NASA CR-165,982

NASA Contractor Report 165982

NASA-CR-165982
19830002292

NECAP 4.1 - NASA'S ENERGY COST ANALYSIS PROGRAM - THERMAL RESPONSE FACTOR ROUTINE

Michael R. Wiese
Computer Sciences Corporation
Hampton, Virginia

Prepared for
Langley Research Center
under Contract NAS1-16078
August 1982

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

FOREWORD

This manual, which documents the THERMAL RESPONSE FACTOR ROUTINE (RESFAC - version 2) used by NASA'S ENERGY COST ANALYSIS PROGRAM (NECAP), is a suppliment to the NECAP ENGINEERING MANUAL (TM 83240) and the NECAP ENGINEERING FLOWCHARTS MANUAL (TM 83242).

The calculation sequences and flowcharts are given in the same format that is used by the NECAP ENGINEERING MANUAL and the NECAP ENGINEERING FLOWCHARTS MANUAL.

This version of RESFAC is the result of modifications and enhancements made to the original RESFAC. Although the RESFAC programming code has changed, the way in which the routine is accessed and used by NECAP is still the same. Therefore, the RESFAC usage information presented in the NECAP INPUT MANUAL (TM 83239) and the NECAP USERS MANUAL (TM 83238) is still accurate.

TABLE OF CONTENTS

OBJECTIVE AND DESCRIPTION

The Response Factor Program (RESFAC2) generates the time series of heat transfer factors (called response factors) required to accurately determine the transient flow of heat into, through, and out of interior/exterior building surfaces as they react to temperature differences across them.

These response factors are a function of the type of materials used and their order of placement in the surface. It is required that the following material properties be known for each material layer:

for non-air layer-

1. XL, thickness (FT)
2. XK, thermal conductivity (BTU/HR-FT-°F)
3. D , density (LB/FT^3)
4. SH, specific heat (BTU/LB-°F)

for air layer-

5. RES, resistivity (HR-FT^2-°F/BTU)

The sequencing of each layer's material characteristics is important. It must follow the way the surface is layered from the outside material to the inside.

DEFINING THERMAL RESPONSE FACTORS

Thermal response factors can be defined as time dependent heat transfer coefficients which depict an object's action upon a particular outside surface/inside surface temperature difference over a period of time. This action is composed of three terms--the X, Y, and Z response factors.

Using the heat flux equation $Q = U * \Delta T$ along with the response factors, the time dependent heat flux values for the planar construction shown in Figure 1 can be obtained.

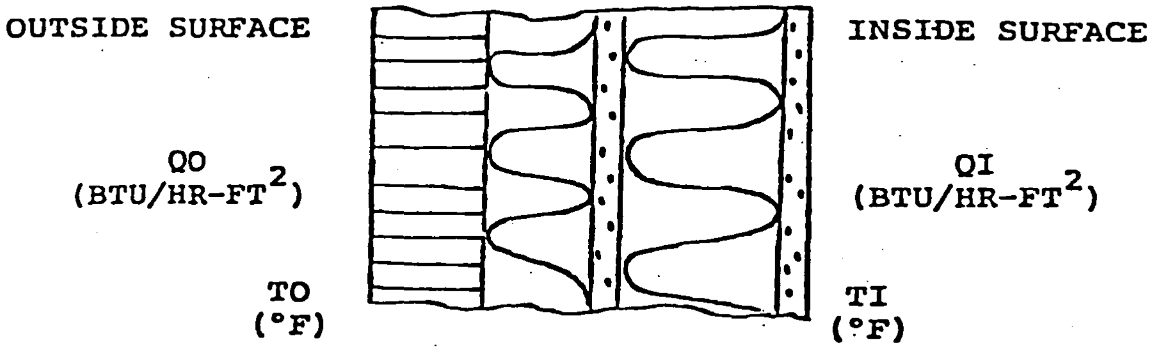


FIGURE 1

The outside surface heat flux (QO) at any time t is calculated by :

$$QO_t = \sum_{i=0}^{NRT} (TO_{t-i} * X_i) - \sum_{i=0}^{NRT} (TI_{t-i} * Y_i)$$

where NRT is the number of response factor terms calculated, TO_{t-i} and TI_{t-i} are outside surface and inside surface temperatures at times t-i hour.

The inside surface heat flux (QI) at any time t is calculated by :

$$QI_t = \sum_{i=0}^{NRT}(TO_{t-i} * Y_i) - \sum_{i=0}^{NRT}(TI_{t-i} * Z_i)$$

In order for the steady state heat conduction situation

$QO_t = QI_t = U(TO - TI)$

$TO = TO_t$ for any time t

$TI = TI_t$ for any time t

to be satisfied, the individual summation of the X, Y, and Z response factors must equal the overall heat transfer coefficient U :

$$U = \left|\sum_{i=1}^{\infty} X_i\right| = \left|\sum_{i=1}^{\infty} Y_i\right| = \left|\sum_{i=1}^{\infty} Z_i\right|$$

ALGORITHMS

The new Response Factor Program is comprised of five program modules:

Main program - RESFAC

Subroutines - RESIDU, ROOTS, and THREBS

Function - BMEQ

The following pages present input/output, calculation sequences, and flowcharts for each of these modules.

RESFAC

A program which determines the thermal response factors for a particular roof, wall, or floor construction.

INPUT

M : number of material layers

XL : thickness of each layer (FT)
if air layer then XL = 0.0

XK : thermal conductivity of each layer (BTU/HR-FT-°F)
if air layer then XK = 0.0

D : density of each layer (LB/FT^3)
if air layer then D = 0.0

SH : specific heat of each layer BTU/Lb-°F)
if air layer then SH = 0.0

RES : thermal resistance of each AIR layer (HR-FT^2-°F/BTU)
if non-air layer then RES = 0.0

DEFC : alpha-numeric description of each layer, maximum of 30 characters

OUTPUT

$\left.\begin{array}{l}\text{RFX}\\ \text{RFY}\\ \text{RFZ}\end{array}\right\}$: Response factors series

NRFT : number of response factor terms

R1 : common ratio

CALCULATION SEQUENCE

```
1. Read number of material layers M
2. For i = 1, M
     a) read and echo
          1) thickness $XL_i$
          2) conductivity $XK_i$
          3) density $D_i$
          4) specific heat $SH_i$
```

        5) resistance RES_i

        6) alpha-numeric description $DEFC_i$

    b) for each non-air layer, calculate:

        1) its resistance $R_i = XL_i/XK_i$

        2) $BETA_i = XL_i*SQRT(D_i*SH_i/XK_i)$

    c) for each air layer:

        1) set $R_i = RES_i$

        2) $BETA_i = 0.0$

3. Call subroutine RESIDU to compute the thermal conductance RK0 and the derivatives of matrix elements A, B, and D (variables RK1, RM1, and RM4) for the condition where P, of B(P), approaches 0.0

4. Call subroutine ROOTS to calculate the roots of B(P) = 0.0 and the number of roots NRT

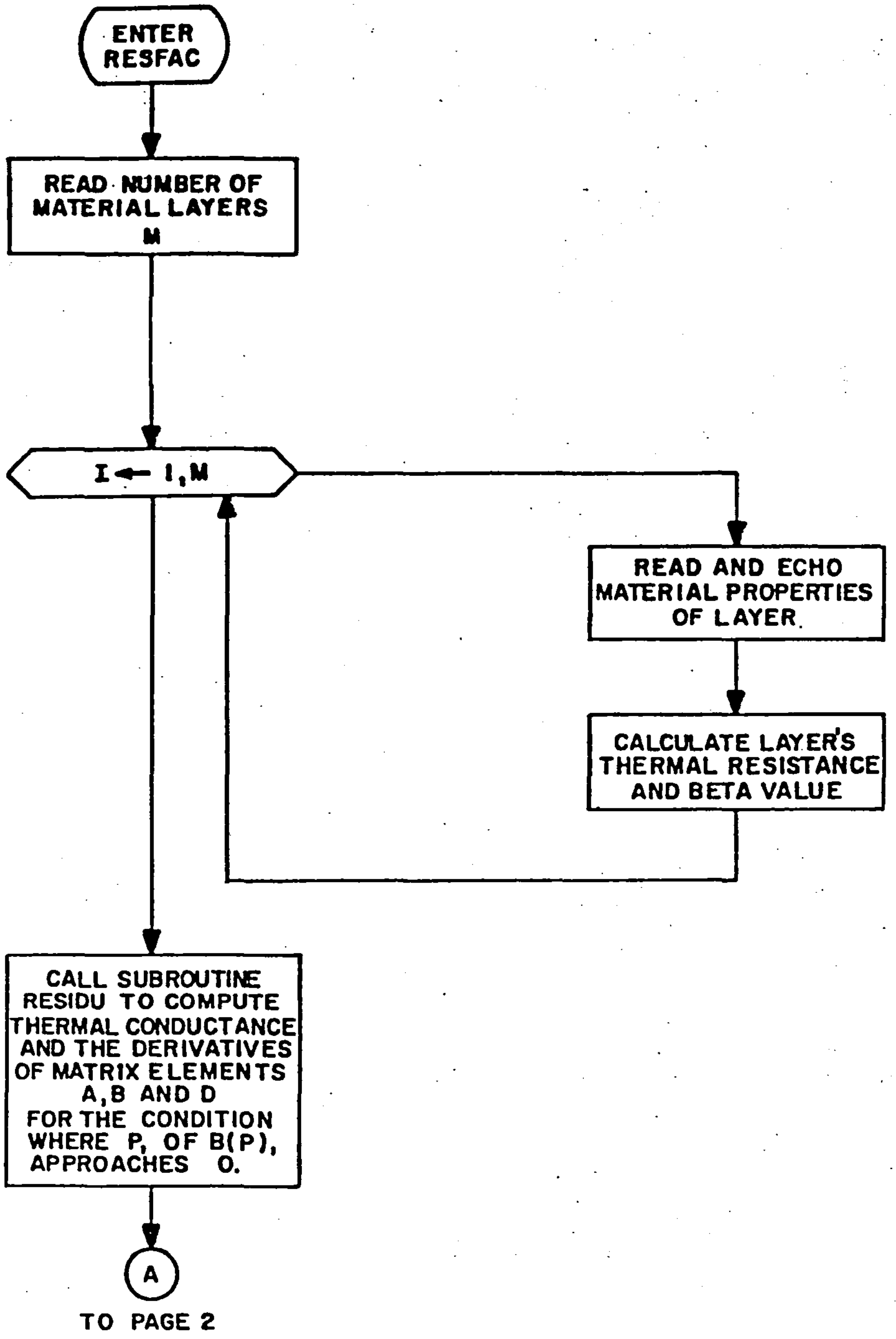
5. For i = 1, NRT

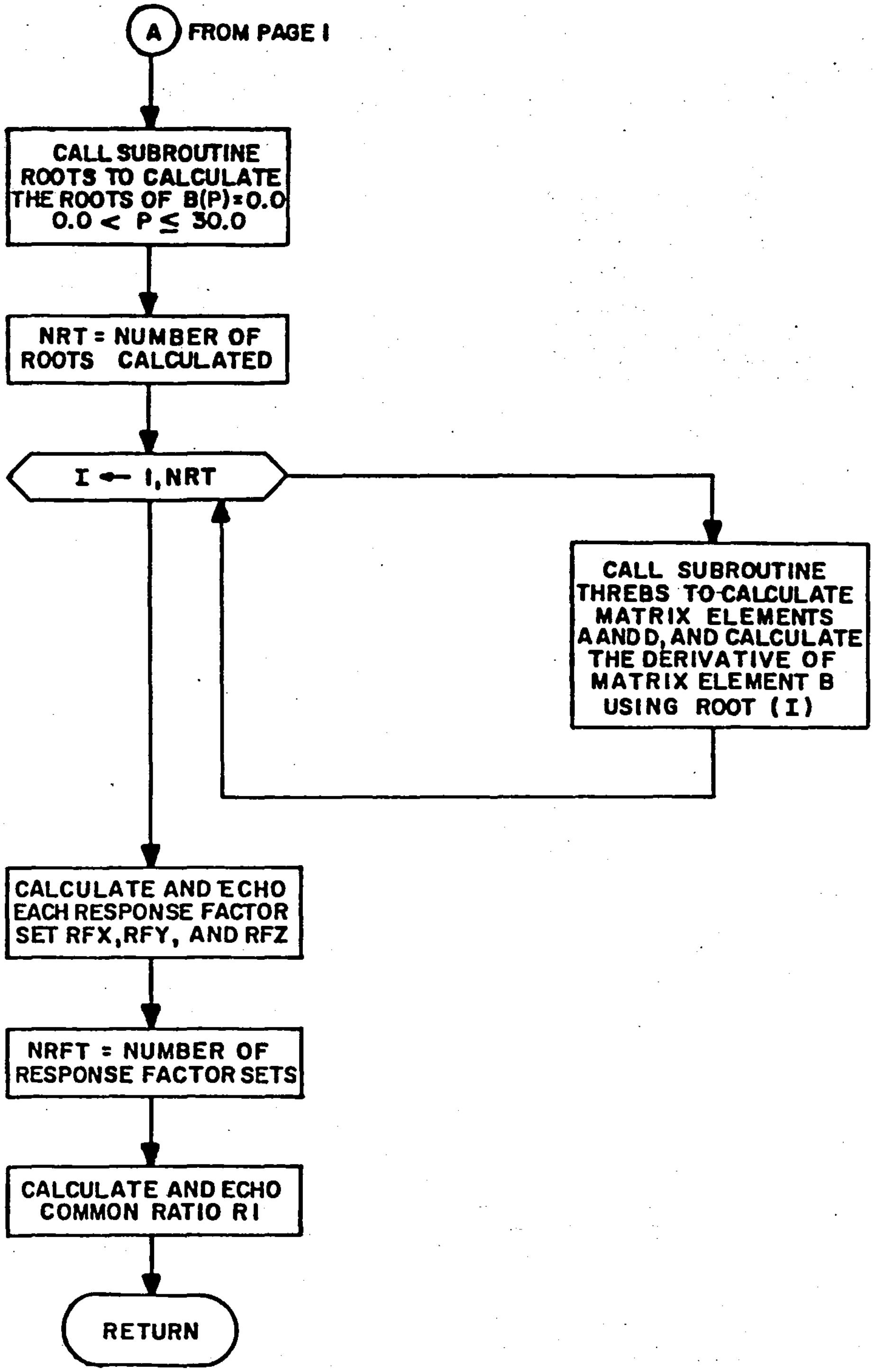
```
        1)  call subroutine THREBS to calculate matrix
            elements A, D and the derivative of matrix
            element B (variables B1, B2, and BP3)

        2)  set RKK(i,2) = 1.0/BP3/ROOT(i)^2
                RKK(i,1) = RKK(i,2)*B1
                RKK(i,3) = RKK(i,2)*B2
```

6. For i = 1,100:

```
    a)  for j = 1, NRT:
        if  (ROOT(j) > 30.0) go to 6b
        BETAY = EXP(-ROOT(j)*i)
        A = A + RKK(j,1)*BETAY
        B = B + RKK(j,2)*BETAY
        C = C + RKK(j,3)*BETAY

    b)  A = A + RK1 + (RM4*RK0) + (i*RK0)
        B = B + (i*RK0) + RK1
        C = C + RK1 + (RM1*RK0) + (i*RK0)

    c)  if (i = 1):
        AA = A
        BB = B
        CC = C
```

```
   d)  if (i = 2):
       AA = A-2.0*X
       BB = B-2.0*Y
       CC = C-2.0*Z

   e)  if ( > 2):
       AA = A-2.0*X + FA
       BB = B-2.0*Y + FB
       CC = C-2.0*Z + FC

   f)  set and echo response factors:
       RFX(i) = AA
       RFY(i) = BB
       RFZ(i) = CC

   g)  if (i < 3) go to 6k

   h)  if (ABS(XX/FAA-AA/XX) > 1.0E-5) go to 6k

   i)  if (ABS(YY/FBB-BB/YY) > 1.0E-5) go to 6k

   j)  if (ABS(ZZ/FCC-CC/ZZ) < 1.0E-5) go to 7

   k)  if (ABS(AA) < 1.0E-8) go to 7

   l)  set:
       FA = X
       FB = Y
       FC = Z
       FAA = XX
       FBB = YY
       FCC = ZZ
       XX = AA
       YY = BB
       ZZ = CC
       X = A
       Y = B
       Z = C
       go to 6

7.  Set and echo common ratio:
    R1 = EXP(-ROOT(1))

8.  End
```


ENTER RESFAC
READ NUMBER OF MATERIAL LAYERS M
I ← 1,M
READ AND ECHO MATERIAL PROPERTIES OF LAYER.
CALCULATE LAYER'S THERMAL RESISTANCE AND BETA VALUE
CALL SUBROUTINE RESIDU TO COMPUTE THERMAL CONDUCTANCE AND THE DERIVATIVES OF MATRIX ELEMENTS A,B AND D FOR THE CONDITION WHERE P, OF B(P), APPROACHES 0.
A
TO PAGE 2


A
FROM PAGE 1
CALL SUBROUTINE ROOTS TO CALCULATE THE ROOTS OF B(P)=0.0 0.0 < P ≤ 30.0
NRT = NUMBER OF ROOTS CALCULATED
I ← 1,NRT
CALL SUBROUTINE THREBS TO CALCULATE MATRIX ELEMENTS A AND D, AND CALCULATE THE DERIVATIVE OF MATRIX ELEMENT B USING ROOT (I)
CALCULATE AND ECHO EACH RESPONSE FACTOR SET RFX, RFY, AND RFZ
NRFT = NUMBER OF RESPONSE FACTOR SETS
CALCULATE AND ECHO COMMON RATIO RI
RETURN

RESIDU

A subroutine which determines thermal conductance and the derivatives of matrix elements A, B, and D for the condition where the root W approaches 0.0.

INPUT

RR : each layers resistance

BETA : each layers BETA

M : number of layers

OUTPUT

RK0 : thermal conductance

RK1 : derivative of matrix element B

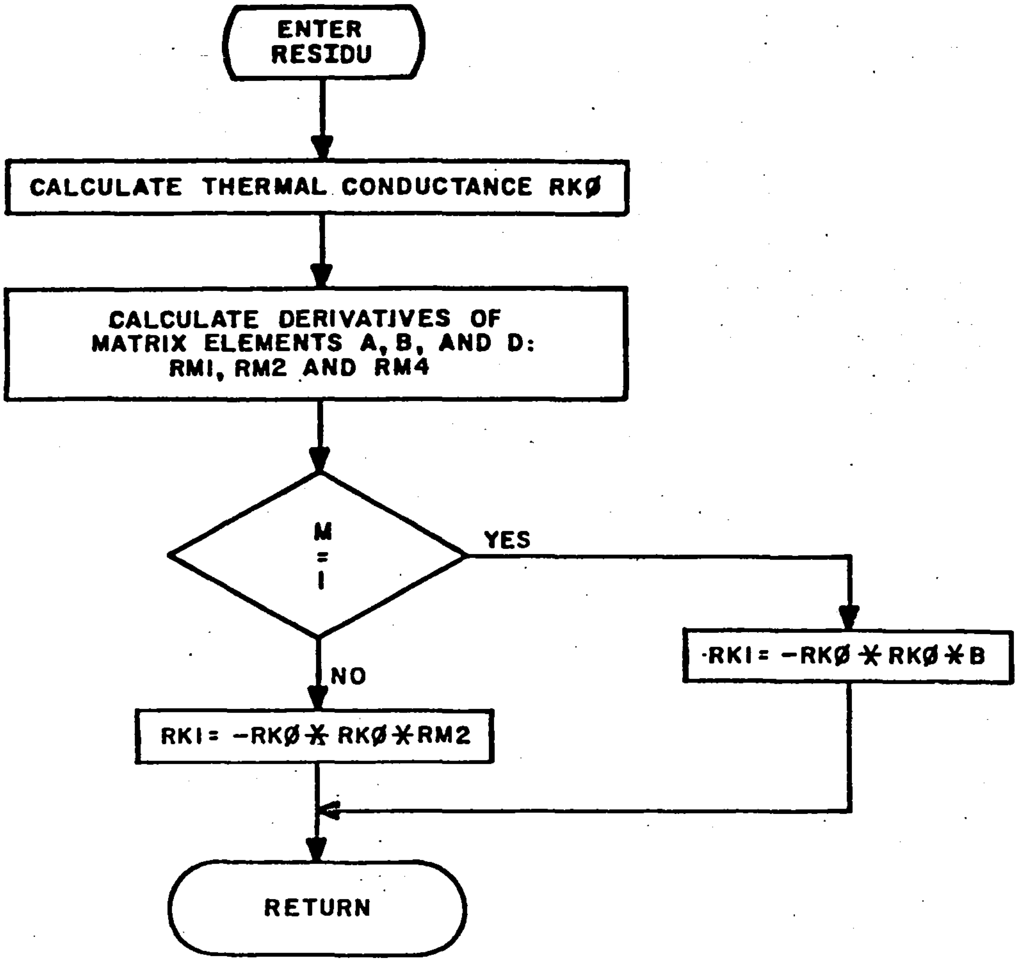
RM1 : derivative of matrix element A

RM4 : derivative of matrix element D

CALCULATION SEQUENCE

```
1.  Sum all layer resistances and compute thermal
    conductance RK0 = 1.0/R total

2.  For i = 1,M

     a)  P = BETA(1)*BETA(1)
         R = RR(1)

     b)  if (i > 1):
         A = 1.0
         B = R
         C = 0.0
         D = 1.0
         go to 2e

     c)  if (i = 1):
         A = P/2.0
         B = R*P/6.0
         C = P/R
         D = P/2.0

     d)  if (M = 1):
         RM1 = A
         RK1 = -RK0*RK0*B
         RM4 = D
         go to 4
```

```
    e)  for j = 2, M

        1)  P = BETA(j)*BETA(j)
            R = RR(j)
            if i = j go to 2e2
            AA = 1.0
            BB = R
            CC = 0.0
            DD = 1.0
            go to 2f

        2)  AA = P/2.0
            BB = P*R/6.0
            CC = P/R
            DD = P/2.0

    f)  F1 = A*AA+B*CC
        F2 = A*BB+B*DD
        F3 = C*AA+D*CC
        F4 = C*BB+D*DD
        A = F1
        B = F2
        C = F3
        D = F4
        go to 2e

    g)  RM1 = RM1+F1
        RM2 = RM2+F2
        RM4 = RM4+F4
        go to 2

3.  RK1 = -RK0*RK0*RM2

4.  End
```


ENTER RESIDU
CALCULATE THERMAL CONDUCTANCE RK0
CALCULATE DERIVATIVES OF MATRIX ELEMENTS A, B, AND D: RM1, RM2 AND RM4
M = 1
YES
NO
RK1 = −RK0 * RK0 * B
RK1 = −RK0 * RK0 * RM2
RETURN

ROOTS

A subroutine which determines the roots of matrix element B(P) = 0.0

INPUT

BTA : each layers BETA

RS : each layers resistance

N : number of material layers

OUTPUT

ROT : root array

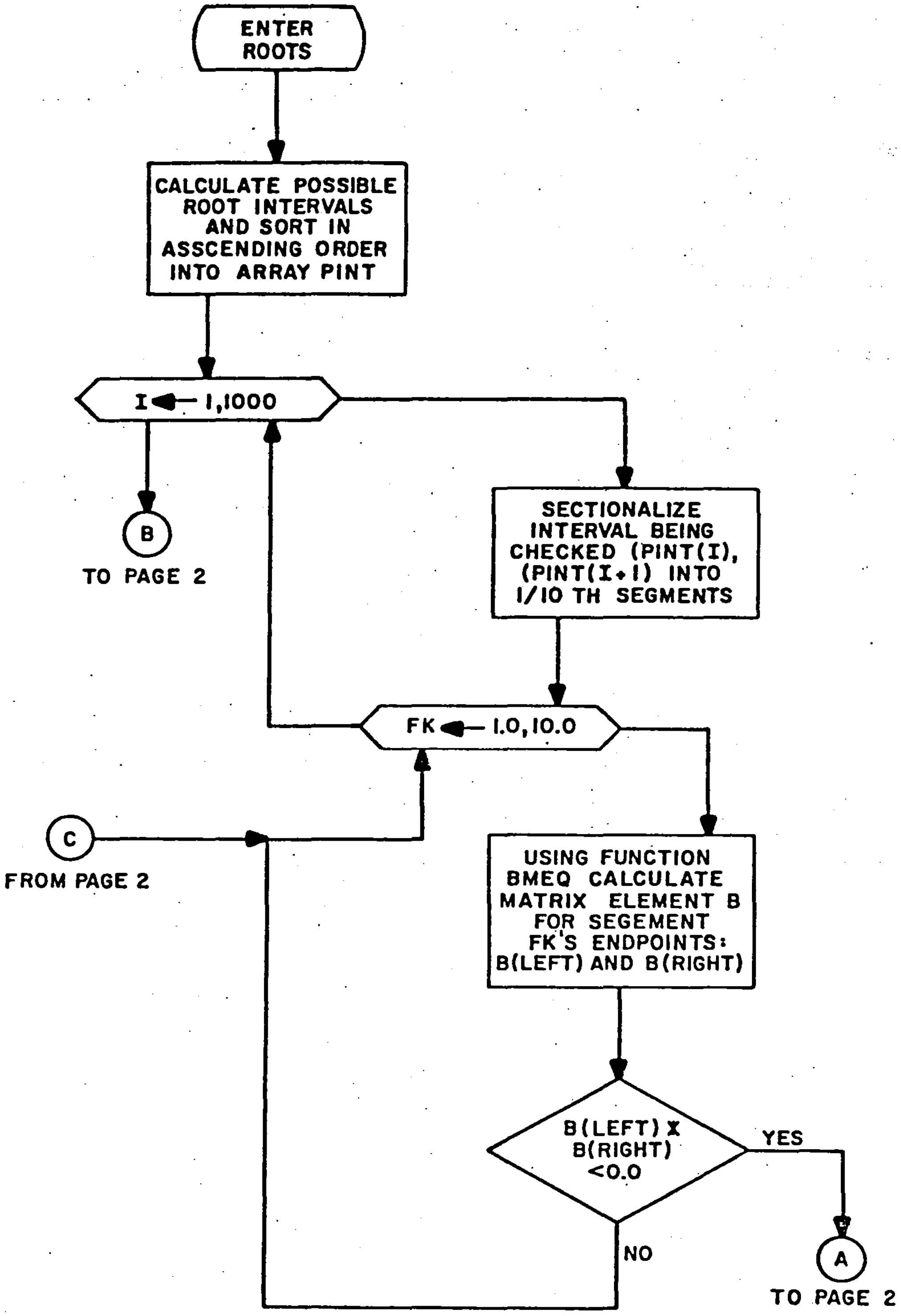
L : number of roots

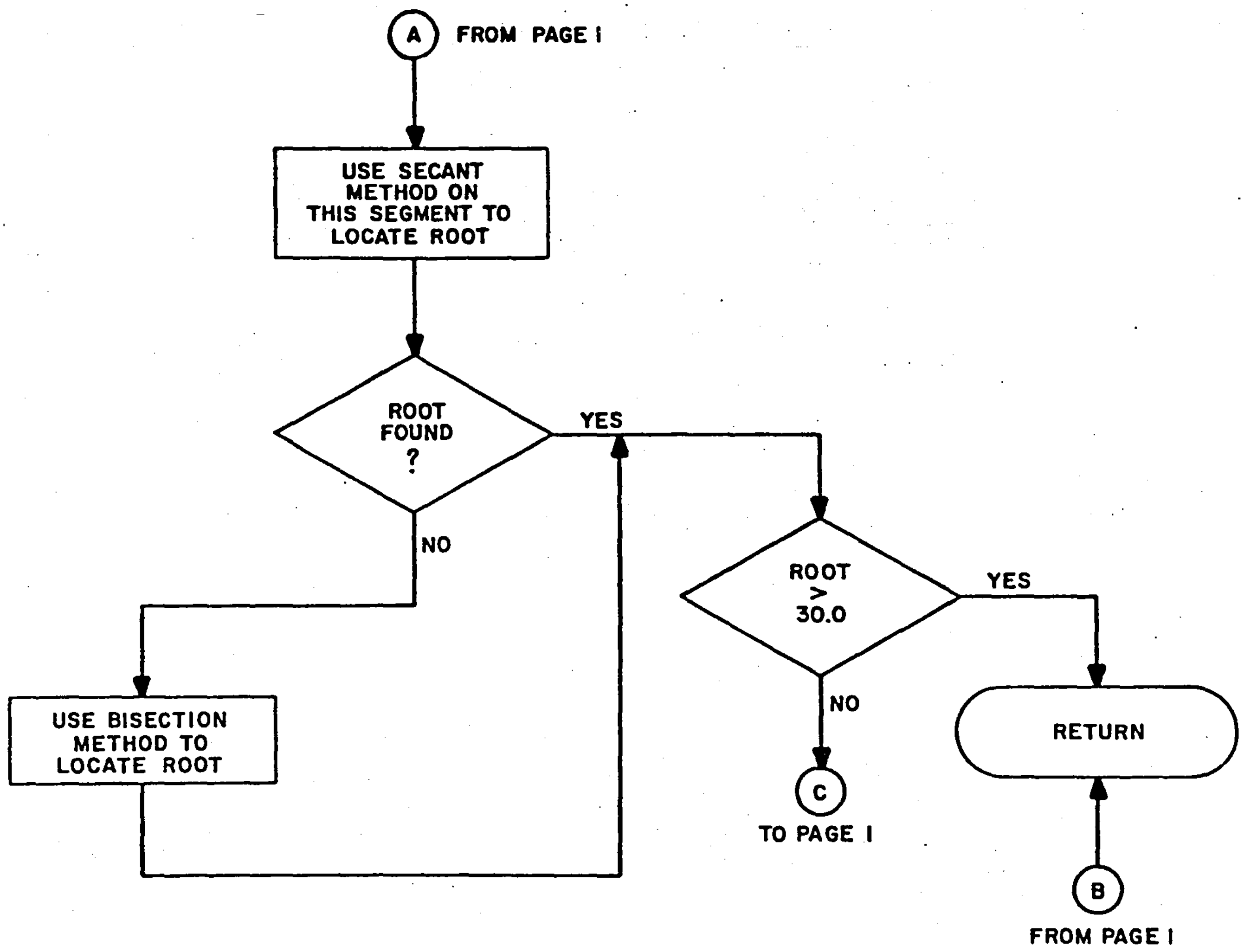
CALCULATION SEQUENCE

1. Set PI = 3.141592654
2. For i = 1, N:
    - a) if BTA(i) >0.0 then calculate the zero points of this layers sine and cosine components:
        - 1) set R = 1.0
        - 2)
          ```
          for j = 1, 99, 2
          BM(j, i) = (R - 0.5)*PI/BTA(i)
          BM(j + 1, i) = R*PI/BTA(i)
          R = R + 1.0
          ```
3. For values >0.0, sort the BM arrays' values into the possible root interval array PINT in ascending order.
4. For i = 1,1000 determine the roots such that no root > 30.0:
    - a) set G1 = PINT(i) and calculate F1(G1) using function BMEQ
        - 1) if (ABS(F1) < 1.0E-8) go to 4e
    - b) set G2 = PINT(i + 1) and calculate F2(G2) using function BMEQ
        - 1) if (ABS(F2) < 1.0E-8) go to 4e
    - c) set the interval length D = G2 - G1

```
d)  sectionalize each interval into 1/10th segments
    and check each segment to determine if there is
    a root there:

     1)  FK = FK + 1.0
         if (FK > 10.0) go to 4

     2)  set DD = PINT(i) + (D*FK/10.0) and
         calculate FD(DD) using function BMEQ

     3)  SM = FD*F1
         if (SM <= 0.0) go to 4d5

     4)  G1 = DD
         F1 = FD
         go to 4d1

     5)  GX = G1
         FX = F1
         G2 = DD
         F2 = FD
         go to 4f

 e)  L = L + 1

      1)  if (ABS(F1) <=1.0E-8)ROT(L) = G1**2

      2)  if (ABS(F2) <= 1.0E-8 and ROT(L) = 0.0)
          ROT(L) = G2**2

      3)  if (ROT(L) > 30.0) go to 5
          else go to 4

f)  use secant method to find root

     1)  G3 = G2 - (F2*(G2-G1)/(F2-F1))

     2)  if (G3 < GX or G3 > DD) then secant method fails
         so use bisection method to locate root - go to g

     3)  if (G3 >GX and G3<DD) then calculate FRT(G3)
         using function BMEQ

     4)  if (ABS(FRT) <= 1.0E-8) go to 4j

     5)  G1 = G2
         G2 = G3
         F1 = F2
         F2 = FRT
         go to 4f1
```

```
    g)   set G4 = 0.5*(G1 + G2) and calculate
         F4(G4) using function BMEQ

    h)   if (ABS(F4) <= 1.0E-8) G3 = G4

         1)  go to 4j

    i)   F5 = F1*F4
         if (F5 < 0.0) G2 = G4
         if (F5 > 0.0) G1 = G4 and F1 = F4
         go to 4g

    j)   L = L + 1
         ROT(L) = G3**2
         if (ROT(L) > 30.0) go to 5
         G1 = DD
         F1 = FD
         go to 4d1

5)  End
```


ENTER ROOTS
CALCULATE POSSIBLE ROOT INTERVALS AND SORT IN ASSCENDING ORDER INTO ARRAY PINT
I ← 1,1000
B
TO PAGE 2
SECTIONALIZE INTERVAL BEING CHECKED (PINT(I), (PINT(I+1) INTO 1/10 TH SEGMENTS
FK ← 1.0,10.0
C
FROM PAGE 2
USING FUNCTION BMEQ CALCULATE MATRIX ELEMENT B FOR SEGEMENT FK'S ENDPOINTS: B(LEFT) AND B(RIGHT)
B(LEFT) X B(RIGHT) <0.0
YES
NO
A
TO PAGE 2


A
FROM PAGE I
USE SECANT METHOD ON THIS SEGMENT TO LOCATE ROOT
ROOT FOUND ?
YES
NO
USE BISECTION METHOD TO LOCATE ROOT
ROOT > 30.0
YES
NO
RETURN
C
TO PAGE I
B
FROM PAGE I

THREBS

A subroutine which determines matrix elements A, D,and the derivative of matrix element B at the root W.

INPUT

RR : each layers resistance

BETA : each layers BETA

W : root

M : number of layers

OUTPUT

B1 : matrix element A

B2 : matrix element D

BP3 : derivative of matrix element B

CALCUALTION SEQUENCE

```
1.  For i = 1, M

    a)  P = SQRT(W)*BETA(i)
        R = RR(i)
        ALPHA = BETA(i)
        SQ = SQRT(W)

    b)  if  (P = 0.0):
        AL(i) = 1.0
        BL(i) = R
        CL(i) = 0.0
        DAL(i) = 0.0
        DBL(i) = 0.0
        DCL(i) = 0.0

    c)  if  (P > 0.0):
        AL(i) = COS(P)
        BL(i) = R/P*SIN(P)
        CL(i) = -P/R*SIN(P)
        DAL(i) = ALPHA*SIN(ALPHA*SQ)/2.0/SQ
        DBL(i) = -R*COS(ALPHA*SQ)/2.0/W+R*SIN(ALPHA*SQ)/
                 ALPHA/2.0/SQ/SQ/SQ
        DCL(i) = ALPHA*ALPHA*COS(ALPHA*SQ)/2.0/R+SIN
                 (ALPHA*SQ)/2.0/SQ*ALPHA/R

    d)  go to 1
```

```
2.  If (M = 1):
    BP3 = DBL(1)
    B1 = DAL(1)
    B2 = AL(1)
    go to 7

3.  For i = 1, M

     a)  DA = 1.0
         DB = 0.0
         DTA = 0.0
         DTB = 0.0

     b)  for j = 1, M

          1)  if (i = j):
              DTA = (DA*DAL(j)) + (DB*DCL(j))
              DTB = (DA*DBL(j)) + (DB*DAL(j))
              go to 3b3

          2)  DTA = (DA*AL(j)) + (DB*CL(j))
              DTB = (DA*BL(j)) + (DB*AL(j))

          3)  DA = DTA
              DB = DTB
              go to 3b

     c)  BP3 = BP3 + DTB

     d)  go to 3

4.  A = AL(1)
    B = BL(1)
    C = CL(1)
    D = AL(1)

5.  For i = 2, M

     a)  TA = (A*AL(i)) + (B*CL(i))
         TB = (A*BL(i)) + (B*AL(i))
         TC = (C*AL(i)) + (D*CL(i))
         TD = (C*BL(i)) + (D*AL(i))
          A = TA
          B = TB
          C = TC
          D = TD

     b)  go to 5

6.  B1 = TD
    B2 = TA

7.  End
```

BMEQ

A function which determines the matrix element B at the root guess W.

INPUT

W : root quess

BETA : each layers BETA

R : each layers resistance

M : number of layers

OUTPUT

BMEQ : matrix element B

CALCULATION SEQUENCE

```
1.  For i = 1, M

      a)  if (BETA(i) = 0.0):
          AL(i) = 1.0
          BL(i) = R(i)
          CL(i) = 0.0
          to to 1

      b)  P = W*BETA(i)
          AL(i) = COS(P)
          SN = SIN(P)
          BL(i) = R(i)*SN/P
          go to 1

2.  If (M = 1):
    TB = BL(1)
    go to 5

3.  A = AL(1)
    B = BL(1)

4.  For i = 2, M

          TA = (A*AL(i)) + (B*CL(i))
          TB = (A*BL(i)) + (B*AL(i))
          A = TA
          B = TB


5.  BMEQ = TB

6.  End
```

1. Report No. NASA CR-165982	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle NECAP 4.1 - NASA'S ENERGY COST ANALYSIS PROGRAM - THERMAL RESPONSE FACTOR ROUTINE		5. Report Date August 1982 6. Performing Organization Code
7. Author(s) Michael R. Wiese		8. Performing Organization Report No.
9. Performing Organization Name and Address Computer Sciences Corporation Hampton, Virginia		10. Work Unit No. 11. Contract or Grant No. NAS1-16078
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Contractor Report 1981-1982 14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitors: John E. Hogge and Ronald N. Jensen

16. Abstract

The RESFAC2 version of the Thermal Response Factor Program (RESFAC) is the result of numerous modifications and additions to the original RESFAC. These modifications and additions have significantly reduced the program's computational time requirement. As a result of this work, the program is more efficient and its code is both readable and understandable.

This report describes what a thermal response factor is and provides calculation sequences and flowcharts for RESFAC2.

RESFAC is used by NASA'S ENERGY COST ANALYSIS PROGRAM (NECAP) to calculate hourly heat transfer coefficients (thermal response factors) for each unique delayed surface. NECAP uses these response factors to compute each spaces' hourly heat gain/loss.

17. Key Words (Suggested by Author(s)) ENERGY ENERGY ANALYSIS HEAT TRANSFER		18. Distribution Statement Subject Category 44 Unclassified-Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 26	22. Price A03

End of Document