```
INTEGER JN(40.5), NN(40), JB(20), IFLOW(40), LP(8,20), JC(50)
REAL D(50), L(50), A(50.51), QJ(20), E(50), KP(50), V(2), Q(50),
      EXPP(50), AR(50), ARL(50)
```

Data type

```
30    READ(5,110,END=99) NP, NJ, NL, MAX, NUNIT, ERR, VIS, DELQ1
110   FORMAT(515.3F10.5)
```

Read in data

```
NN – number of nodes
JN – number of pipes at node
JB – unknown
INFLOW – consumption flow rate
LP – unknown
JC – to do with solution of linear equations

D – pipe diameter
L – pipe length
A – flow
QJ – unknown
E – initial absolute roughness
      converted to relative roughness
V – velocity
Q – flow in pipe
KP – Kp value for pipe
EXPP – n exponent in equation h_f = Kp * Q^n
AR – pipe x-sectional area
ARL – coefficient in equation Kp = a * (L / (2 * g * D * A^2))
```

Used in equations 2-15 to 2-18 Ref 1

Variable definitions

```
# NP - NO. OF PIPES.
# NJ - NO. OF JUNCTIONS,
# NL - NO. OF LOOPS,
# MAX - NO. OF ITERATIONS ALLOWED,
# ERR – allowed error in calculated flows before accepting results
# VIS – kinematic viscosity
# DELQ1 – deviation used to calculate Q1 and Q2 from average Flow
# IF NUNIT=0 D AND E IN INCHES AND L IN FEET,
# IF NUNIT=1 D AND E IN FEET AND L IN FEET,
# IF NUNIT=2 D AND E IN METERS AND L IN METERS,
# IF NUNIT=3 D AND E IN CM AND L IN METERS.
```

The 100 series line numbers reference formatting only and are not addresses here.

```
100   FORMAT(1615)
      NPP = NP + 1
      NJI = NJ - 1
      READ(5,101) (D(I), I=1, NP)
      READ(5,101) (L(I), I=1, NP)
      READ(5,101) (E(I), I=1, NP)
101   FORMAT(8FI0.5)
      DO 48 I=1, NP
48    E(I) = E(I) / D(I)
      IF(NUNIT-1) 40, 41, 42
40    WRITE(6.102) D(I), I=1, NP)
102   FORMAT('PIPE DIAMETERS (INCHES)'./,(1H, 16F8.1))
      DO 43 I=1, NP
43    D(I) = D(I)/12
      GO TO 44
41    WRITE(6.112) (D(I), I=1, NP)
112   FORMAT('PIPE DIAMETERS (FEET)'./,(1H, 16F8.3)
44    WRITE (16.103) L(I), I=1, NP)
103   FORMAT('LENGTH OF PIPE (FEET)'./,(1H, 16F8.0)
```

Convert absolute roughness to relative roughness

Based on units selected, convert dia and length to feet or meter

```
      G2 = 64.4
      GO TO 50
42    IF (NUNIT .EQ. 2) GO TO 45
      DO 46 I=1, NP
46    D(I) = .01 * D(I)
45    WRITE(6.113) (D(I), I=1, NP)
113   FORMAT('PIPE DIAMTERS (METERS)',/,(1H.16.F8.4))
      WRITE(6.114) (L(I), I=1, NP)
114   FORMAT('LENGTH OF PIPE (METERS)',/,(1H, 16F8.0)
      G2=19.62
      WRITE(6.115) (E(I), I=1, NP)
115 FORMAT('RELATIVE ROUGHNESS OF PIPES',/,(1H, 16F8.6)
```

NOTE
Undefined

Length and dia convertion cont'd

```
# INFLOW - IF 0 NO INFLOW,
      IF 1 THEN NEXT CARD GIVES MAGNITUDE IN GPM
      IF 2 NEXT CARD GIVES MAGNITUDE IN CFS
      IF 3 NEXT CARD GIVES MAGNITUDE IN CMS.
# NNJ - NO. OF PIPES AT JUNCTIONS POSITIVE FOR INFLOW
  NEGATIVE FOR OUTFLOW.
# JN - THE NUMBER OF PIPES AT JUNCTION,
      IF FLOW ENTERS MINUS
      IF FLOW LEAVES THE PIPE NUMBER IS POSITIVE.
```

Variables for nodes

```
      DO 70 I=1, NP
      AR(I) = .78539392 * D(I)**2
70    ARL(I) = L(I) / (G2*D(I) * AR(I)**2)
```

Pipe x-sect area
Coefficient for Kp - equation 2-28 Ref 1

```
      II = I
      DO 1 I=1, NJ
      READ(5,100) IFLOW(I), NNJ, (JN(I,J),J=I,NNJ)
      NN(l)=NNJ
      IF(IFLOW(I)-1) 1.2,3
2     READ(5,101) QJ(II)
      QJ(II) = QJ(II)/449.
      JB(II)=1
      GO TO 4
3     READ(5, 101) QJ(II)
      BJ(II) = 1
4     II = II + 1
1     CONTINUE
```

Determine the coefficient for the variables in the node equation same as: Calc_Network.py.node_matrix

```
# NUMBER OF PIPES IN EACH LOOP (SIGN INCLUDED)
      DO 35 I=1, NL
      READ(5,100) NNJ, (LP(J,I), J=1 ,NNJ)
35    LP(8,I)=NNJ
      DO 5 I=1, NP
      IF(NUNIT .GT. 1) GO TO 66
      KP(I)=.0009517 * L(I) / D(I)**4.87
      GO TO 5
66    KP(I) =.00212 * L(I) / D(I)**4.87
5     CONTINUE
      ELOG = 9.35 * ALOG10(2.71828183)
      SUM=100
      NCT=O
20    II = 1
      D0 6 I=1, NJI
      D0 7 J=1, NP
7     A(I,J) = O
      NNJ = NN(I)
```

Calculate initial Kp based on Chw = 100 – equation 2-22 Ref 1

Set constants to be used further on in iteration loop.

MAIN LOOP

Calculation of the energy equations for closed loops and pseudo loops same as Calc_Network.py loop_matrix and pseudo_matrix

Calculation of the energy equations cont'd

```
      DO 8 J=1, NNJ
      IJ = JN(I,J)
      IF(IJ .GT. 0) GO TO 9
      IIJ = ABS(IJ)
      A(I,IIJ) = -1.
      GO TO 8
9     A(I,IJ) = 1
8     CONTINUE
      IF(IFLOW(I).EQ.O)GO TO 10
      A(I,NPP) = QJ(II)
      II = II + 1
      GO TO 6
10    A(I,NPP) = O.
6     CONTINUE
      DO 11 I=NJ, NP
      DO 22 J=I, NP
22    A(I,J)=O.
      II = I-NJI
      NNJ = LP(8,II)
      DO 12 J=I, NNJ
      IJ = LP(J,II)
      IIJ = ABS(IJ)
      IF(IJ .LT. 0) GO TO 13
      A(I,IIJ )=KP(IIJ)
      GO TO 12
13    A(I,IIJ) = -KP(IIJ)
12    CONTINUE
11    A(I,NPP) = O.
      V(I) = 4.
```

MAIN LOOP uses the new Kp and n exponent calculated in the lower iteration loop with the code above to rebuild the energy equations. This will continue until the allowed error is reached or the number of allowed iterations is exceeded.

```
# SYSTEM SUBROUTINE FROM UNIVAC MATH PACK TO SOLVE LINEAR SYSTEM OF EQ.
      CALL GJR(A, 51, 50, NP, NPP, $98, JC, V)
```

Replaced with numpy solver

```
      IF (NCT .GT. 0) SUM=O.
      DO 51 I=1, NP
      BB = A(I,NPP)
      IF(NCT) 60,60,61
60    QM = BB
      GO TO 62
61    QM = .5 * (Q(I) + BB)
      SUM = SUM + ABS(Q(I)-BB)
62    Q(I) = QM
      DELQ = QM * DELQ1
      QM = ABS(QM)
      VI = (QM - DELQ) / AR(I)
      IF(VI .LT. .00l) V1=.002
      V2 = (QM + DELQ) / AR(I)
      VE = QM / AR(I)
      REI = V1 * D(I) / VIS
      RE2 = V2 * D(I) / VIS
      IF(RE2 .GT. 2.1E3) GO TO 53
      F1 = 64./RE1
      F2 = 64./RE2
      EXPP(I) = 1.
      KP(I) = F2 * (Lgth+Le) / Dia
      KP(I) = 64.4 * VIS * ARL(I) / D(I)
      GO TO 51
53    MM = O
      F = 1 / (1.14 - 2 * ALOG10(E(I)))**2
```

Resetting setting of SUM

1st iteration QM = initially calculated Q, all others use last 2 calculated Qs for QM. BB is previous Q

Iteration loop to calculate the Kp and n exponent for the energy equations based on previous values of Q. For first iteration use Q1 and DELQ. Afterwards use the previous Q and newly calculated Q. Reassign the new Kp and n exponents in the energy equations.

Laminar Flow

NOTE; the original laminar KP equation was replaced with the equation 3-15 from Crane C-410 paper.

```
      PAR = VE * SQRT(.125 * F) * D(I) * E(I) / VIS
      IF(PAR .GT. 65.) GO TO 54
      RE = RE1      120
57    MCT = 0
52    FS = SQRT(F)
      FZ = .5 / (F*FS)
      ARG = E(I) + 9.35 / (RE * FS)
      FF = 1./FS - 1.14 + 2.*ALOG10(ARG)
      DF = FZ + ELOG *FZ / (ARG * RE)
      DIF = FF / DF
      F = F + DIF
      MCT = MCT + 1
      IF(ABS(DIF) .GT. .00001 .AND. MCT .LT. 15) GO TO 52
      IF(MM .EO. I) GO TO 55
      MM = I
      RE = RE2
      F1 = F
      GO TO 57
55    F2 = F
      BE = (ALOG(F1) - ALOG(F2)) / (ALOG(QM + DELQ) - ALOG(QM - DELQ))
      AE = F1 * (QM - DELQ)**BE
      EP = 1 - BI:
      EXPP(I) = EP + 1
      KP(I) = AE * ARL(I) * QM**EP
      GO TO 51
54    KP(I) = F * ARL(I) * QM**2
      EXPP(I) = 2
51    CONTINUE
17    NCT = NCT + 1
      IF(SUM .GT. ERR AND NCT .LT. MAX) GO TO 20

      IF(NCT . EQ. MAX) WRITE(6,108) NCT,SUM
108   FORMAT('DID NOT CONVERGE IN 15 ITERATIONS SUM OF DIFFERENCES')

      IF(NUNIT ,LT. 2) GO TO 63
      WRITE(6.127) (Q(I), I=1, NP)
127   FORMAT('FLOWRATE IN PIPES IN CMS',/,(1H, 131.10.4))
      DO 64 I=1, NP
64    KP(I) = KP(I) * ABS(Q(I))
      WRITE(6, 139) (KP(I), I=1, NP)
      GO TO 30
63    WRITE(6,107) (Q(I), I=1, NP)
107   FORMAT('O FLOW RATES IN PIPES IN CFS'./.(IH.13F10,3))
      DO 21 I=1, NP
      KP(I) = KP(I) * ABS(Q(I))
21    Q(I) = 449. * Q(I)
      WRITE(6,138) (KP(I), I=1, NP)
138   FORMAT(' HEAD LOSSES IN PIPES',/,(IH ,13FI0.3))
      WRITE(6,105) (Q(I), I=1, NP)
105   FORMAT('FLOW RATES (GPM)',/,(IH ,13FI0.1))
      GO TO 30

98    WRITE(6,106) JC(1),V
106   FORMAT('OVERFLOW OCCURRED -- CHECK SPECIFICATIONS FOR REDUNDANT EQ.
      RESULTING IN SINGULAR MATRIX',15,2F8.2)
      GO TO 30
99    STOP
      END
```

PAR is proportional to velocity high par allows for higher pipe velocity. Recommend PAR <= 120

Loop is equivalent to Calc_Network Iterate_Flow

Equation 2-25 to 2-28 in Ref 1

MAIN LOOP

Beginning of program

Ouput Kp and Q

Beginning of program

Beginning of program

REFERENCE 1 ‘Steady Flow Analysis of Pipe Networks An Instructional Manual.pdf’