channels. The channels serve merely as independent flow paths, with relatively small variations in the mass flow rates along the length of the magnet. Hence the pressure gradients remain fairly constant along the channels.
- For high crossflow conductance values, there is significant crossflow near the inlet from the annular channel to the upper channel due to the large pressure differential caused by the orifice. There is little crossflow, however, from the lower channel to the annular channel. As a result, the annular channel mass flow rate decreases near the inlet while the upper channel mass flow rate increases rapidly. The situation is reversed at the exit end of the magnet: crossflow is now driven by the large pressure difference between the lower and annular channels. The annular channel mass flow rate increases, while the lower channel flow rate decreases near the exit. Hence, for high conductance values, the lower channel pressure gradient decreases along the length of the magnet as its mass flow rate increases monotonically, while the converse is true for the upper channel. This leads toward equalization of channel pressures in the mid-section of the magnet, which in turn suppresses crossflows in this region.

The axial variations of crossflows from the lower channel to the annular channel and from the annular channel to the upper channel are shown in Figure 4 for three conductance values. For low conductance values, the upper and lower crossflow rates are low and comparable to each other. At intermediate Cx values, the crossflow rates rise but become skewed toward the ends. At higher conductance values, crossflow rates are quite large at the respective ends, but vanish in the center of the magnet. Figure 5 shows the corresponding annular mass flow rates. For low conductance values, the annular flow rate is higher than that for a non-orificed magnet (without crossflow), but this is primarily driven by the large pressure drop across the magnet caused by the orificed channels. As the crossflow conductance increases, the annular mass flow rate increases near the ends but decreases in the center of the magnet.

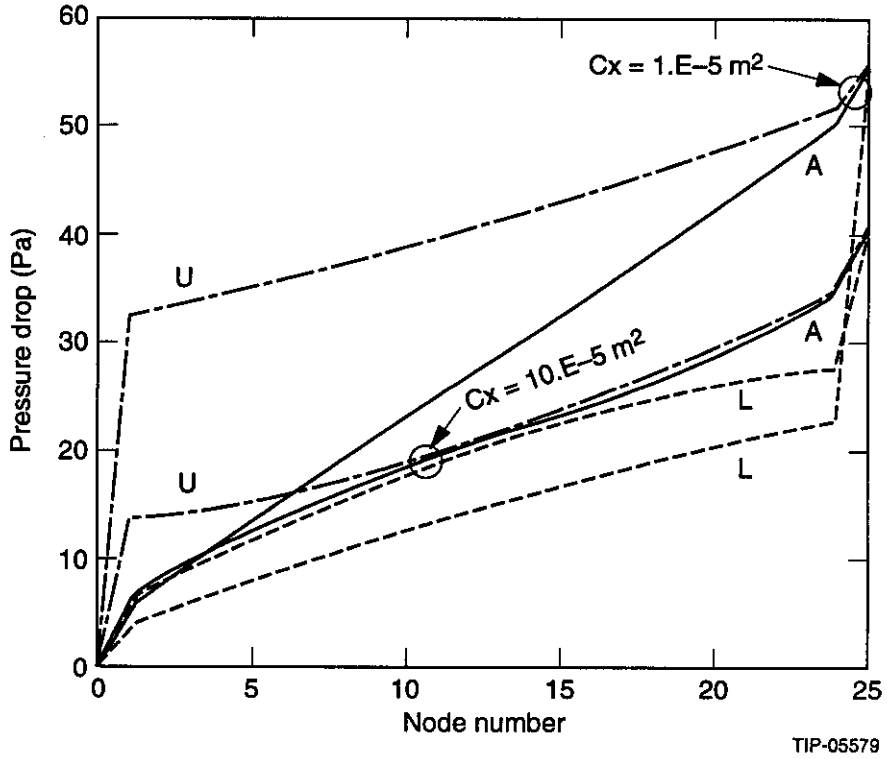

Figure 2. Pressure Drops Along a Dipole Magnet with Crossflow.

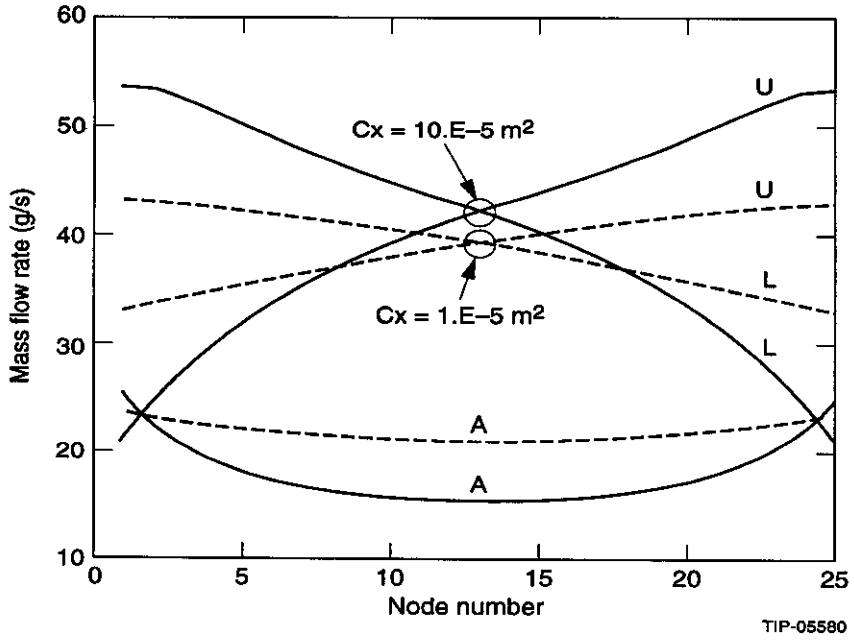

Figure 3. Mass Flow Rates Along a Dipole Magnet with a Crossflow.

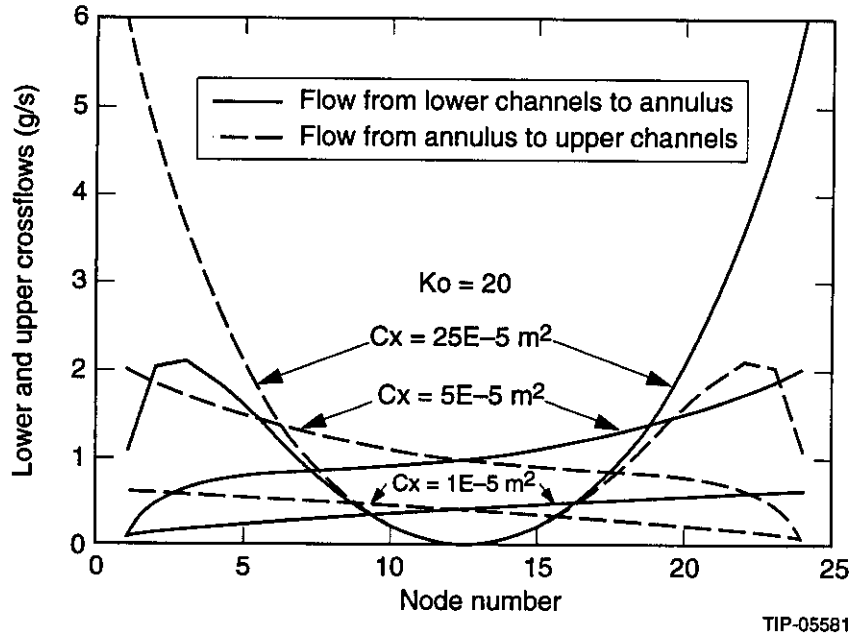

Figure 4. Lower and Upper Crossflows Along a Dipole Magnet.

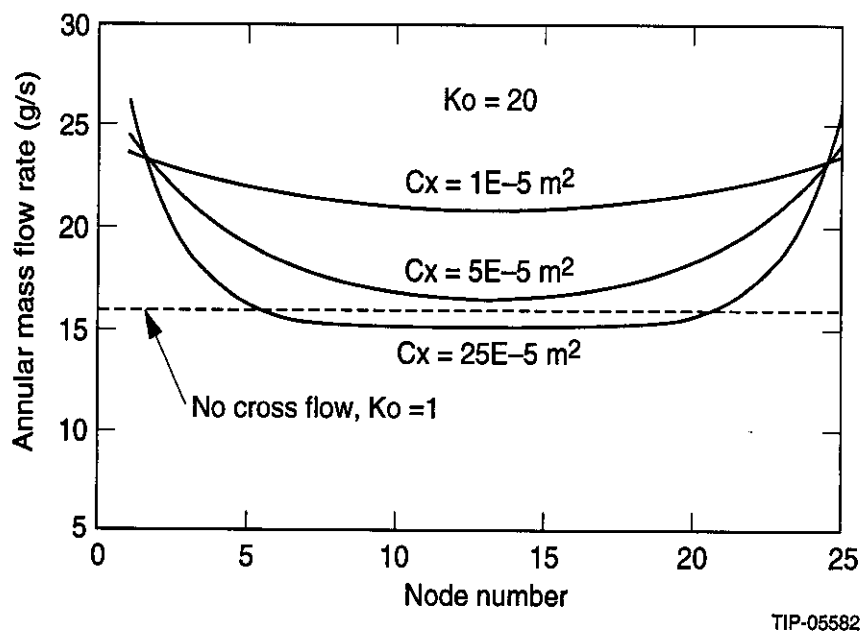

Figure 5. Annular Mass Flow Rate Along a Dipole Magnet.

The variations in the annular channel mass flow rate discussed above have a strong influence on the temperature profiles and the maximum temperature attained in the annular channel (Figure 6). As expected, the temperature rise in the annular channel is lower everywhere compared to that for a magnet without crossflow. However, for higher conductance values, the location of the peak temperature in the annular channel shifts from the end of the magnet toward the middle. Furthermore, the peak temperature level rises rapidly with increasing crossflow conductance as the center of the magnet is deprived of crossflow.

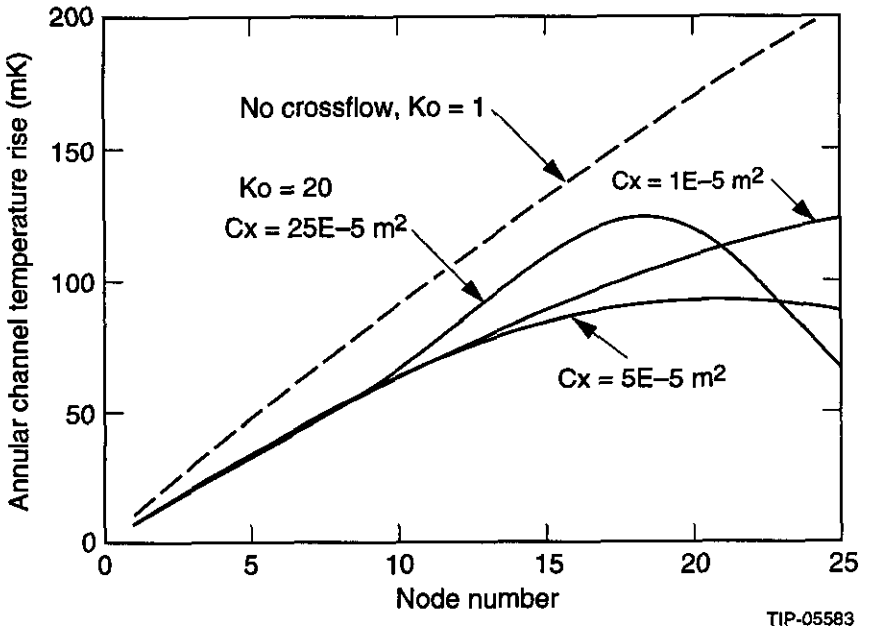

Figure 6. Annular Channel Temperature Rise Along a Dipole Magnet with Crossflow.

3.2 Total Pressure Drop and Maximum Annular Channel Temperature Rise

This section presents a summary of the total pressure drops and the maximum annular channel temperatures for different orifice loss coefficients. For a magnet without crossflow cooling (i.e., with non-orificed cooling channels), the pressure drop is about 30 Pa with 15.9 g/s of flow in the annular channel; the increase in the annular channel helium temperature at the exit is 204 mK (per W/m of heating). The effectiveness of crossflow cooling in a magnet with a particular crossflow conductance and orifice area is judged based on comparisons with the above set of numbers. Additional constraints are the maximum allowable total pressure drop (60 Pa) and coil temperature (4.25 K).

Figure 7 shows the total pressure drop for a dipole magnet and the maximum temperature rise attained in the annular channel for four different orifice coefficients. As expected, the total pressure drop is higher for a magnet with orificed cooling channels; the smaller the orifice (higher Ko), the larger the pressure drop. The effect of crossflow is to reduce the total pressure drop: increased communication between the channels leads towards equalization of pressures; hence the pressure drop decreases as crossflow conductance increases. The reduction in the pressure drop is relatively small for lower Ko values but becomes significant for higher Ko values (greater blockage). A point of diminishing returns is reached around Cx = 10E-5 m^2; beyond this conductance, the decrease in the pressure drop is negligible for all Ko values.

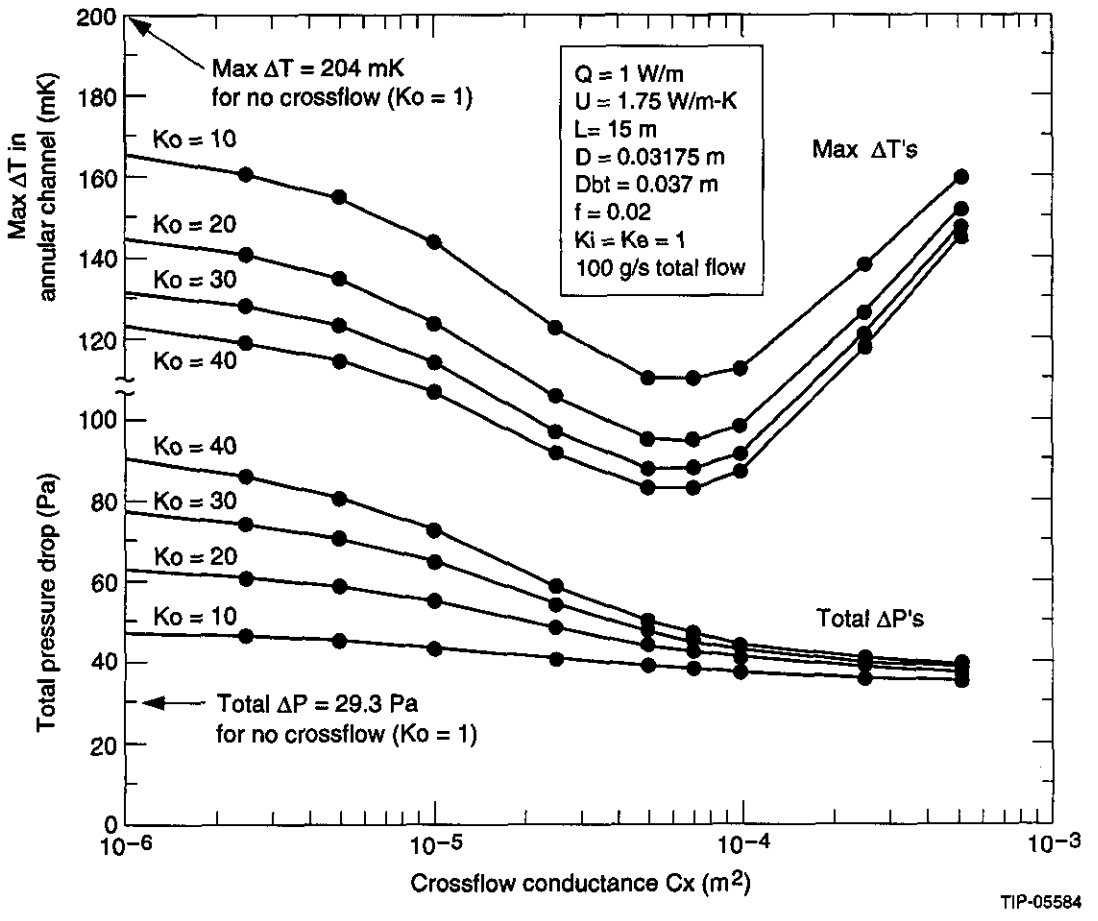

Figure 7. Annular Channel Maximum Temperature Rise and Total Pressure Drop for a Dipole Magnet with Crossflow.

For a given Ko value, the maximum temperature rise in the annular channel decreases initially with increasing crossflow conductance, but increases rapidly for higher conductance values. There is a narrow range of Cx values (4.0 E-5 to 10.0 E-5 m^2) for which the temperature rise in the annular channel is minimized (and significantly reduced compared to the no-crossflow case). Within this range of Cx values, Ko values between 20 and 30 yield the highest reduction in the maximum temperature with "minimal" increase in the total pressure drop. Further reductions in the maximum temperature come at the expense of increasingly higher pressure drops for Ko > 30.

3.2.1 Sensitivity to Coldmass Heat Transfer Conductance

The above results are not very sensitive to the effective heat transfer conductance for the cold mass. Calculations using a value of U = 1.2 W/m-K yield annular channel temperatures slightly higher than those presented above.

3.2.2 System Behavior for Large Ko

As Ko is increased further (i.e., orifice diameter decreased), the total pressure drop increases rapidly, particularly at lower conductance values (Figure 8). However, the decrease in the maximum temperature rise in the annular channel with increasing crossflow conductance declines. Eventually, as Ko $\rightarrow \infty$ (cooling channel blocked fully at the inlet or exit), the maximum temperature increases monotonically with Cx (Figure 9). This is because as the orifice area is reduced, almost all of the flow is channeled into the annulus. Consequently, the temperature rise in the annular channel remains low at lower crossflow conductance values. As the conductance increases, crossflows reduce the annular channel mass flow rate, causing the annular channel temperature to rise.

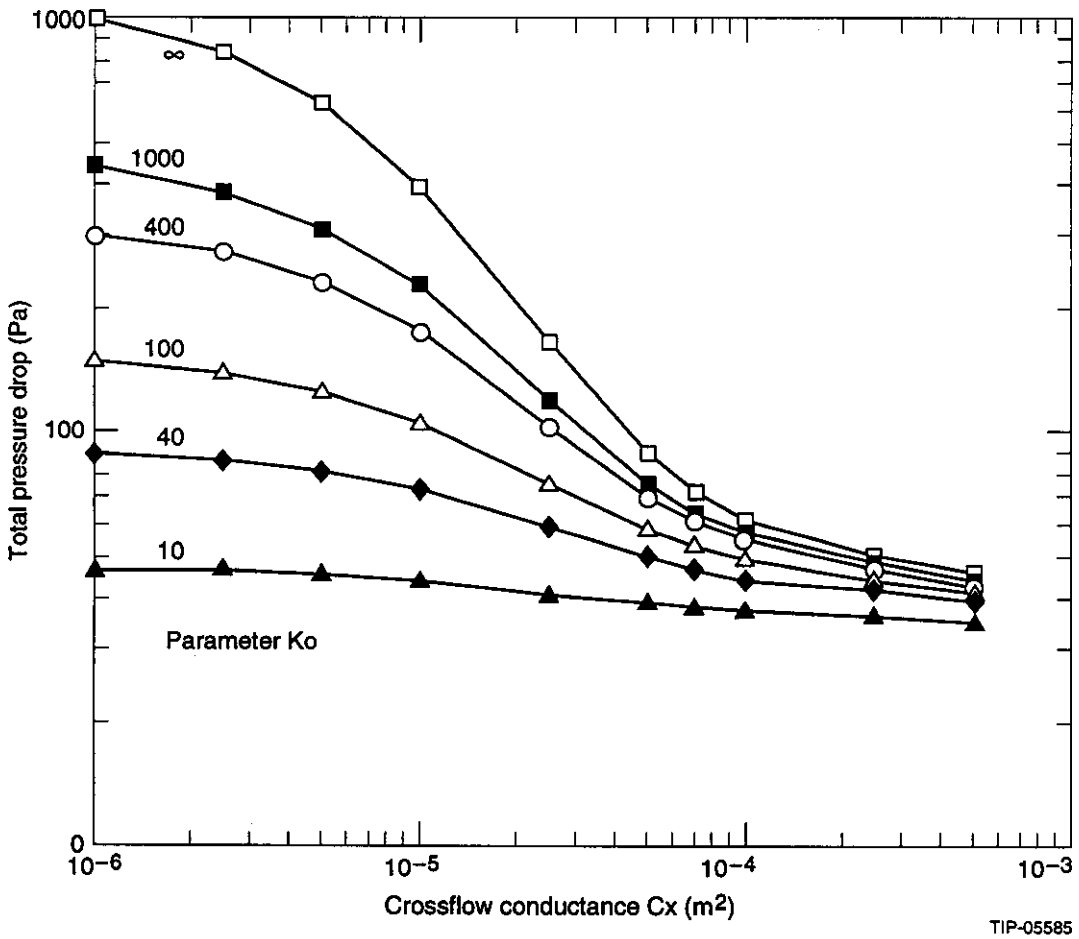

Figure 8. Total Pressure Drop for a Dipole Magnet with Crossflow.

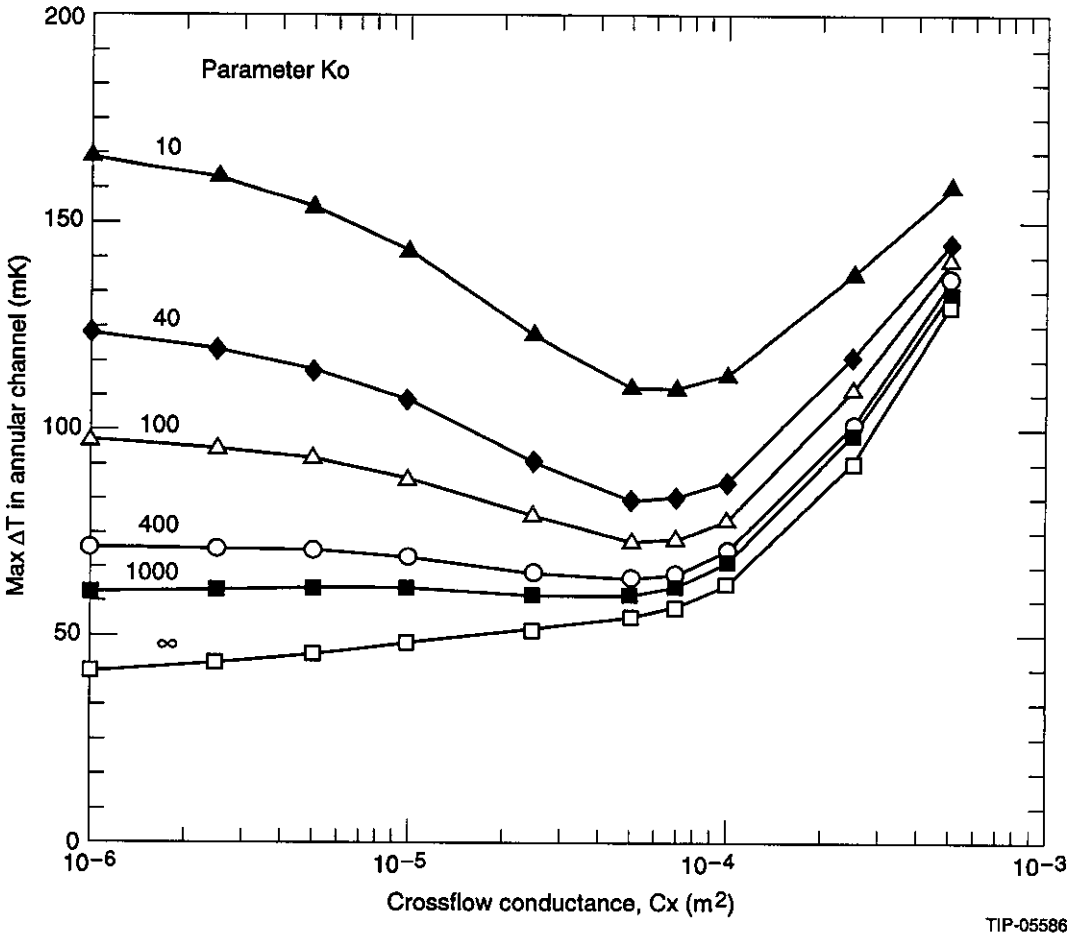

Figure 9. Annular Channel Maximum Temperature Rise for a Dipole Magnet with Crossflow.

3.2.3 Effect of Increased Beam Tube Diameter

The baseline beam tube outer diameter is 37 mm in this analysis. In order to quantify the effects of a reduced annular gap size, simulations were carried for beam tube outer diameters of 42 mm and 45 mm. The friction factor for the annular channel was increased from 0.020 to 0.025 in these simulations to account for the increased flow resistance in the annular channel. Without crossflow, the annular channel mass flow rate is 8.1 g/s for a beam tube diameter of 42 mm, and 4.4 g/s for a beam tube diameter of 45 mm. The corresponding total pressure drops are 35 Pa and 38 Pa, and the maximum annular channel temperature rises (per W/m of linear heating) are 334 mK and 466 mK, respectively. The quantitative effects of crossflow on the total pressure drop and the maximum temperature rise in the annular channel for the three beam tube diameters can be compared in Figures 10 and 11. As expected the annular channel mass flow rate decreases and the total pressure drop increases as the beam tube diameter is increased. The increase in the pressure drop is more pronounced for larger Ko, i.e., for smaller orifices (Figure 10). For a given Ko, the maximum temperature rises in the annular channel can be considerably higher for the larger beam tube diameters as a result of the reduced flow rates in the annular channel (Figure 11). While the annular temperature rise can be reduced using smaller orifices (larger Ko), the accompanying total pressure drop may be prohibitive.

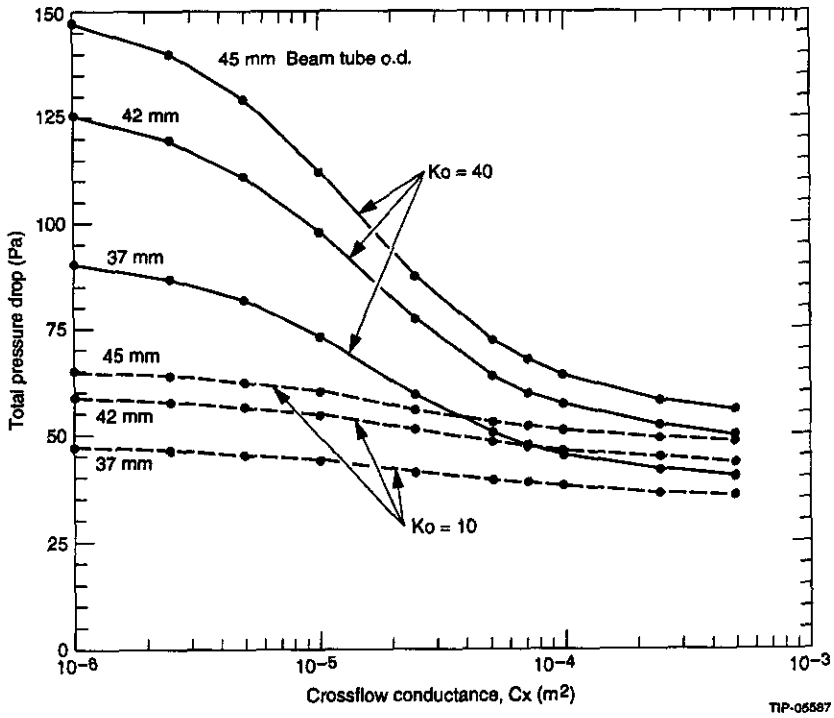

Figure 10. Total Pressure Drop for a Dipole Magnet with Crossflow, for Three Beam Tube Diameters.

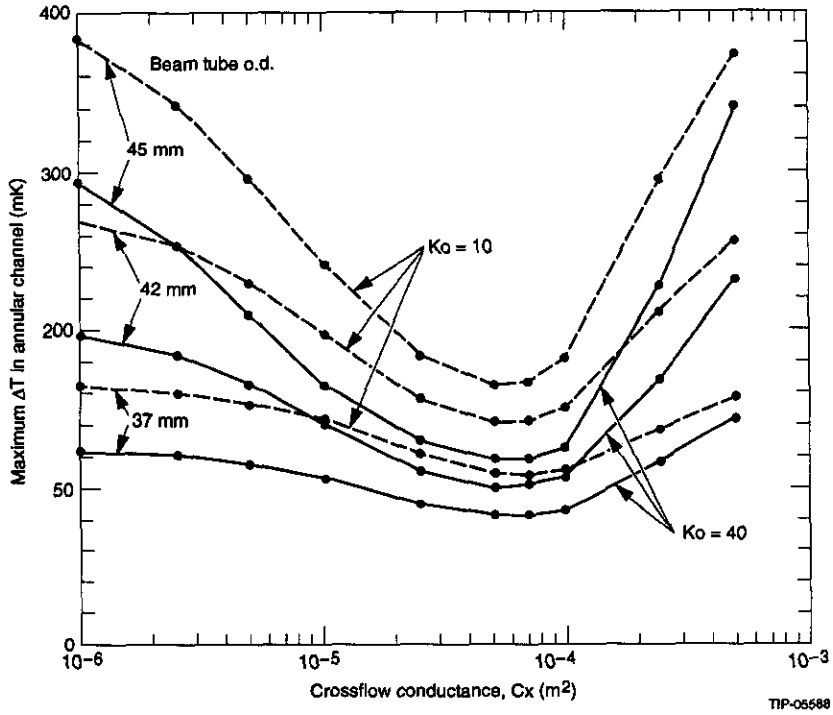

Figure 11. Annular Channel Maximum Temperature Rise for a Dipole Magnet with Crossflow, for Three Beam Tube Diameters.

4.0 SUMMARY

A network flow model for crossflow in a collider dipole magnet with orificed cooling channels is developed. A parametric study of the effects of transverse flow conductance and orifice loss coefficient is conducted. Crossflow cooling is found to be most effective for values within a narrow range of these parameters.

5.0 NOMENCLATURE

A, A_a	Cross-sectional areas for the cooling channels and the annular channel, m^2
Cp	Specific heat, J/kg-K
Cx	Crossflow conductance between the lower channels and the annular channel, and between the annular channel and the upper channels, m^2
D, D_a	Hydraulic diameters for the cooling channels and for the annular channel, m
f	Friction factor
Ki, Ke	Inlet and exit loss coefficients
Ko	Orifice loss coefficient (inlet or exit)
MA	Annular mass flow rate, kg/s
MU, ML	Combined mass flow rate for the upper two channels and for the lower two channels, kg/s
MT	Total mass flow rate, kg/s
MXU, MXL	Upper and lower crossflow rates, kg/s
N	Number of sections
P_0	Pressure at inlet, Pa
PU, PL, PA	Upper, lower, and annular channel pressures, Pa
Q	Linear heat input, W/m
T_0	Temperature at inlet, K
TU, TL, TA	Upper, lower, and annular channel temperatures, K
U	Effective heat transfer conductance for the cold mass, W/m-K
x_i	Axial location for node i ($= i\Delta x$), m
ΔP_i	Cumulative pressure drop at node i ($= P_0 - P(x_i)$), Pa
Δx	Section length, m
δ_{ij}	Kronecker delta
ρ	Density, kg/m^3
i	quantity for section (or node) i

REFERENCES

1. General Dynamics Space Systems Division, Collider Dipole Magnets Preliminary Design Review, San Diego, CA, March 23–27, 1992.
2. E. Fried and I. E. Idelchik, *Flow Resistance*, Hemisphere Publ. Co., New York, 1989.
3. R. Carcagno, "Temperature Rise in the SSC Dipole Coils under Synchrotron Radiation Heating, Preliminary Analysis," SSC-N-514, May 27, 1988.
4. B. Aksel, "Parametric Study for Coil Temperatures in 50-mm CDM and 50-mm CQM," in *Study on Beam Tube Vacuum with Consideration of Synchrotron Light, Potential Liner Intercept, and Collider Quad/Spool Coil Diameter*, H. T. Edwards, ed., SSC-N-771, August 1991.

APPENDIX

xflow.f

```
c-----------------------------------------------------------------------
c
C PROGRAM XFLOW
C
C AUTHOR:
c  Adnan Yucel, Cryogenics Systems Department, SSC Laboratory, Dallas, Texas.
c  May 1992. Revised September 1992, April 1994.
c
c Crossflow cooling analysis program
c to predict pressure drops and temperature rises
c in the annular and main cooling channels of collider dipole magnets
c (The upper channels are orificed at the inlet, the lower channels at the exit) c
c For further information, see
c A. Yucel, "Analysis of Crossflow Cooling in 50mm-CDM with Orificed Channels",
c  SSC Cryo Note 92-5, 1992.
c
c-----------------------------------------------------------------------
c
c SUBROUTINES REQUIRED:
c
c The program needs to be linked with the following subroutines:
c
c decsol.f      : Solver for a linear system of algebraic equations [Ref1]
c dnsqe.f       : Solver for a nonlinear system of algebraic equations [Ref2]
c rkf45.f       : Runge-Kutta ODE integrator [Ref1]
c
c References
c Ref1          : G. Forsythe, M. Malcolm, and C. Moler,
c                 "Computer Methods for Mathematical Computations,"
c                  Prentice-Hall, New York, 1977.
c Ref2          : D. Kahaner, C. Moler, and S. Nash,
c                 "Numerical Methods and Software,"
c                  Prentice-Hall, New York, 1989.
c
c-----------------------------------------------------------------------
c
c INPUT PARAMETERS (INTERACTIVE INPUT):
c
c  fiul         : name of output file where results are written
c  ck           : inlet and/or outlet loss coefficient for the annular channel,
c                  and for the upper and lower cooling channels w/o orifices
c  co           : loss coefficient for the orifice (at the inlet of the upper
c                  channels, at the exit of the lower channels)
c  cxmult       : ( = crossflow conductance x 1.0e+5) m^2
c
c  init         : 0 of cold start; 1 to read solution of a previous case
c                                 (from file XINIT) to use as an initial guess
c
c OTHER PARAMETERS:
c
c  XINIT        : name  of output file which contains the solution vector
c                  for the present set of parameters ( which can be used
c                  as an initial guess for solution vector for another set)
c  dh1, dh2     : hydraulic diameters for the upper & lower cooling channels [m]
c  dbt, dic     : beam tube od and annular channel od (coil id) [m]
c  tl           : length of magnet [m]
c  n            : number of sections along length of magnet
c                 (therefore, number of crossflow paths = n-1)
c  rho          : density of helium at inlet conditions
c  totm         : total mass flow rate [kg/s]
c  pt0          : absolute pressure at inlet plenum [Pa]
c  p0           : relative pressure at inlet plenum [=0 Pa]
c  ff1,ff2,ff3  : friction factors for the upper, lower and annular channels
c  u            : effective heat transfer coefficient between the annular
c                 and main cooling channels [W/m-K]
c  qperm        : synchrotron heat load per unit length [W/m].
c                 Use qperm = 1.0 W/m; temperature results for this case can be
c                 scaled for obtain temperature rises for other qperm values
c  cp           : specific heat of helium at inlet conditions [W/g-K]
c
c
c FORMAT AND CONTENTS OF OUTPUT FILE (NAME ="fiul"):
c
c The output file contains 15 columns, written in the following format
c 6100  format(1x,i3,3(f8.3),3(f8.3),6f9.3,2f10.5)
c
c Column        1 : node/section number (i)
c Column    2/3/4 : Upper/Lower/Annular channel pressure drops
c                   from inlet to end of section i (Pa)
c Column    5/6/7 : Upper/Lower/Annular channel mass flows in section i (g/s)
c Column   8/9/10 : Upper/Lower/Annular channel temperature rises
c                   to end of section i, prior to mixing (mK)
c Column 11/12/13 : Upper/Lower/Annular channel temperature rises
c                   to end of section i, after mixing (mK)
c Column       14 : Crossflow from annular to upper channel (g/s)
c Column       15 : Crossflow from lower to annular channel (g/s)
c
c-----------------------------------------------------------------------
c
c
      implicit double precision (a-h,o-z)
      parameter (n=25,nm1=n-1,neq=5*n+3,lwa=(3*neq**2+13*neq)/2)
      dimension fvec(neq),warray(lwa),fjac(neq,neq)
      dimension decwk(6*neq+3),ipvt(neq)
      dimension xflw(2,nm1)
      common/arrayx/x(neq)
      common/coefio/ci1,ci2,ci3,co1,co2,co3
      common/coeffr/cf1,cf2,cf3
      common/coefxf/cx31,cx23
      common/global/p0,totm
      external fcn,jac
c
      parameter(ntemp=3)
      dimension temp(ntemp),wk(6*ntemp+3),iwk(5)
      dimension ca(ntemp,ntemp),rhsv(ntemp),
     *          decwkt(6*ntemp+3),ipvtt(ntemp)
      dimension temps(ntemp,0:n)
      common/coefxt/xm1,xm2,xm3,cp,u,qperm
      external ftemp
      character*16 fiul
c
c  output file
      write(*,*) 'enter output file name'
      read(*,*) fiul
      open(11,file=fiul,status='unknown',form='formatted')
c
c  geometry data
      dh1 = 0.03175d0
      dh2 = 0.03175d0
      dbt = 0.037d0
      dic = 0.050d0
      dh3 = dic - dbt
      pi  = 4.0d0*datan(1.0d0)
      ar1 = 0.25*pi*dh1**2
      ar2 = 0.25*pi*dh2**2
      ar3 = 0.25*pi*(dic**2 - dbt**2)
c
      tl   = 15.0d0
```

```
      dx   = tl/dfloat(n)
c
c  global data
      rho  = 137.8d0
      totm = 0.10d0
      pt0  = 4.0d5
      p0   = pt0-pt0
      thou = 1.0d3
      totm = totm*thou
c
c  friction factors and coefficients
      ff1 = 0.02d0
      ff2 = 0.02d0
      ff3 = 0.02d0
      cf1 = 0.125d0*ff1*dx/(rho*dh1*ar1**2)/(thou**2)
      cf2 = 0.125d0*ff2*dx/(rho*dh2*ar2**2)/(thou**2)
      cf3 = 0.500d0*ff3*dx/(rho*dh3*ar3**2)/(thou**2)
c
c  read inlet and outlet loss coefficients
      write(*,*) 'enter ck,cor,cxmult'
      read(*,*) ck,co,cxmult
      cki = ck
      cko = ck
      cor = co
      ci1 = 0.125d0*cor/(rho*ar1**2)/(thou**2)
      ci2 = 0.125d0*cki/(rho*ar2**2)/(thou**2)
      ci3 = 0.500d0*cki/(rho*ar3**2)/(thou**2)
      co1 = 0.125d0*cko/(rho*ar1**2)/(thou**2)
      co2 = 0.125d0*cor/(rho*ar2**2)/(thou**2)
      co3 = 0.500d0*cko/(rho*ar3**2)/(thou**2)
c
c  crossflow conductances
c  free parameter
      cx23   = (1.0d-5*cxmult)**2*rho*thou**2
      cx31   = cx23
      write(*,*) 'cx31,cx23',cx31,cx23
c  initial guess
      nout = 8
      open(nout,file='XINIT',form='unformatted',status='unknown')
      write(*,*) 'enter 0 to cold start, 1 to read from file'
      read(*,*) init
      if (init.eq.0)  then
         call xinit
      else
         read(nout) (x(i),i=1,neq)
      endif
c
c  set tol to the square root of the machine precision.
c  unless high precision solutions are required,
c  this is the recommended setting by authors of dnsqe.
      tol = dsqrt(d1mach(4))
c  set iopt = 2 for no jacobian, iopt = 1 if jacobian supplied
      iopt = 1
      nprint = 0
c
      tol = 1.0d-12
      write (*,*) tol
      call dnsqe(fcn,jac,iopt,neq,x,fvec,tol,nprint,info,warray,lwa)
      fnorm = denorm(n,fvec)
      write (*,*) tol
      write (*,1000) fnorm,info
      if (info.ne.1)  stop
c
c  save solution in XINIT
      rewind(nout)
      write(nout) (x(i),i=1,neq)
c
check condition of jacobian
      call fcn(neq,x,fvec,iflag)
      sum = 0.0d0
      do 9999  i = 1,neq
         sum = sum + fvec(i)**2
9999  continue
      call jac(neq,x,fvec,fjac,neq,iflag)
c optional statements to output fvec and jacobian
c      open(51,file='fvec.dat',status='unknown',form='formatted')
c      open(52,file='fjac.dat',status='unknown',form='formatted')
c      do 9901  i = 1,neq
c         write(51,*) i,x(i),fvec(i)
c9901  continue
c      write(*,*) ' enter 1 to print jacobian'
c      read(*,*) ijac
c      ijac = 0
c      if (ijac.eq.1)  then
c      do 9902  i = 1,neq
c         write(*,98) i,(fjac(i,j),j=1,neq)
c98    format(/1x,i2,13(1x,1pe11.4))
c9902  continue
c      endif
c      do 9903  j = 1,neq
c      do 9903  i = 1,neq
c         write(52,*) i,j,fjac(i,j)
c9903  continue
      call decomp(neq,neq,fjac,cond,ipvt,decwk)
      write(*,*) '** sum=',sum,'  cond =',cond
c
6100  format(1x,i3,3(f8.3),3(f8.3),6f9.3,2f10.5)
1000  format (5x,' final l2 norm of the residuals',e15.7/
     *        5x,' exit parameter',i10 /)
c
c  temperature analysis
c
      no = 11
      u = 1.75d0
      qperm = 1.0d0
      cp    = 3.74d0
c
      rerr  = 1.0d-9
      aerr  = 1.0d-8
      iflag = 1
      xin   = 0.0d0
      write(*,*)
      tempin  = 0.0d0
      temp(1) = tempin
      temp(2) = tempin
      temp(3) = tempin
      temps(1,0) = tempin
      temps(2,0) = tempin
      temps(3,0) = tempin
c
      do 100  i = 1,n
         i5  = 5*i
         im1 = i5-1
         im2 = i5
         ip1 = i5+1
         ip2 = i5+2
         ip3 = i5+3
c
```

```
c  compute temperatures
        xout = xin + dx
        xm1 = x(im1)
        xm2 = x(im2)
        xm3 = totm - x(im1) - x(im2)
        call rkf45(ftemp,ntemp,temp,xin,xout,rerr,aerr,iflag,wk,iwk)
        if (iflag.ne.2)  write(*,*) '!! WARNING, iflag NE 2 !!'
c
        if (i.lt.n)  then
           qtot1  = qperm*dx
           qconv1 = xm1*cp*(temp(1)-temps(1,i-1))
           qconv2 = xm2*cp*(temp(2)-temps(2,i-1))
           qconv3 = xm3*cp*(temp(3)-temps(3,i-1))
           qconv  = qconv1+qconv2+qconv3
c           write(*,*)  i,qtot1,qconv,qconv1,qconv2,qconv3
c  compute crossflows
           p3mp1 = x(ip3) - x(ip1)
           ara31 = dsign(1.0d0,p3mp1)*(cx31*dabs(p3mp1))**0.5d0
           xflw(1,i) = ara31
           p2mp3 = x(ip2) - x(ip3)
           ara23 = dsign(1.0d0,p2mp3)*(cx23*dabs(p2mp3))**0.5d0
           xflw(2,i) = ara23
c  compute new initial temperatures using energy balances at mixing nodes
           xm1p5 = x(im1+5)
           xm2p5 = x(im2+5)
           xm3p5 = totm - xm1p5 - xm2p5
c  coefficients
           ca(1,1) = xm1p5 + dmax1(-ara31,0.0d0)
           ca(1,2) = 0.0d0
           ca(1,3) =-dmax1(+ara31,0.0d0)
           ca(2,1) = 0.0d0
           ca(2,2) = xm2p5 + dmax1(+ara23,0.0d0)
           ca(2,3) =-dmax1(-ara23,0.0d0)
           ca(3,1) =-dmax1(-ara31,0.0d0)
           ca(3,2) =-dmax1(+ara23,0.0d0)
           ca(3,3) = xm3p5 + dmax1(+ara31,0.0d0) + dmax1(-ara23,0.0d0)
c
           call decomp(ntemp,ntemp,ca,cond,ipvtt,decwkt)
           condp1 = cond + 1.0d0
           if (condp1.eq.cond)  then
              write(*,*) '  cond =',cond
              stop
           endif
           rhsv(1) = xm1*temp(1)
           rhsv(2) = xm2*temp(2)
           rhsv(3) = xm3*temp(3)
           call solve(ntemp,ntemp,ca,rhsv,ipvtt)
           tm1 = rhsv(1)
           tm2 = rhsv(2)
           tm3 = rhsv(3)
           write(*,6100)  i,-x(ip1),-x(ip2),-x(ip3),x(im1),x(im2),xm3,
     *        temp(1)*1.d3,temp(2)*1.d3,temp(3)*1.d3,
     *        tm1*1.d3,tm2*1.d3,tm3*1.d3,ara31,ara23
           write(no,6100)  i,-x(ip1),-x(ip2),-x(ip3),x(im1),x(im2),xm3,
     *        temp(1)*1.d3,temp(2)*1.d3,temp(3)*1.d3,
     *        tm1*1.d3,tm2*1.d3,tm3*1.d3,ara31,ara23
c
           temp(1) = tm1
           temp(2) = tm2
           temp(3) = tm3
           qtot1  = qperm*dx
           qconv1 = x(im1+5)*cp*temp(1)-xm1*cp*temps(1,i-1)
           qconv2 = x(im2+5)*cp*temp(2)-xm2*cp*temps(2,i-1)
           qconv3 = (totm-x(im1+5)-x(im2+5))*cp*temp(3)
     *                                   -xm3*cp*temps(3,i-1)
           qconv  = qconv1+qconv2+qconv3
c           write(*,*)  i,qtot1,qconv,qconv1,qconv2,qconv3
           temps(1,i) = temp(1)
           temps(2,i) = temp(2)
           temps(3,i) = temp(3)
        else
           write(*,6100)  i,-x(ip1),-x(ip2),-x(ip3),x(im1),x(im2),xm3,
     *        temp(1)*1.d3,temp(2)*1.d3,temp(3)*1.d3
           write(no,6100)  i,-x(ip1),-x(ip2),-x(ip3),x(im1),x(im2),xm3,
     *        temp(1)*1.d3,temp(2)*1.d3,temp(3)*1.d3
           temps(1,i) = temp(1)
           temps(2,i) = temp(2)
           temps(3,i) = temp(3)
           qtot1  = qperm*dx
           qconv1 = xm1*cp*(temp(1)-temps(1,i-1))
           qconv2 = xm2*cp*(temp(2)-temps(2,i-1))
           qconv3 = xm3*cp*(temp(3)-temps(3,i-1))
           qconv  = qconv1+qconv2+qconv3
c           write(*,*)  qtot1,qconv,qconv1,qconv2,qconv3
c  energy check
           qtot1 = qperm*t1
           qconv1 = xm1*cp*temp(1)
           qconv2 = xm2*cp*temp(2)
           qconv3 = xm3*cp*temp(3)
           qconv = qconv1+qconv2+qconv3
           write(*,*)  qtot1,qconv,qconv1,qconv2,qconv3
        endif
100   continue
c
      stop
      end
c
c-----------------------------------------------------------------------
c
      subroutine fcn(neq,x,fvec,iflag)
      implicit double precision (a-h,o-z)
      dimension x(neq),fvec(neq)
      parameter (n=25)
c
      common/coefio/ci1,ci2,ci3,co1,co2,co3
      common/coeffr/cf1,cf2,cf3
      common/coefxf/cx31,cx23
      common/global/p0,totm
c
      fvec(1) = x(1) - p0
      fvec(2) = x(2) - p0
      fvec(3) = x(3) - p0
c
      do 10  i = 1,n
         i5  = 5*i
         im1 = i5-1
         im2 = i5
         ip1 = i5+1
         ip2 = i5+2
         ip3 = i5+3
c
         if (i.eq.1)  then
            ct1 = cf1 + ci1
            ct2 = cf2 + ci2
            ct3 = cf3 + ci3
         elseif (i.eq.n)  then
            ct1 = cf1 + co1
            ct2 = cf2 + co2
```

```
            ct3 = cf3 + co3
         else
            ct1 = cf1
            ct2 = cf2
            ct3 = cf3
         endif
c
         fvec(i5-1) = x(ip1) - x(ip1-5) + ct1*dabs(x(im1))*x(im1)
         fvec(i5  ) = x(ip2) - x(ip2-5) + ct2*dabs(x(im2))*x(im2)
         fvec(i5+1) = x(ip3) - x(ip3-5)
     *              + ct3*dabs(totm-x(im1)-x(im2))*(totm-x(im1)-x(im2))
c
         if (i.lt.n)  then
            p3mp1 = x(ip3) - x(ip1)
            fvec(i5+2) = dsign(1.0d0,p3mp1)*(cx31*dabs(p3mp1))**0.5d0
     *                 - (x(im1+5) - x(im1))
            p2mp3 = x(ip2) - x(ip3)
            fvec(i5+3) = dsign(1.0d0,p2mp3)*(cx23*dabs(p2mp3))**0.5d0
     *                 - (x(im2) - x(im2+5))
         endif
10    continue
c
         i5  = 5*n
         ip1 = i5+1
         ip2 = i5+2
         ip3 = i5+3
      fvec(i5+2) = x(ip3) - x(ip1)
      fvec(i5+3) = x(ip2) - x(ip3)
c
c      do 20 i=1,neq
c20      write(*,*) i,x(i),fvec(i)
      return
      end
c
c-----------------------------------------------------------------
c
      subroutine jac(neq,x,fvec,fjac,ldfjac,iflag)
      implicit double precision (a-h,o-z)
      dimension x(neq),fvec(neq),fjac(ldfjac,neq)
      parameter (n=25)
c
      common/coefio/ci1,ci2,ci3,co1,co2,co3
      common/coeffr/cf1,cf2,cf3
      common/coefxf/cx31,cx23
      common/global/p0,totm
c
c  initialize jacobian (assign zero elements)
      do 1  j = 1,neq
      do 1  i = 1,neq
         fjac(i,j) = 0.0d0
1     continue
c
c  nonzero elements
c
c fvec(1) = x(1) - p0
      fjac(1,1) = 1.0d0
c fvec(2) = x(2) - p0
      fjac(2,2) = 1.0d0
c fvec(3) = x(3) - p0
      fjac(3,3) = 1.0d0
c
      do 10  i = 1,n
         i5  = 5*i
         im1 = i5-1
         im2 = i5
         ip1 = i5+1
         ip2 = i5+2
         ip3 = i5+3
c
         if (i.eq.1)  then
            ct1 = cf1 + ci1
            ct2 = cf2 + ci2
            ct3 = cf3 + ci3
         elseif (i.eq.n)  then
            ct1 = cf1 + co1
            ct2 = cf2 + co2
            ct3 = cf3 + co3
         else
            ct1 = cf1
            ct2 = cf2
            ct3 = cf3
         endif
c
c fvec(i5-1) = x(ip1) - x(ip1-5) + ct1*dabs(x(im1))*x(im1)
         fjac(i5-1,ip1)   = 1.0d0
         fjac(i5-1,ip1-5) =-1.0d0
         fjac(i5-1,im1)   = 2.0d0*ct1*dabs(x(im1))
c
c fvec(i5  ) = x(ip2) - x(ip2-5) + ct2*dabs(x(im2))*x(im2)
         fjac(i5  ,ip2)   = 1.0d0
         fjac(i5  ,ip2-5) =-1.0d0
         fjac(i5  ,im2)   = 2.0d0*ct2*dabs(x(im2))
c
c fvec(i5+1) = x(ip3) - x(ip3-5)
c                     + ct3*dabs(totm-x(im1)-x(im2))*(totm-x(im1)-x(im2))
         fjac(i5+1,ip3)   = 1.0d0
         fjac(i5+1,ip3-5) =-1.0d0
         fjac(i5+1,im1)   =-2.0d0*ct3*dabs(totm-x(im1)-x(im2))
         fjac(i5+1,im2)   =-2.0d0*ct3*dabs(totm-x(im1)-x(im2))
c
         if (i.lt.n)  then
c
c p3mp1 = x(ip3) - x(ip1)
c fvec(i5+2) = dsign(1.0d0,p3mp1)*(cx31*dabs(p3mp1))**0.5d0
c            - (x(im1+5) - x(im1))
            p3mp1 = x(ip3) - x(ip1)
            if (p3mp1.ne.0.0d0)  then
               ara31 = 0.5d0*(cx31/dabs(p3mp1))**0.5d0
            else
               ara31 = 0.0d0
            endif
            fjac(i5+2,ip1)   =-ara31
            fjac(i5+2,ip3)   = ara31
            fjac(i5+2,im1)   =+1.0d0
            fjac(i5+2,im1+5) =-1.0d0
c
c p2mp3 = x(ip2) - x(ip3)
c fvec(i5+3) = dsign(1.0d0,p2mp3)*(cx23*dabs(p2mp3))**0.5d0
c            - (x(im2) - x(im2+5))
            p2mp3 = x(ip2) - x(ip3)
            if (p2mp3.ne.0.0d0)  then
               ara23 = 0.5d0*(cx23/dabs(p2mp3))**0.5d0
            else
               ara23 = 0.0d0
            endif
            fjac(i5+3,ip2)   = ara23
            fjac(i5+3,ip3)   =-ara23
            fjac(i5+3,im2)   =-1.0d0
```

```
            fjac(i5+3,im2+5) =+1.0d0
c
          endif
10      continue
c
          i5  = 5*n
          ip1 = i5+1
          ip2 = i5+2
          ip3 = i5+3
c
c fvec(i5+2) = x(ip3) - x(ip1)
          fjac(i5+2,ip1)   =-1.0d0
          fjac(i5+2,ip3)   = 1.0d0
c
c fvec(i5+3) = x(ip2) - x(ip3)
          fjac(i5+3,ip2)   = 1.0d0
          fjac(i5+3,ip3)   =-1.0d0
c
      return
      end
c
c-----------------------------------------------------------------
c
      subroutine xinit
c
      implicit double precision (a-h,o-z)
      parameter (n=25,neq=5*n+3)
      common/arrayx/x(neq)
      common/coefio/ci1,ci2,ci3,co1,co2,co3
      common/coeffr/cf1,cf2,cf3
      common/coefxf/cx31,cx23
      common/global/p0,totm
c
      x40 = 0.5d0*(0.85d0*totm)
      x50 = 0.5d0*(0.85d0*totm)
      x60 = totm - x40 - x50
      x10 = cf1*x40**2
      x20 = cf2*x50**2
      x30 = cf3*x60**2
c
      x(1) = p0
      x(2) = p0
      x(3) = p0
      do 20  i = 1,n
         x(5*i-1)  = x40
         x(5*i  )  = x50
         x(5*i+1)  = x(5*i+1-5) - x10
         x(5*i+2)  = x(5*i+2-5) - x20
         x(5*i+3)  = x(5*i+3-5) - x30
20    continue
c
      return
      end
c
c-----------------------------------------------------------------
c
      subroutine ftemp(x,temp,dtemp)
c
      implicit double precision (a-h,o-z)
      parameter (ntemp=3)
      dimension temp(ntemp),dtemp(ntemp)
      common/coefxt/xm1,xm2,xm3,cp,u,qperm
c
      dtemp(1) = 0.5d0*u*(temp(3)-temp(1))/(xm1*cp)
      dtemp(2) = 0.5d0*u*(temp(3)-temp(2))/(xm2*cp)
      dtemp(3) = (qperm
     *          -0.5d0*u*(temp(3)-temp(1))-0.5d0*u*(temp(3)-temp(2)))
     *           /(xm3*cp)
c
      return
      end
```