APPENDIX

Source code for optimisation of surface area with residence time and geometric constraints

```
C**** SPECIFY NUMBER OF VARIABLES N ***********************************C
C                                                                       C
      N=3
C                                                                       C
C*****SPECIFY STARTING POINT (INITIAL GUESS) : X(I), I=1,N ************C
C                                                                       C
      X(1)=0.38016051d0
      X(2)=0.5d0
      X(3)=0.63872483d0
C                                                                       C
C*****SPECIFY NUMBER OF INEQUALITIES NI (NI=0 IF NONE)*****************C
C                                                                       C
      NI=10

C                                                                       C
C*****SPECIFY NUMBER OF EQUALITIES NE   (NE=0 IF NONE)*****************C
C                                                                       C
      NE=0
C                                                                       C
C***********************************************************************C
C               ONLY REQUIRED FOR CONSTRAINTS                           C
C     SPECIFY PENALTY FUNCTION PARAMETERS : XMU & XMUMAX                C
C                                                                       C
C     STANDARD SETTINGS :  XMU=100.0  ;  XMUMAX=10000.0                 C
C                                                                       C
C***********************************************************************C
C                                                                       C
      XMU=1.D2
      XMUMAX=1.D4
C                                                                       C
C***********************************************************************C
C     SPECIFY CONVERGENCE CRITERIA TOLERANCES : EG & XTOL               C
C                                                                       C
C     CODE TERMINATES IF                                                C
C               || X(CURRENT STEP) - X(PREVIOUS STEP)|| < XTOL          C
C     OR                                                                C
C               || GRADIENT VECTOR OF PENALTY FUNCTION || < EG          C
C***********************************************************************C
C                                                                       C
      XTOL=1.D-9
      EG=1.0D-8
C                                                                       C
C***********************************************************************C
C     SET MAXIMUM STEP SIZE : DELT                                      C
C                                                                       C
C     DELT SHOULD BE OF THE SAME ORDER OF MAGNITUDE AS THE "DIAMETER"   C
C     OF THE REGION OF INTEREST : DELT = SQRT(N)*VARIABLE-RANGE         C
C***********************************************************************C
C                                                                       C
      DELT=1
C                                                                       C
C***********************************************************************C
C     SPECIFY MAXIMUM NUMBER OF STEPS PER PHASE (THREE PHASES: 0,1&2)   C
C     ONE STEP = EVALUATION OF ONE SET OF GRADIENTS = ONE DER CALL      C
C***********************************************************************C
C                                                                       C
      KMAX=100000
C                                                                       C
C***********************************************************************C
C    IPRINT CONTROLS PRINTING : PRINTING OCCURS EVERY ]IPRINT] STEPS.   C
C         IPRINT=KMAX+1:PRINTING ON STEP 0 AND ON EXIT ONLY.            C
C     VALUES OF X SUPPRESSED ON INTERMEDIATE STEPS IF IPRINT.LT.0       C
C***********************************************************************C
```

```
C                                                                       C
      IPRINT= 10
C                                                                       C
C***********************************************************************C
C     USER SPECIFIED SUBROUTINE                                         C
C                                                                       C
      SUBROUTINE FUN(N,X,F)
C                                                                       C
C     COMPUTE OBJECTIVE FUNCTION : F                                    C
C                                                                       C
C***********************************************************************C
      implicit real*8 (a-h,o-z), integer (I-N)
      real pi,epsilon,a,vb,V,num,SA
      dimension X(N)

      pi=3.141595d0
      epsilon = 0.4d0

      F =(-1.d0*3*pi*(X(1)**2.d0)*X(3)*(1.d0-epsilon))/(200.d0*X(2))
      return
      end

C***********************************************************************C
C     USER SPECIFIED SUBROUTINE                                         C
C                                                                       C
      SUBROUTINE CONIN(N,NI,X,C)
C                                                                       C
C     COMPUTE INEQUALITY CONSTRAINT FUNCTIONS                           C
C                                                                       C
C***********************************************************************C
      IMPLICIT REAL*8 (A-H,O-Z),INTEGER(I-N)
      real pi,X1min,X1max,X2min,X2max,X3min,X3max
      DIMENSION X(N),C(NI)

      pi=3.141595d0
      epsilon = 0.4d0

      X1min = 0.1d0
      X1max = 1.0d0
      X2min = 0.5d0
      X2max = 7.5d0
      X3min = 0.1d0
      X3max = 3.0d0

      C(1)  = X(1) - X1max
      C(2)  = X1min - X(1)
      C(3)  = X(2) - X2max
      C(4)  = X2min - X(2)
      C(5)  = X(3) - X3max
      C(6)  = X3min - X(3)
      C(7)  = X(2) - 100*X(1)
      C(8)  = (X(1)**2)*X(3) - 0.116/(pi*epsilon)
      C(9)  = 0.058/(pi*epsilon) - (X(1)**2)*X(3)
      C(10) = X(1)/X(3) - (2d0/3d0)


      RETURN
      END
C***********************************************************************C
C     USER SPECIFIED SUBROUTINE                                         C
C                                                                       C
      SUBROUTINE CONEQ(N,NE,X,H)
C                                                                       C
C     COMPUTE EQUALITY CONSTRAINT FUNCTIONS                             C
C                                                                       C
C***********************************************************************C
      IMPLICIT REAL*8 (A-H,O-Z),INTEGER(I-N)
      DIMENSION X(N),H(NE)
C
c-----------no eq constr
c
```

```
      RETURN
      END
C*********************************************************************C
C     USER SPECIFIED SUBROUTINE                                       C
C                                                                     C
      SUBROUTINE GRADF(N,X,GF)
C                                                                     C
C     COMPUTE THE GRADIENT VECTOR OF THE OBJECTIVE FUNCTION W.R.T.    C
C     THE VARIABLES X(I) : GF(I),I=1,N                                C
C                                                                     C
C*********************************************************************C
      IMPLICIT REAL*8 (A-H,O-Z),INTEGER(I-N)
      DIMENSION delx(N),X(N),GF(N),Xst(N)
      open(unit=3,file='history.out',status='unknown')
      pi=3.141595d0
      epsilon = 0.4d0
c start save to output
      F =(-1.d0*3*pi*(X(1)**2.d0)*X(3)*(1.d0-epsilon))/(200.d0*X(2))
      f0=F
      write(3,123)f0,X(1),X(2),X(3)
c   to save to output
      GF(1) = -1.d0*3.d0*pi*X(1)*X(3)*(1-epsilon)/(100d0*X(2))
      GF(2) = 3d0*pi*(X(1)**2)*X(3)*(1-epsilon)/(200d0*(X(2)**2))
      GF(3) = -1d0*3d0*pi*(X(1)**2)*(1-epsilon)/(200d0*X(2))
  123 FORMAT(4E13.6)
      RETURN
      END
C                                                                     C
C*********************************************************************C
C     USER SPECIFIED SUBROUTINE                                       C
C                                                                     C
      SUBROUTINE GRADC(N,NI,X,GC)
C                                                                     C
C     COMPUTE THE GRADIENT VECTORS OF THE INEQUALITY CONSTRAINTS CJ   C
C     W.R.T. THE VARIABLES X(I) : GC(J,I), J=1,NI I=1,N               C
C                                                                     C
C*********************************************************************C
      IMPLICIT REAL*8 (A-H,O-Z),INTEGER(I-N)
      DIMENSION X(N),GC(NI,N),Xst(N)

      GC(1,1)  =  1d0
      GC(2,1)  = -1d0
      GC(3,2)  =  1d0
      GC(4,2)  = -1d0
      GC(5,3)  =  1d0
      GC(6,3)  = -1d0
      GC(7,1)  = -100d0
      GC(7,2)  =  1.d0
      GC(8,1)  =  2d0*X(1)*X(3)
      GC(8,3)  =  X(1)**2
      GC(9,1)  = -2d0*X(1)*X(3)
      GC(9,3)  = -1d0*(X(1)**2)
      GC(10,1) =  1d0/X(3)
      GC(10,3) = -1d0*X(1)/(X(3)**2)

      RETURN
      END
C                                                                     C
C*********************************************************************C
C     USER SPECIFIED SUBROUTINE                                       C
C                                                                     C
      SUBROUTINE GRADH(N,NE,X,GH)
C                                                                     C
C     COMPUTE THE GRADIENT VECTORS OF THE EQUALITY CONSTRAINTS HJ     C
C     W.R.T. THE VARIABLES X(I) : GH(J,I), J=1,NE I=1,N               C
C                                                                     C
C*********************************************************************C
      IMPLICIT REAL*8 (A-H,O-Z),INTEGER(I-N)
      DIMENSION X(N),GH(NE,N)
C
c--------------no eq
```

```
C
      RETURN
      END
C                                                                       C
C*************************************************************************C
C                            FINISH                                       C
C*************************************************************************C
```