

KGS
OF
71-3

**Computer Program to Solve the Three
Dimensional Heat Flow Equation**

by

John C. Halepaska and Fred W. Hartman

**Open-File Report
Kansas Geological Survey**

September 1971

The program solves the following partial differential equation of heat flow

$$\frac{\partial}{\partial x} (\kappa \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y} (\kappa \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z} (\kappa \frac{\partial T}{\partial z}) + Q = C_p \rho \frac{\partial T}{\partial t} \quad (1)$$

where T = temperature $^{\circ}\text{F}$

κ = thermal conductivity $\text{Btu}/\text{T-L-}^{\circ}\text{F}$

x, y, z = cartesian coordinates L

C_p = specific heat $\text{Btu}/\text{lb-}^{\circ}\text{F}$

ρ = density lb/L^3

t = time T

Q = heat generation rate per unit volume $\text{Btu}/\text{L}^3\text{-T}$

The program solves (1) by the Crank-Nicholson finite differencing scheme. For the description of how the program solves (1) see Halepaska and Hartman, 1971, "Numerical Solution of the 3-Dimensional Heat Flow Equation."

The present form of the program assumes the conductivity equal in all directions, but the parameters are indexed so that variable conductivities could easily be incorporated.

The following parameters must be defined at each nodal center (fig. 1).

The initial temperature array:

$T_{i,j,k}$

the specific heat array:

$C_{p,i,j,k}$

the thermal conductivity array:

$\kappa_{i,j,k}$

and the density array:

$$P_{i,j,k}$$

and the source or sink term array:

$$Q_{i,j,k}$$

The arrays are presently set up internally in the program, but the program could be easily modified to read in the arrays of data.

Other parameters, such as the initial time step, termination time, and Δx , Δz spacing are presently defined internally in the program.

At present, the only input is on three cards:

Card #1 - M, L, K where

M = number of nodes along the x direction (fig. 1)

L = number of nodes along the y direction

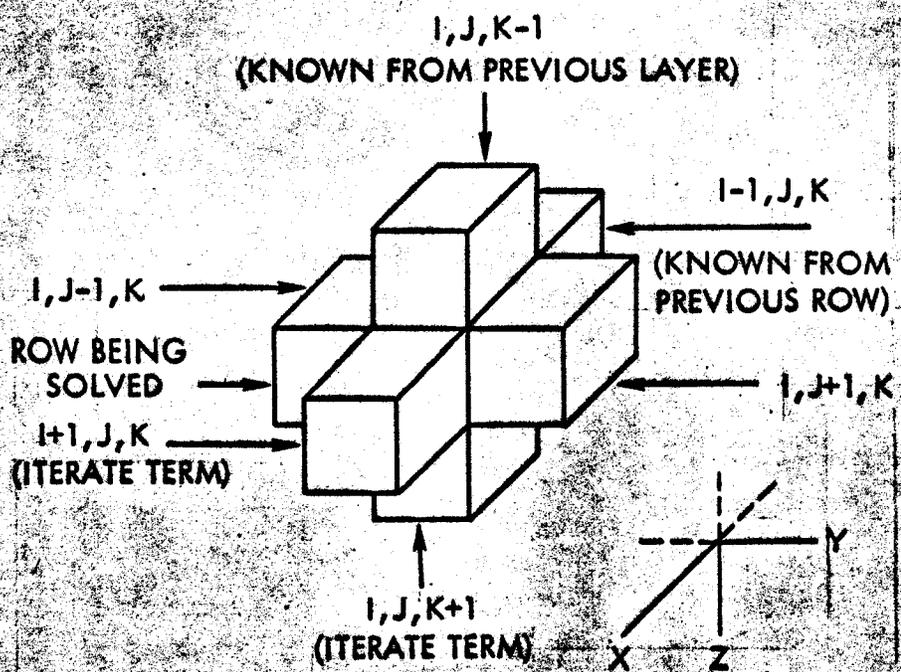
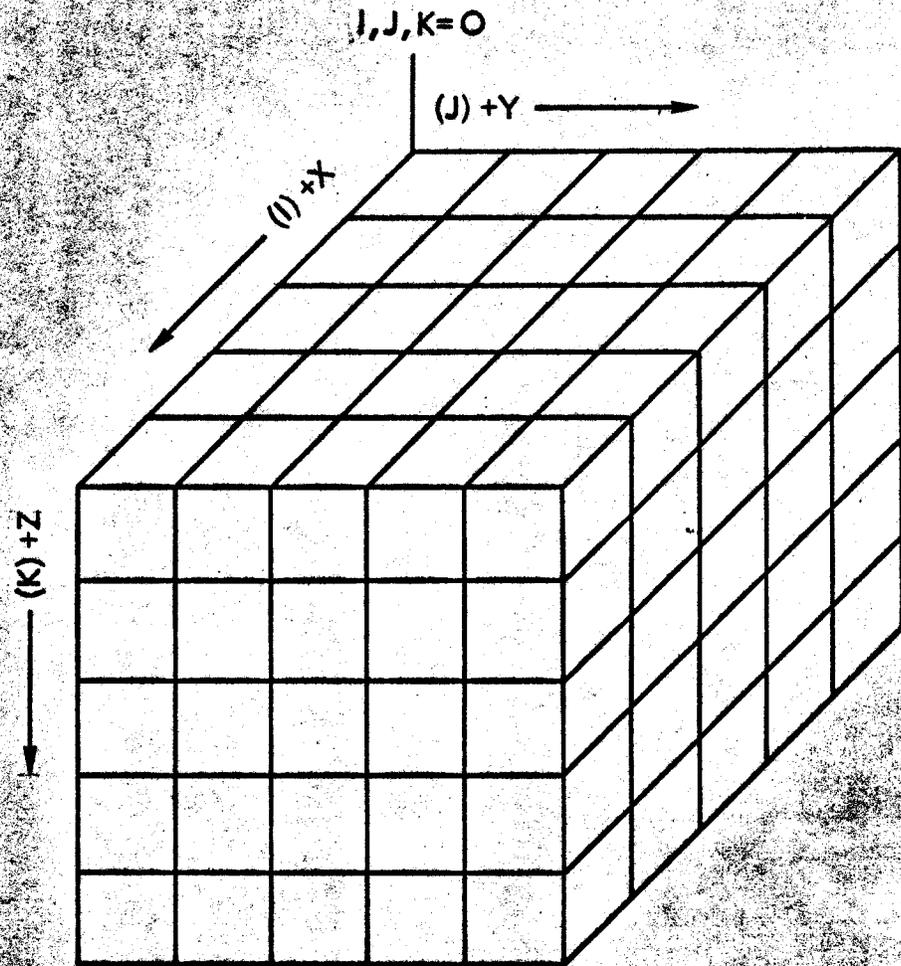
K = number of nodes along the z direction

Format for Card #1 is 314.

Card #2 - N = number of cycles to print out temperature array

Card #3 - ϵ = epsilon value in convergence test of iteration

A listing of an example input is given on page __.



27 1/2 PAGES

References

Halepaska, John C., and Fred W. Hartman, 19--, Numerical Solution
of the 3-Dimensional Heat Flow Equation: Kansas Geol. Survey,
Short Papers on Research in 1971 (in preparation).

14 14 4
5
.000001

```

C   THREE DIMENSIONAL HEAT FLOW XYZ
    DIMENSION A(15),AX(15,15,5),BX(15,15,5),CX(15,15,5),DX(15,15,5),
    IEX(15,15,5),FX(15,15,5),S(15,15,5),H(15,15,5),C(5),CE(15,15,5),
    2SS(15,15,5),U(15),V(15),G(15,15,5),HH(15,15),
    3SK(15,15,5),Q(15,15,5),PPK(15,15,5),E(15,15,5)
    DOUBLE PRECISION A,AX,BX,CX,DX,EX,FX,S,H,CE,SS,U,V,G,HH,SK,Q
    DOUBLE PRECISION E,DDX1,DDX,DDZ1,DDZ,DEL
    READ(5,1)NMZON,NZON,KZON
    1  FORMAT(3I4)
      L=NZON+1
      M=NMZON+1
      KN=KZON+1
    61 FORMAT(1H1)
    62 FORMAT(F12.6,2F10.2)
      DO 200 K=1,KN
      DO 200 I=1,L
      DO 200 J=1,M
C   TEMPERATURE - DEGREES FAHRENHEIT
      H(I,J,K)=65.
C   SPECIFIC HEAT - CP - BTU PER POUND-DEGREE
      CE(I,J,K)=.22
C   HEAT CONDUCTIVITY - K - BTU PER HOUR-FOOT-DEGREE
      SS(I,J,K)=1.81
C   DENSITY - RHO - POUND PER CUBIC FEET
      S(I,J,K)=135.
    200 CONTINUE
      DO 201 I=1,L
      DO 201 J=1,M
      SS(I,J,1)=0.0
      SS(I,J,5)=0.0
    201 CONTINUE
      7  FORMAT(18F4.0)
      6  FORMAT(1X,18(F6.2,1X))
    66  FORMAT(/)
C
      MED=0
      NEDIT=0
C   NED = NUMBER OF CYCLES TO PRINT OUT TEMP MATRIX
      READ(5,1)NED
C   SIGMA = CONVERGENCE LIMIT IN ITERATION LOOP
      READ(5,62)SIGMA
C   RDT = TIME STEP PROGRESSION RATIO
      RDT=1.2
C   INITIAL TIME STEP (TIME IS IN HOURS)
C
      DTNMH=1./60.
      DTNPH=DTNMH
      TIME=0.0
C   TSTOP = ENDING TIME TO TERMINATE PROGRAM
      TSTOP=1.0E+04
C   L = NUMBER OF NODES IN X DIRECTION
C   M = NUMBER OF NODES IN Y DIRECTION
C   KN = NUMBER OF NODES IN Z DIRECTION
C
      COMPUTE RHO TIMES SPECIFIC HEAT
      SET Q MATRIX EQUAL TO ZERO
      DO 10 K=1,KN
      DO 10 I=1,L
      DO 10 J=1,M

```

CE(I,J,K)=CE(I,J,K)*S(I,J,K)

Q(I,J,K)=0.0

10 CONTINUE

A(I) = 1/DELTA X SQUARED = 1/DELTA Y SQUARED

C(K) = 1/DELTA Z SQUARED

DDX1=20.

DDZ1=20.

DDX=1./DDX1**2

DDZ=1./DDZ1**2

DO 12 I=1,L

A(I)=DDX

12 CONTINUE

DO 101 K=1,KN

C(K)=DDZ

101 CONTINUE

Q = TWO TIMES RATE PER CUBIC VOLUME

SET UP LOCATIONS OF SOURCE TERMS

Q(8,8,2)=(2.*20000.)/(DDX1**2*DDZ1)

Q(8,8,3)=(2.*20000.)/(DDX1**2*DDZ1)

Q(8,8,4)=(2.*20000.)/(DDX1**2*DDZ1)

COMPUTE AVERAGE CONDUCTIVITIES ACROSS NODES

15 DO 16 K=2,KZON

DO 16 I=2,NZON

DO 16 J=2,NMZON

AX(I,J,K)=(2.*SS(I,J,K)*SS(I+1,J,K))/(SS(I,J,K)+SS(I+1,J,K))

BX(I,J,K)=(2.*SS(I,J,K)*SS(I-1,J,K))/(SS(I,J,K)+SS(I-1,J,K))

CX(I,J,K)=(2.*SS(I,J,K)*SS(I,J+1,K))/(SS(I,J,K)+SS(I,J+1,K))

DX(I,J,K)=(2.*SS(I,J,K)*SS(I,J-1,K))/(SS(I,J,K)+SS(I,J-1,K))

EX(I,J,K)=(2.*SS(I,J,K)*SS(I,J,K+1))/(SS(I,J,K)+SS(I,J,K+1))

FX(I,J,K)=(2.*SS(I,J,K)*SS(I,J,K-1))/(SS(I,J,K)+SS(I,J,K-1))

16 CONTINUE

DO 18 K=1,KN

DO 18 I=1,L

DO 18 J=1,M

S(I,J,K)=H(I,J,K)

SS(I,J,K)=H(I,J,K)

18 CONTINUE

BEGIN INTEGRATION AND TIME STEP LOOP

20 DTNMM=DTNPH

DTNPH=RTD*DTNMM

TIME=TIME+DTNPH

TEST TO TERMINATE

IF((TIME-TSTOP).GE.0.0)GO TO 125

EXTRAPOLATION FOR ITERATE TERMS

DO 30 K=1,KN

DO 30 I=1,L

DO 30 J=1,M

SS(I,J,K)=S(I,J,K)+RTD*(S(I,J,K)-SS(I,J,K))

SK(I,J,K)=SS(I,J,K)

E(I,J,K)=(2.*CE(I,J,K))/DTNPH

30 CONTINUE

SET UP COEFFICIENTS FOR THOMAS ALGORITHM

DO 35 K=2,KZON

```
DO 35 I=2,NZON
DO 35 J=2,NMZON
G(I,J,K)=A(I)*BX(I,J,K)*(S(I,J,K)-S(I-1,J,K))+A(I)*DX(I,J,K)*
1(S(I,J,K)-S(I,J-1,K))-A(I)*AX(I,J,K)*(S(I+1,J,K)-S(I,J,K))-A(I)*
2CX(I,J,K)*(S(I,J+1,K)-S(I,J,K))+C(K)*FX(I,J,K)*(S(I,J,K)-S(I,J,K-1
3))-C(K)*EX(I,J,K)*(S(I,J,K+1)-S(I,J,K))-Q(I,J,K)
35 CONTINUE
```

```
ITERA=0
```

```
37 DO 72 K=2,KZON
```

```
DO 72 I=2,NZON
DO 40 J=2,NMZON
HH(I,J)=G(I,J,K)-A(I)*AX(I,J,K)*SK(I+1,J,K)-A(I)*BX(I,J,K)*
1SS(I-1,J,K)-C(K)*EX(I,J,K)*SK(I,J,K+1)-C(K)*FX(I,J,K)*SS(I,J,K-1)
2-E(I,J,K)*S(I,J,K)
```

```
40 CONTINUE
```

```
U(1)=0.0
V(1)=S(I,1,K)
DO 60 J=2,NMZON
DEL=A(I)*AX(I,J,K)+A(I)*BX(I,J,K)+A(I)*CX(I,J,K)+A(I)*DX(I,J,K)+
1C(K)*EX(I,J,K)+C(K)*FX(I,J,K)+E(I,J,K)-A(I)*DX(I,J,K)*U(J-1)
IF(DEL.LE.1.0E-20)GO TO 55
COEFFICIENTS FOR LINEAR EQUATIONS IN TRIDIAGONAL MATRIX
```

```
U(J)=A(I)*CX(I,J,K)/DEL
```

```
V(J)=(A(I)*DX(I,J,K)*V(J-1)-HH(I,J))/DEL
GO TO 60
```

```
55 U(J)=0.0
V(J)=S(I,J,K)
```

```
60 CONTINUE
BACK SUBSTITUTION TO SOLVE FOR TEMP VALUES IN JTH ROW
```

```
DO 70 J=2,NMZON
KK=M-J+1
SS(I,KK,K)=U(KK)*SS(I,KK+1,K)+V(KK)
```

```
70 CONTINUE
```

```
72 CONTINUE
```

```
PKMAX=0.0
DO 75 K=1,KN
TEST SUCCESSIVE TEMP VALUES OVER ENTIRE GRID
DO 75 I=1,L
DO 75 J=1,M
PPK(I,J,K)=ABS(1.-SK(I,J,K)/SS(I,J,K))
IF(PPK(I,J,K)-PKMAX)75,73,73
```

```
73 PKMAX=PPK(I,J,K)
```

```
75 CONTINUE
```

```
IF(PKMAX.LE.SIGMA)GO TO 88
ITERA=ITERA+1
IF(ITERA.GT.10)GO TO 88
```

```
DO 80 K=1,KN
DO 80 I=1,L
DO 80 J=1,M
```

SK(I,J,K)=SS(I,J,K)
80 CONTINUE

C
GO TO 37

C
C
TRADE VALUES IN S AND SS ARRAYS

C
88 DO 90 K=1,KN
DO 90 I=1,L
DO 90 J=1,M
TEMP=S(I,J,K)
S(I,J,K)=SS(I,J,K)
SS(I,J,K)=TEMP
90 CONTINUE
MED=MED+1

C
NEDIT=NEDIT+1

DTNMH=DTNPH

C
DETERMINE IF PRINTOUT REQUIRED

IF((NEDIT-NEDIT/NEO*NEO).EQ.0)GO TO 100

IF(NEDIT.LT.2)GO TO 100

GO TO 20

C
100 WRITE(6,105)TIME,DTNPH,NEDIT,ITERA

C
105 FORMAT(1H1,5X,5HTIME=,E11.4,6HDTNPH=,E11.4,6HNEDIT=,I4,6HITERA=,I4
1)

63 FORMAT(1X,18F7.5)

C
118 DO 120 K=1,KN

DO 120 I=1,L

WRITE (6, 6) (S(I, J, K), J = 1, M)

WRITE(6,6)(H(I,J,K),J=1,M)

WRITE(6,63){PPK(I,J,K),J=1,M)

WRITE(6,66)

120 CONTINUE

GO TO 20

C
125 STOP

END