```
0101          V= (M2+A8*D5+B8*D6+D8*D7+E8*T)/X9
0102          A8= R*U+S
0103          B8= R*V+T
0104          AX7= A7*U+B7
0105          BX7= A7*V+D7
0106          A7= AX7
0107          B7= BX7
0108          AX6= A6*U+B6
0109          BX6= A6*V+D6
0110          A6= AX6
0111          B6= BX6
0112          AX5= A5*U+B5
0113          BX5= A5*V+D5
0114          A5= AX5
0115          B5= BX5
0116          A9= -G0*SS+H0*ZC
0117          B9= G0*ZS+H0*SC
0118          D9= H0*ZS-G0*SC
0119          E9= H0*SS+G0*ZC
0120          X10= G0-H0*U-(A9*A5+B9*A6+D9*A7+E9*A8)
0121          C10= (A9*B5+B9*B6+D9*B7+E9*B8+H0*V)/X10
0122          c9= U*C10 + V
0123          C8= R*C9 + S*C10 + T
0124          C7= O*C8 + P*C9 +Q
0125          C6= L*C7 + M*C8 + N
0126          C5 = H*C6 + I*C7 + J*C8 + K
0127          c4= 0.0
0128          C3= D*C5 + E*C7 + F*C8 + G
0129          C2= A*C3 + B*C7 + C
0130          C1= 0.0
0131      C
0132          RETURN
0133          END
```

```
0001          SUBROUTINE STRMAX(STM,STL)
0002          IMPLICIT REAL*16 (A-Z)
0003          INTEGER*4 I,IMAX
0004      C
0005          COMMON/PARA/QK,FP,KS,NM,QON,QOK,W1,H1,L,L0,LM,LM2
0006          COMMON/CONST/C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,T1,T2,T3,T4
0007          COMMON/GREEK/AG,BG,AG2,BG2,E0,F0,G0,H0
0008          COMMON/INPUT/OD ,TD ,EM,GAMMA,SIGMA, STRLMT, SIGINT
0009          COMMON/STRM/DXL ,ST1 ,ST2,ST3
0010          ST1= 0.0
0011          ST2= 0.0
0012          ST3= 0.0
0013          WOLD=0.0
0014          SLEN=0.0
0015          L1= L*3.
0016          IMAX= NINT(L1/DXL) + 1
0017          DX= L1/FLOAT(IMAX)
0018          DX2= DX*DX
0019      C
0020          DO 1000 I=0,IMAX
0021          X= DX*FLOAT(I)
0022          AX= AG*X
0023          BX=BG*X
0024          SS= SIN(AX)*SINH(BX)
0025          ZC= COS(AX)*COSH(BX)
0026          LX=LM*X
0027          CLL= COSH(LX)
0028          x2= X*X
0029          WT=  T1 + T2*CLL - QON/2.*X2
0030          M= C2*(F0*ZC+E0*SS) + C3*(E0*ZC-F0*SS)
0031         &   +LM2*T2*CLL - QON
0032      C
0033          DW=WT-WOLD
0034          IF(X.NE.0.0)THEN
0035            DL=SQRT(DX2+DW*DW)
0036            SLEN=SLEN+DL
0037          ENDIF
0038          WOLD=WT
0039          ST= ABS(M)*EM*OD/2.
0040     1000 ST1=   MAX(ST1,ABS(ST))
0041      C
0042          WOLD=0.0
0043          DO 2000 I=0,IMAX
0044          X= DX*FLOAT(I)
0045          AX=AG*X
0046          BX=BG*X
0047          SS= SIN(AX)*SINH(BX)
0048          SC= SIN(AX)*COSH(BX)
0049          ZS= COS(AX)*SINH(BX)
0050          ZC= COS(AX)*COSH(BX)
```

```
0051          XL= L1-X
0052          LX= LM*XL
0053          XL2= XL*XL
0054          CLL= COSH(LX)
0055          WT=  T3 + T4*CLL + QON/2.*XL2
0056          M= C5*(E0*ZS-F0*SC)+C6*(E0*SS+F0*ZC)+C7*(E0*ZC-F0*SS)
0057         &   +C8*(F0*ZS+E0*SC)
0058         &   + LM2*T4*CLL + QON
0059      C
0060          DW=WT-WOLD
0061          IF(X.NE.0.0)THEN
0062            DL=SQRT(DX2+DW*DW)
0063            SLEN=SLEN+DL
0064          ENDIF
0065          WOLD=WT
0066          ST= ABS(M)*EM*OD/2.
0067   2000   ST2=   MAX(ST2,ABS(ST))
0068      C
0069          DO 3000 I=0,120,6
0070          X= FLOAT(I)
0071          AX=AG*X
0072          BX=BG*X
0073          EBX=EXP(-BX)
0074          CEX= COS(AX)*EBX
0075          SEX= SIN(AX)*EBX
      6   C
0077          M= (C9*E0-C10*F0)*CEX + (C9*F0+C10*E0)*SEX
0078      C
0079          ST= ABS(M)*EM*OD/2.
0080   3000   ST3=   MAX(ST3,ABS(ST))
0081          STL= EM*(SLEN/L1/2.0-1.0)
0082          ST1=ST1+ABS(STL+SIGINT)
0083          ST2=ST2+ABS(STL+SIGINT)
0084          ST3=ST3+ABS(STL+SIGINT)
0085          STM= MAX(ST1,ST2,ST3)
0086          RETURN
0087          END
```

PROGRAM SUPPORT

The computer program SUPPORT calculates the distance between supports required to minimize the stress in the pipeline during lowering. The program is based on the free deflection equations described in Appendix B. The input to the program consists of:

1. Outside diameter of the pipe, OD (inches)
2. Wall thickness of the pipe, t (inches)
3. Elastic modulus of the pipe material, E (psi)
4. Specific weight of the pipe material, y (lb/in^3)
5. Height difference between two adjacent supports, H (inches).
6. Specific gravity of the fluid in the pipe, SG (dimensionless)
7. Existing axial stress in the pipeline σ_L (psi)
8. Maximum allowable axial stress, σ_{max} (psi).

The SUPPORT program calculates the minimum and maximum distances between supports for the given height difference between adjacent supports. The axial stress induced by the lengthening of the pipeline is taken into account in the program and is added to the existing stress to determine the total axial'stress. The program iterates on the solution until the lengthening stress from two successive iterations is equal.

The program was designed to be run interactively. The output from the program is displayed on the terminal screen. An additional copy of the output may be written to file FOR001.DAT, if desired. The output consists of:

1. An echo of the input parameters
2. The minimum and maximum support spacings, or the support spacing for the minimum stress, if the calculated stress is never less than the allowable stress.

A computer listing of the program (VAX-11/780 version) is given below.

```
0001            PROGRAM SUPPORT
0002            IMPLICIT REAL*8 (A-Z)
0003            INTEGER*4 IERR,IADD,LSTX
0004            CHARACTER*1 FILE
0005            COMMON/    /LM,L1,QN,SIGMA,QEI,H,EI
0006            DIMENSION LINC(3)
0007            DATA LINC/50.,10.,1./
   B
0009            PARAMETER (PI=3.141592654)
0010      C
0011            WRITE(6,1)
0012            WRITE(6,2)
0013            WRITE(6,3)
0014            READ(5,*)OD
0015            WRITE(6,4)
0016            READ(5,*)T
0017            WRITE(6,5)
0018            READ(5,*)EM
0019            WRITE(6,6)
0020            READ(5,*)GAMMA
0021            WRITE(6,7)
0022            READ(5,*)SG
0023            WRITE(6,8)
0024            READ(5,*)SIGINT
0025            WRITE(6,9)
0026            READ(5,*)STRLMT
0027            WRITE(6,10)
0028            READ(5,*)H
0029      C
0030            WRITE(6,11)
0031            WRITE(6,12)
0032            READ(5,13)FILE
0033            IERR= STR$UPCASE(FILE,FILE)
0034            ID= OD - 2.*T
0035            IM= PI*(OD**4.-ID**4.)/64.
0036            EI= EM*IM
0037            AREA= PI*(OD*OD-ID*ID)/4.
0038            AREAI= PI*ID*ID*9.01331D-03
0039            QM= GAMMA*AREA + SG*AREAI
0040            QEI=QM/EI
0041            IF(FILE.EQ.'Y')THEN
0042            WRITE(1,1)
0043              WRITE(1,3)OD
0044              WRITE(1,4)T
0045              WRITE(1,5)EM
0046              WRITE(1,6)GAMMA
0047              WRITE(1,7)SG
0048              WRITE(1,8)SIGINT
0049              WRITE(1,9)STRLMT
0050              WRITE(1,10)H
```

```
      ENDIF
      LMO=0.0
      LM1=0.0
      LMIN= 0.0
      L=10.0
      STMN= 1.D+20
      IF((SIGINT.GE.STRLMT).AND.(H.EQ.0.0))THEN
          STMN= SIGINT
          STL=0.0
          GOTO 600
      ENDIF
      LSTX=0
      IADD= 1
      MC5=1.D+12
      MC4=1.D+11
      MC3=1.D+10
      MC2=1.D+09
      MC1=1.D+08
      MC0=1.D+07
      LMO= 0.0
  100 STLX= 0.75*STL + 0.25*STOLD
      STOLD=STL
      SIGMA= SIGINT+STLX
      IF(SIGMA.EQ.0)SIGMA=0.1
      NM= ABS(SIGMA*AREA)
      LM= SQRT(NM/EI)
      QN= QM/NM
  200 L1= L*12.
      CALL MOMENT(MCX,STL)
      STL=STL*EM
      DSTL=ABS(STL-STOLD)
      IF(DSTL.GT.1.0)GOTO 100
      STMX= MCX*OD/2./IM + SIGMA
      IF(STMX.LE.STMN)THEN
        STMN= STMX
        LMN= L
      ENDIF
      IF(LMIN.EQ.0.0)THEN
        IF(STMX.GT.STRLMT)THEN
          LMO= L
          L=L+LINC(IADD)
          IF((L.LT.100).AND.(LSTX.EQ.0))GOTO 200
          MC5=MC4
          MC3=MC2
          MC2=MC1
          MC1=MC0
          MC0=STMX
          IF((MC0.GT.MC1).AND.(MC1.GT.MC2).AND.(MC2.GT.MC3)
```