```
0001          SUBROUTINE STRESS(SMAX,STL)
0002          IMPLICIT REAL*8 (A-Z)
0003          INTEGER*4 I,IMAX
0004     C
0005          PARAMETER (PI=3.141592654)
0006          COMMON /PARA/N,EI,L,K,QN,QK,LM,LM2,EM,OD,QEI,Q,SIGMA
0007          COMMON /CONST/WO,PO,C1,C2,C3,C4,C5,C6,C7
0008          COMMON /GREEK/AG,AG2,BG,BG2
0009     C
0010          SLEN=0.0
0011          IMAX= NINT(L/6.0) + 1
0012          DX= L/FLOAT(IMAX)
0013          DX2= DX*DX
0014          M1M=0.0
0015          M2M=0.0
0016          IF (SIGMA) 1000,2000,3000
0017     C
0018      1000 DO 1010 I=0,IMAX
0019          X= DX*FLOAT(I)
0020          X2= X*X
0021          LX= LM*X
0022          CLX= COS(LX)
0023          SLX= SIN(LX)
0024     C
0025          W1= C1 + C2*X + C3*CLX + C4*SLX + C5*X2
0026          DW=W1-WOLD
0027          WOLD=W1
0028          IF(X.NE.0.0)THEN
0029            DS=SQRT(DW*DW+DX2)
              SLEN=SLEN+DS
0031          ENDIF
0032     C
0033          M1= -LM2*(C3*CLX + C4*SLX) + 2.*C5
0034      1010 M1M= MAX(M1M,ABS(M1))
0035     C
0036          IMAX= 10
0037          DX= 12.0
0038          DO 1020 I=0,IMAX
0039          X= DX*FLOAT(I)
0040          AX= AG*X
0041          BX=BG*X
0042          EBX= EXP(-BX)
0043          CAX= COS(AX)
0044          SAX= SIN(AX)
0045          M2= EBX*((C6*(BG2-AG2)-C7*(2.*BG*AG))*CAX
0046         &   +  (C6*(2.*BG*AG)+C7*(BG2-AG2))*SAX)
0047     C
0048      1020 M2M= MAX(M2M,ABS(M2))
0049     C
0050          GOTO 5000
```

```
0051     2000 DO 2010 I=0,IMAX
0052          X= DX*FLOAT(I)
0053          x2= X*X
0054          x3= X*X2
0055          X4= X2*X2
0056     C
0057          W1= C1 + C2*X + C3*X2 + C4*X3 +C5*X4
0058          DW=W1-WOLD
0059          WOLD=W1
0060          IF(X.NE.0.0)THEN
0061            DS=SQRT(DW*DW+DX2)
0062            SLEN=SLEN+DS
0063          ENDIF
0064     C
0065          M1= 2.*C3 +6.*C4*X + 12.*C5*X2
0066     2010 M1M= MAX(M1M,ABS(M1))
0067     C
0068          IMAX= 10
0069          DX= 12.0
0070          DO 2020 I=0,IMAX
0071          X= DX*FLOAT(I)
0072          AX= AG*X
0073          EAX= EXP(-AX)
0074          CAX= COS(AX)
0075          SAX= SIN(AX)
0076          M2= 2.*AG2*EAX*(-C7*CAX + C6*SAX)
0077     C
0078     2020 M2M= MAX(M2M,ABS(M2))
0079     C
0080          GOTO 5000
0081     C
0082     3000 DO 3010 I=0,IMAX
0083          X= DX*FLOAT(I)
0084          x2= X*X
0085          LX= LM*X
0086          CLX= COSH(LX)
0087          SLX= SINH(LX)
         C
0089          W1= C1 + C2*X + C3*CLX + C4*SLX + C5*X2
0090          DW=W1-WOLD
0091          WOLD=W1
0092          IF(X.NE.0.0)THEN
0093            DS=SQRT(DW*DW+DX2)
0094            SLEN=SLEN+DS
0095          ENDIF
0096     C
0097          M1= LM2*(C3*CLX + C4*SLX) + 2.*C5
0098     3010 M1M= MAX(M1M,ABS(M1))
0099     C
0100          IMAX= 10
```

```
0101          DX= 12.0
0102          DO 3020 I=0,IMAX
0103          X= DX*FLOAT(I)
0104          AX= AG*X
0105          BX=BG*X
0106          EAX= EXP(-AX)
0107          CBX= COS(BX)
0108          SBX= SIN(BX)
0109          M2= EAX*((C6*(AG2-BG2)-C7*(2.*AG*BG))*CBX
0110         &   +  (C6*(2.*AG*BG)+C7*(AG2-BG2))*SBX)
0111      C
0112       3020 M2M= MAX(M2M,ABS(M2))
0113      C
0114          GOTO 5000
0115        5000 STL=EM*(SLEN/L-1.0)
0116          MMAX=MAX(M1M,M2M)
0117          SMAX=ABS(SIGMA)+EM*MMAX*OD/2.
0118          RETURN
0119      C
0120          END
```

PROGRAM TRENCH

The computer program TRENCH calculates the trench profile and pipeline stresses for lowering a pipeline onto a contoured trench surface. The program is based on the equations for the contoured trench solution described in Appendix 8. The input for 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, γ (lb/in^3)
5. Soil stiffness parameter, k (psi)
6. Existing axial tensile stress in the pipeline, σ_L (psi)
7. Maximum allowable axial stress, σ_{max} (psi)
8. Maximum lowering depth, H (inches).

The TRENCH program calculates the minimum* trench length, for the given lowering depth and maximum allowable stress. 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 and the maximum stress is less than or equal to the maximum allowable stress.

The equations in the program were derived for a pipeline in tension. If the acutal axial stress (initial plus lengthening) is compressive, the program uses a very small tensile stress for the axial stress. This results in a longer, and thus, conservative, contoured trench than is actually required. With a longer trench, both the lengthening stress and bending stress are reduced.

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 FORØØØ.DAT, if desired. The output consists of:

1. An echo of the input parameters
2. The minimum trench length and maximum stress
3. Trench depth at selected intervals (on FORØØ1.DAT only)

4. Intermediate solutions, showing trench length, lengthening stress, and maximum stress (terminal screen only).

A computer listing of the program (VAX-11/780 version) is given below. Note that the coefficients require the use of REAL*16 variables to obtain an accurate solution. However, even with this high degree of precision, some situations lead to erroneous solutions, such as lowering a 8-inch pipe with an initial axial stress of 15 ksi to a depth of ten feet.

---

* Minimum refers to the minimum length for a trench profile of the form described in Appendix 8. Although there are an infinite number of possible trench profiles, this type was chosen because of the relatively simple equations involved and the trench lengths appear to be reasonable.

```
0001            PROGRAM TRENCH
0002            IMPLICIT REAL*16 (A-Z)
0003            INTEGER*4 IADD,I,IMAX,ITN,ISX,IZ
0004            CHARACTER*1 ANS,FILE,ITER
0005      C
0006            PARAMETER (PI=3.141592654)
0007      C
0008            COMMON/PARA/QK,FP,KS,NM,QON,QOK,W1,H1,L,LO,LM,LM2
0009            COMMON/CONST/C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,T1,T2,T3,T4
0010            COMMON/GREEK/AG,BG,AG2,BG2,EO,FO,GO,HO
0011            COMMON/INPUT/OD,TID,EM,GAMMA,SIGMA,STRLMT,SIGINT
0012            COMMON/STRM/DXL,ST1,ST2,ST3
0013            DIMENSION FINC(7),DX(4)
0014            DATA FINC/0.1,0.01,0.001,0.0001,0.00001,0.000001,0.0000001/
0015            DATA DX/60.,24.,12.,6./
0016      C
0017            IADD= 0
0018            WRITE(6,1)
0019            WRITE(6,2)
0020            WRITE(6,3)
0021            READ(5,*)OD
0022            WRITE(6,4)
0023            READ(5,*)TID
0024            WRITE(6,5)
0025            READ(5,*)EM
0026            WRITE(6,6)
0027            READ(5,*)GAMMA
0028            WRITE(6,7)
0029            READ(5,*)KS
0030            WRITE(6,8)
0031            READ(5,*)SIGINT
0032            WRITE(6,9)
0033            READ(5,*)STRLMT
0034            WRITE(6,10)
0035            READ(5,*)W1
0036            FPS=0.935
0037      C
0038            WRITE(6,11)
0039            WRITE(6.12)
0040            READ(5,13)FILE
0041            IF(FILE.EQ.'y')FILE='Y'
0042            WRITE(6,14)
0043            READ(5,13)ITER
0044            IF(ITER.EQ.'y')FILE='Y'
0045            ID= OD - 2.*TID
0046      C
0047            IM= PI*(OD**4-ID**4)/64.
0048            EI= EM*IM
0049            AREA= PI*(OD*OD-ID*ID)/4.
0050            QM= GAMMA*AREA
```

```
0051          QK=QM/KS
0052          STM=0.0
0053          STL= 0.0
0054          STOLD= 0.0
0055          LC=SQRT(KS/4./EI)
0056          FP= FPS
0057          FMAX= 0.935
0058          IF(FILE.EQ.'Y')THEN
0059            WRITE(1,1)
0060            WRITE(1,3)OD
0061            WRITE(1,4)TID
0062            WRITE(1,5)EM
0063            WRITE(1,6)GAMMA
0064            WRITE(1,7)KS
0065            WRITE(1,8)SIGINT
0066            WRITE(1,9)STRLMT
0067            WRITE(1,10)W1
0068          ENDIF
0069       50 ISX= 1
0070          DXL= DX(ISX)
0071          IF(ITER.EQ.'Y')WRITE(6,26)DXL
0072      100 STLX= 0.70*STL + 0.30*STOLD
0073          SIGMA=SIGINT+STLX
0074          STOLD=STLX
0075          IF(SIGMA.LE.0.0)SIGMA=0.1
0076          NM= ABS(SIGMA*AREA)
0077          DC= NM/4./EI
0078          BG=SQRT(LC+DC)
0079          AG=SQRT(LC-DC)
0080          AG2= AG*AG
0081          BG2= BG*BG
0082          E0= BG2-AG2
0083          F0= 2.*AG*BG
0084          G0= AG*(3.*BG2-AG2)
0085          H0= BG*(BG2-3.*AG2)
0086          LM2= NM/EI
0087          LM=SQRT(LM2)
0088          CALL DEPTH
0089          L1= L*3.
0090          CLL= COSH(LM*L1)
0091          T3= QON/LM2/CLL
0092          T4= -T3
0093          T2=  T3
0094          T1= H1-T2
0095          CALL COEFF
0096          CALL STRMAX(SMAX,STL)
0097          IF(ITER.EQ.'Y')WRITE(6,15)L,FP,STL,SMAX
0098          SDIF=ABS(STL-STOLD)
0099          DLL= ABS(L-L0)
0100          IF(DLL.GT.0.001)GOTO 100
```