NASA/TM—2016–219104

Use of the NLP10x10 Sequential Quadratic Programming Algorithm to Solve Rotorcraft Hub Loads Minimisation Problems

Appendix D:
Fortran Codes for the NLP10x10 Main Driver and Subroutines

Jane Anne Leyland
Ames Research Center
Moffett Field, California

May 2016

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM—2016–219104

Use of the NLP10x10 Sequential Quadratic Programming Algorithm to Solve Rotorcraft Hub Loads Minimisation Problems

Appendix D:
Fortran Codes for the NLP10x10 Main Driver and Subroutines

Jane Anne Leyland
Ames Research Center
Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, CA 94035-1000

May 2016

Available from:

NASA STI Support Services
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
757-864-9658

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312
webmail@ntis.gov
703-605-6000

This report is also available in electronic form at

http://ntrs.nasa.gov

Table of Contents

D.1 NLP10x10 Main Driver Code

The purpose of the NLP10x10 Main Driver Code is to read and process the input, initialise the data prior to optimisation, and perform the required optimisation.

```
C
      PROGRAM NLP10x10
C
C***********************************************************************
C
C
C     S E Q U E N T I A L   Q U A D R A T I C   P R O G R A M M I N G
C
C             A L G O R I T H M   F O R   C O N S T R A I N E D
C
C                      O P T I M I Z A T I O N
C
C
C
C               EASY-TO-USE VERSION WITH NUMERICAL GRADIENTS
C
C
C
C
C   GENERAL PROBLEM DESCRIPTION:
C   -------------------
C
C   The program solves the general nonlinear programming problem
C
C
C
C             Minimize      F(CV)
C
C             subject to    G (CV)   =  0  ,  j=1,...,m
C                            j                        e
C
C                           G (CV)  >=  0  ,  j=m +1,...,m
C                            j                   e
C
C                           CVL  <=  CV  <=  CVU
C
C
C
C   with differentiable, real-valued functions subject to an n-dimensional
C   vector X.
C
C
C
C   THE NUMERICAL ALGORITHM:
C   -----------------------
C
C   The used code with name 'NLPQLP' is based on a sequential quadratic programming
C   (SQP) method. In each iteration, a linearly constrained quadratic subproblem
C   is formulated by approximating the Lagrangian function quadratically and
C   by linearizing constraints. The Hessian matrix is computed by BFGS quasi-Newton
C   updates. Subsequently, a one-dimensional line search subject to an
C   augmented Lagrangian penalty function is performed to get the next iterate..
C
C
C
C   SPECIFIC PROBLEM DESCRIPTION:  Control Optimisation for a Linear Plant
C                                  [(10 x 10) T-Matrix] Model with Constraints.
C   --------
C
C   Minimize  F  =  EC1*EC1  +  EC2*EC2  +  EC3*EC3  +  EC4*EC4  +  EC5*EC5  +
C                   EC6*EC6  +  EC7*EC7  +  EC8*EC8  +  EC9*EC9  +  EC10*EC10
C
C   Subject to:   CVL(1)   <=   CV1   <=   CVU(1)
C                 CVL(2)   <=   CV2   <=   CVU(2)
C                 CVL(3)   <=   CV3   <=   CVU(3)
C                 CVL(4)   <=   CV4   <=   CVU(4)
C                 CVL(5)   <=   CV5   <=   CVU(5)
C                 CVL(6)   <=   CV6   <=   CVU(6)
C                 CVL(7)   <=   CV7   <=   CVU(7)
C                 CVL(8)   <=   CV8   <=   CVU(8)
C                 CVL(9)   <=   CV9   <=   CVU(9)
C                 CVL(10)  <=   CV10  <=   CVU(10)
C
C                 A(1)     <=   GMAX(1)
```

```
C                 A(2)     <=   GMAX(2)
C                 A(3)     <=   GMAX(3)
C                 A(4)     <=   GMAX(4)
C                 A(5)     <=   GMAX(5)
C
C                 A(1) + A(2) + A(3) + A(4) + A(5)  <=  GMAX(6)
C
C
C
C   Where:        NMAX  >=  (N = NX) + 2     Use:  NMAX  =  (N = NX) + 4
C                 MMAX  >=  (M = MG) + 1     Use:  MMAX  =  (M = MG) + 4
C
C                 (N = NX)  =  Number of optimisation variables.  =  10
C                 (M = MG)  =  Total Number of Constraints.       =   6
C
C                 NMAX  =  14
C                 MMAX  =  10
C
C
C
C   VERSION:
C   -------
C
C   3.1  -  February 2010
C
C
C*********************************************************************
C
      IMPLICIT         NONE
C
C
C  *****   These statements establish and define the Dimensions and
C                Type of some of the Non-linear Programming Parameters.   *****
C
      INTEGER          NMAX, MMAX, MNN2X, LWA, LKWA, LACTIV
C
      PARAMETER (      NMAX   = 14,
     /                 MMAX   = 10,
     /                 MNN2X  = MMAX + NMAX + NMAX + 2,
     /                 LWA    = 1.5*NMAX*NMAX + 33*NMAX + 9*MMAX + 200,
     /                 LKWA   = NMAX + 20,
     /                 LACTIV = 2*MMAX + 10)
C
C
      INTEGER          KWA(LKWA), N, ME, M, L, MNN2, MAXIT, MAXFUN,
     /                 IPRINT, MAXNM, IOUT, MODE, IFAIL, I, J, NFUNC
C
      DOUBLE PRECISION CV(NMAX), F, G(MMAX), DF(NMAX), DG(MMAX,NMAX),
     /                 U(MNN2X), CVL(NMAX), CVU(NMAX), C(NMAX,NMAX),
     /                 D(NMAX), WA(LWA), ACC, ACCQP, STPMIN, EPS,
     /                 EPSREL, FBCK, GBCK(MMAX), RHOB
C
      LOGICAL          ACTIVE(LACTIV), LQL
C
      EXTERNAL         QL
C
C
C  *****   These statements establish and define the Dimensions and
C                Type of some of the NLP10x10 Main Driver Parameters.   *****
C                                                                          *
      INTEGER*4  NCC,   NXX,   NXZ,   NZNX,   NZZ
C                                                                          *
      PARAMETER  (NCC=6,NXX=10,NZZ=10,NZNX=NZZ*NXX,NXZ=MAX(NXX,NZZ))
C
      REAL*8    RANQ
C
      CHARACTER*2  ITC
C
      INTEGER*4  CVOUT,  ICASE,  ICYCL,  ICYCL0, IDATA,  II,     II1,
     1           II2,    IIC,    IN,     INFO,   IOPT,   IPIV(NXX),
     2           ISEED1(4),      ISEED2(4),      ISEED3(4),
     3           ISEED4(4),      ITOUT,  IZERO,  JJ,     LSAVE(NCC),
     4           LWORK,  MG,     MI,     MSAVE(NZZ),     MSAVE0(NZZ),
     5           MULT,   NSAVE(NXX),     NSAVE0(NXX),    NX,     NX0,
```

```
     6           NXXX,   NZ,     NZ0,    NZZZ,   OPTEND
C
      PARAMETER  (LWORK=512)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      REAL*8 A(NCC),     A0(NCC),    A00(NCC),   AL(NCC),    AL0(NCC),
     1       AMP(NCC),   APRV(NCC),  APRV0(NCC), ASUM,       AU(NCC),
     2       AU0(NCC),   CRAN1,      CRAN2,      CRAN3,      CRAN4,
     3       CV0(NMAX),  CV00(NMAX), CVL0(NMAX), CVPRV(NMAX),
     4       CVPRV0(NMAX),           CVU0(NMAX), DCV(NMAX),  DEG,
     5       DUM,        EC(NZZ),    EC0(NZZ),   ECPRV(NZZ),
     6       ECPRV0(NZZ),            ECPRV00(NZZ),
     7       ECRAN0(NZZ),            G0(MMAX),   GMAX(MMAX), MAXASUM,
     8       MINASUM,    ONE,        ONE80,      PHASE(NCC),
     9       PHASE0(NCC),            PHS(NCC),   PHSPRV(NCC),
     o       PHSPRV0(NCC),           PI,         RAD,        SUMF,
     1       SUMZ,       T(NZZ,NXX), T0(NZZ,NXX),            THREE60,
     2       TWO,        WDT(NZZ),   WDT0(NZZ),  ZERO
C
      REAL*8  ALPHA,              DD(NXX,NXX),       DELTCV(NXX),
     1        DUMQ(NZZ,1),        DUMT(NXX,1),       DUMT1(NXX,1),
     2        DUMTT(NXX,NXX),     DUMX(NXX,1),       DUMX1(NXX,1),
     3        DUMXX(NXX,NXX),     DUMXX1(NXX,NXX),   DUMXZ(NXX,NZZ),
     4        DUMZ(NZZ,1),        DUMZT(1,NZZ),      EE(NXX,NXX),
     5        FF(NXX,NXX),        JJJ(1,1),          RSSDCV,
     6        THETA(NXX),         TT(NZZ,NXX),       TTT(NXX,NZZ),
     7        WDX(NXX),           WDXX(NXX,NXX),     WORK(LWORK),
     8        WX(NXX),            WXX(NXX,NXX),      WZ(NZZ),
     9        WZZ(NZZ,NZZ),       ZZ(NZZ)
C
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      DATA  A00,                ACC,      ACCQP,    AL0,
     1      ALPHA,    AU0,                CRAN1,    CRAN2,    CRAN3,
     2      CRAN4,    CVOUT,    ECRAN0,             EPS,      ICASE,
     3      ICYCL0,   IDATA,    IN,       IOPT,     IOUT,     IPRINT,
     4      ISEED1(1),          ISEED1(2),          ISEED1(3),
     5      ISEED1(4),          ISEED2(1),          ISEED2(2),
     6      ISEED2(3),          ISEED2(4),          ISEED3(1),
     7      ISEED3(2),          ISEED3(3),          ISEED3(4),
     8      ISEED4(1),          ISEED4(2),          ISEED4(3),
     9      ISEED4(4),          ITC,      ITOUT,    IZERO,    L,
     A      LQL,      LSAVE,    MAXFUN,   MAXIT,    MAXNM,    MAXASUM,
     B      MI,       MINASUM,  MODE,     MSAVE0,   MULT,     NSAVE0,
     C      NX0,      NZ0,      ONE,      ONE80,              OPTEND,
     D      PHASE0,             PI,                           RHOB,
     E      STPMIN,   T0,                 THREE60,            TWO,
     F      WDT0,               WDX,                WX,
     G      WZ,                 ZERO                                /
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
     o      NCC*1.0D+00,        1.0D-07,  0.0D+00,  NCC*-1.0D-07,
     1      1.0D+00,  NCC*3.0D+00,        2.0D+00,  3.0D+00,  1.0D+00,
     2      1.0D+00,  0,        NZZ*0.0D+00,        1.0D-07,  1,
     3      2000,     1,        5,        1,        6,        2,
     4      2395,               4013,               3813,
     5      1837,               1843,               4011,
     6      3364,               2835,               3962,
     7      1111,               3215,               2637,
     8      2397,               1504,               4031,
     9      3173,               ' ',      1,        0,        1,
     A      .TRUE.,   NCC*1,    30,       100,      0,        12.0D+00,
     B      0,        0.0D+00,  0,        NZZ*1,    0,        NXX*1,
     C      10,       10,       1.0D+00,  180.0D+00,          3,
     D      NCC*90.0D+00,       3.141592653589793D+00,        100.00,
     E      0.0D+00,  NZNX*0.0D+00,       360.0D+00,          2.0D+00,
     F      NZZ*1.0D+00,        NXX*0.0D+00,        NXX*0.0D+00,
     G      NZZ*1.0D+00,        0.0D+00                         /
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
```

```
 1000 FORMAT(//9X,45H   *****   Number of Function Evaluations  = ,I6,
     1 8H   *****)
 1901 FORMAT(//2H  )
 1902 FORMAT(///79H ****************************************************
     1**************************////8X,39H   ***************   Start Cas
     2e Number ,I4,18H   ***************/)
 1903 FORMAT(//2X,49H *****   Solution Control Vector for Case Number
     1,I4,8H   *****)
 1904 FORMAT(5X,I2,3X,3D20.8)
 1905 FORMAT(4D20.8)
 1906 FORMAT(//2X,43H *****   NLP Solution Performance Index  = ,D16.8,
     18H   *****)
 1907 FORMAT(/2X,77H *****   Previous Actual NLP Control Vector CVPRV0 B
     1efore Compression   *****)
 1908 FORMAT(//17X,30H **********   End Case Number ,I4,13H   **********
     1/)
 1909 FORMAT(//2X,38H *****   Initial Performance Index  = ,D16.8,
     18H   *****)
 1910 FORMAT(5X,I2,5X,I2,5X,I2,5X,2D20.8)
 1911 FORMAT(/2X,8H Element,7X,5H CVL0,15X,7H CVPRV0,13X,5H CVU0/)
 1912 FORMAT(19X,41H *****   Completed CALL to NLPQLP   *****)
 1913 FORMAT(/7X,6H CRAN1,14X,6H CRAN2,14X,6H CRAN3,14X,6H CRAN4/)
 1914 FORMAT(/5X,48H   ***************   INPUT DATA for Case Number ,I4,
     1 18H   ***************//)
 1915 FORMAT(///79H ****************************************************
     1**************************///3X,49H   ***************   OUTPUT DAT
     2A for Case Number ,I4,18H   ***************//)
 1916 FORMAT(/2X,39H *****   Inequality Constraints   *****//2X,8H Eleme
     1nt,7X,11H Constraint,9X,10H Amplitude,10X,8H Max Amp/)
 1917 FORMAT(//60H *****   Initial Constraint Function Values for Case N
     1umber ,I4,8H   *****)
 1918 FORMAT(//61H *****   Solution Constraint Function Values for Case
     1Number ,I4,8H   *****)
 1919 FORMAT(/4X,3H II,3X,4H II1,3X,4H II2,13X,8H CV0(II),11X,
     1 9H CV0(II2)/)
 1920 FORMAT(//2X,50H *****   End Conditions Vector EC0, Previous Cycle/
     120X,44H End Conditions Vector ECPRV0, and Weighting/29x,51H Coeffi
     2cient Vector WDT0 Before Compression   *****/)
 1921 FORMAT(//2X,49H *****   Initial End Conditions Vector EC   *****//
     1)
 1922 FORMAT(/2X,8H Element,3X,4H AL0,14X,6H APRV0,12X,4H AU0,14X,8H PHS
     1PRV0/)
 1923 FORMAT(5X,I2,1X,4D18.8)
 1924 FORMAT(//2X,26H *****   Error   IDATA  = ,I3,8H   *****)
 1925 FORMAT(//2X,8H Element,7X,8H ECPRV00,12X,7H ECRAN0,13X,7H ECPRV0/)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 1926 FORMAT(2X,58H *****   Randomly Define the Initial T-Matrix (T0) an
     1d the/20X,53H Initial Previous Control Amplitude (APRV0) and Phase
     2/29X,51H Angle (PHSPRV0) Vectors Before Compression   *****/)
 1927 FORMAT(/2X,8H Element,3X,4H AL0,14X,3H A0,15X,4H AU0,14X,7H PHASE0
     1/)
 1928 FORMAT(3X,I4,6X,I2,5X,2D20.8)
 1929 FORMAT(/2X,8H Element,7X,5H CVL0,15X,4H CV0,16X,5H CVU0/)
 1930 FORMAT(2X,57H *****   The Initial T-Matrix (T0) and Either the Ini
     1tial/20X,47H Previous Actual NLP Control Vector (CVPRV0) or/26X
     2,47H the Initial Previous Control Amplitude (APRV0)/32X,41H and Ph
     3ase Angle (PHSPRV0) Vectors Before/41X,39H Compression are Directl
     4y Input   *****/)
 1931 FORMAT(//2X,29H *****   NLP Special Control ,I2,22H Vector Output
     1  *****)
 1932 FORMAT(/2X,5H CV =,/(5X,3(D24.15,1H,)))
 1933 FORMAT(/2X,65H *****   Either the BEFORE Compression Initial Actua
     1l NLP Control/14X,51H Vector Estimates (CV00), OR the BEFORE Compr
     2ession/17X,50H Initial Control Vector Amplitudes (A00) and Phase/
     3 20X,45H Angles (PHASE0) Estimates are Directly Input/23X,57H via
     4NAMELIST Data CDATA                            *****/)
 1934 FORMAT(/2X,62H *****   Initial Actual NLP Control Vector Estimates
     1 (CV0) and/28X,52H Its Limits (CVL0 & CVU0) Before Compression   *
     2****)
 1935 FORMAT(/2X,61H *****   Initial Control Amplitude Vector Estimates
     1(A0), Its/23X,47H Limits (AL0 & AU0), and Its Phase Angle Vector/
     2 34X,46H Estimates (PHASE0) Before Compression   *****)
```

```
 1936 FORMAT(/2X,47H *****   Predicted Control Amplitude Vector (A),/19X
     1,42H Its Limits (AL & AU), and Its Phase Angle/38X,42H Vector (PHA
     2SE) Before Compression   *****)
 1937 FORMAT(/2X,8H Element,3X,4H CVL,14X,3H CV,15X,4H CVU,14X, 9H CV -
     1CV0/)
 1938 FORMAT(/2X,8H Element,3X,3H AL,14X,2H A,16X,3H AU,15X,6H PHASE/)
 1939 FORMAT(//18X,26H **********   Case Number ,I4,13H   **********//)
 1940 FORMAT(/37X,6H LSAVE/)
 1941 FORMAT((10I7/))
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 1942 FORMAT( /15X,50H Specification of CV, T-matrix, and EC Compression
     1//33X,13H MSAVE0/MSAVE/)
 1943 FORMAT((I11,4I14))
 1944 FORMAT(/33X,13H NSAVE0/NSAVE/)
 1945 FORMAT(/2X,48H *****   Predicted Measurement Vector EC   *****/)
 1946 FORMAT(//2X,54H *****   No Constraints are Specified for Case Numb
     1er ,I4,8H   *****//)
 1947 FORMAT(/2X,78H *****   Initial Control Amplitude (A) and Phase Ang
     1le (PHASE) Vectors   *****)
 1948 FORMAT(/2X,61H *****   Initial Previous Control Amplitude (APRV0)
     1and Phase/29X,51H Angle (PHSPRV0) Vectors before Compression   ***
     2**)
 1949 FORMAT(2X,7H ICYCL0,3X,8H Element,9X,6H APRV0,14X,8H PHSPRV0/)
 1950 FORMAT(//2X,50H *****   Solve the NLPQLP Problem for Case Number ,
     1 I4,8H   *****)
 1951 FORMAT(//7X,51H *****   The Initial Previous Phase Angle (PHSPRV0)
     1/32X,46H is Not in the Principal Cycle [0,360)   *****/)
 1952 FORMAT(2X,54H *****   Input the Initial Previous Actual NLP Contro
     1l/19X,50H Vector (CVPRV0) Directly Via NAMELIST Input CDATA/25X
     2,56H and then Compute the Previous Control Amplitude (APRV0)/30X
     3,50H and Control Phase Angle (PHSPRV0) Vectors   *****/)
 1953 FORMAT(2X,53H *****   Input the Initial Previous Control Amplitude
     1/25X,42H (APRV0) and Phase Angle (PHSPRV0) Vectors/38X,42H Directl
     2y Via NAMELIST Input CDATA   *****/)
 1954 FORMAT(2X,61H *****   Input the Initial Actual NLP Control Vector
     1Estimate/16X,46H (CV00) Directly Via NAMELIST Input CDATA, and/24X
     2,43H then Compute the Control Vector Amplitudes/30X,50H (A00) and
     3Phase Angles (PHASE0) Estimates   *****/)
 1955 FORMAT(2X,62H *****   Adjust the Initial Control Amplitude Estimat
     1es Vector/11X,56H (A00) to define the Initial Control Amplitude Es
     2timates/12X,68H Vector (A0) to Within Limits If Required Before Co
     3mpression   *****/)
 1956 FORMAT(/2X,8H Element,7X,4H EC0,16X,7H ECPRV0,13X,5H WDT0/)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 2906 FORMAT(//51H   *****   Regulator Solution Performance Index  = ,
     1 D16.8,8H   *****)
 2928 FORMAT(5X,I2,1X,4D18.8/(8X,4D18.8))
 2931 FORMAT(//10X,37H   *****   Regulator Special Control ,I2,22H-Vecto
     1r Output   *****)
 2950 FORMAT(//2X,53H *****   Solve the Regulator Problem for Case Numbe
     1r ,I4,8H   *****/)
 2960 FORMAT(/4X,4H Row,4X,49H *****   [DUMXX1] = Matrix to be Inverted
     1  *****/)
 2961 FORMAT(/4X,4H Row,2X,54H *****   [DD] = The Inverse of Matrix [DUM
     1XX1]   *****/)
 2962 FORMAT(/4X,4H Row,1X,56H *****   [EE] = The Identity Matrix [DUMXX
     11][DD]   *****/)
 2963 FORMAT(/4X,4H Row,1X,56H *****   [FF] = The Identity Matrix [DD][D
     1UMXX1]   *****/)
 2964 FORMAT(/20X,19H *****   Alpha  =  ,D18.8,8H   *****/)
 2965 FORMAT(/3X,5H  Dim,18X,26H *****   WZ-Vector   *****/)
 2966 FORMAT(/3X,5H  Dim,18X,26H *****   WX-Vector   *****/)
 2967 FORMAT(/3X,5H  Dim,18X,27H *****   WDX-Vector   *****/)
 2968 FORMAT(//1X,62H *****   The Regulator Solution Control Vector [THE
     1TA]   *****//2X,8H Element,5X,5H [CV],11X,8H Element,5X,
     2 10H Amplitude,10X,6H Phase/)
 2969 FORMAT(//1X,53H *****   The NLP Solution Control Vector [CV]   ***
     1**//2X,8H Element,5X,5H [CV],11X,8H Element,5X,10H Amplitude,10X,
     2 6H Phase/)
 2970 FORMAT(//2X,50H *****   Matrix [DUMXX1] was Successfully Inverted/
```

```
     1 50X,30H to Yield Matrix [DD].   *****)
 2971 FORMAT(// 4X,48H  *****   Root-Sum-Squared Delta CV Elements  = ,
     1 D16.8,8H   *****)
 2972 FORMAT(/2X,11H Delta CV =,/(5X,3(D24.15,1H,)))
 2973 FORMAT(//5X,19H   *****   Element ,I4,44H  of Matrix [DUMXX1] has
     1an illegal value.  ,/38X,30H Regulator problem is stopped./38X,
     2 32H Go on to the next case.   *****)
 2974 FORMAT(//5X,19H   *****   Element ,I4,44H  of Matrix [DUMXX1] is z
     1ero and the matrix ,/38X,32H is singular; its inverse could ,/38X,
     2 17H not be computed.,/38X,30H Regulator problem is stopped./38X,
     3 32H Go on to the next case.   *****)
 2975 FORMAT(5X,I3,D20.8)
 2976 FORMAT(5X,I3,D20.8,6X,I3,D20.8,D20.8)
 2977 FORMAT(//1X,73H *****   The Regulator Solution End Conditions Vect
     1or [EC] = [ZZ]   *****//2X,8H Element,5X,12H [EC] = [ZZ]/)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 3961 FORMAT(/4X,4H Row,1X,40H *****   The Diagonal Matrix WZZ   *061*/)
 3962 FORMAT(/4X,4H Row,1X,40H *****   The Diagonal Matrix WXX   *062*/)
 3963 FORMAT(/4X,4H Row,1X,41H *****   The Diagonal Matrix WDXX   *063*/
     1)
 3964 FORMAT(/4X,4H Row,1X,34H *****   The T-Matrix [TT]   *064*/)
 3965 FORMAT(/4X,4H Row,1X,52H *****   The Transpose of the T-Matrix [TT
     1T]   *065*/)
 3966 FORMAT(/4X,4H Row,1X,41H *****   The DUMXZ-Matrix [DUMXZ]   *066*/
     1 )
 3967 FORMAT(/4X,4H Row,1X,41H *****   The DUMXX-Matrix [DUMXX]   *067*/
     1)
 3968 FORMAT(/4X,4H Row,1X,41H *****   The DUMXX-Matrix [DUMXX]   *068*/
     1)
 3969 FORMAT(/4X,4H Row,1X,41H *****   The DUMXX-Matrix [DUMXX]   *069*/
     1)
 3970 FORMAT(/4X,4H Row,1X,49H *****   The DUMX-Vector [DUMX] = [CVPRV]
     1  *070*/)
 3971 FORMAT(/4X,4H Row,1X,39H *****   The DUMX-Vector [DUMX]   *071*/)
 3972 FORMAT(/4X,4H Row,1X,49H *****   The DUMZ-Vector [DUMZ] = [ECPRV]
     1  *072*/)
 3973 FORMAT(/4X,4H Row,1X,41H *****   The DUMX1-Vector [DUMX1]   *073*/
     1)
 3974 FORMAT(/4X,4H Row,1X,41H *****   The DUMX1-Vector [DUMX1]   *074*/
     1)
 3975 FORMAT(/4X,4H Row,1X,41H *****   The DUMX1-Vector [DUMX1]   *075*/
     1)
 3976 FORMAT(/4X,4H Row,1X,39H *****   The DUMZ-Vector [DUMZ]   *076*/)
 3977 FORMAT(/4X,4H Row,1X,39H *****   The DUMZ-Vector [DUMZ]   *077*/)
 3978 FORMAT(/4X,4H Row,1X,41H *****   The DUMZT-Vector [DUMZT]   *078*/
     1)
 3991 FORMAT(/4X,4H Row,1X,41H *****   The DUMXX-Matrix [DUMXX]   *091*/
     1)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
C
      NAMELIST / CDATA / A00,    ACC,    ACCQP,  AL0,    ALPHA,  APRV0,
     1                   AU0,    CRAN1,  CRAN2,  CRAN3,  CRAN4,  CV00,
     2                   CVOUT,  CVPRV0, ECPRV0, EPS,    ICASE,  ICYCL0,
     3                   IDATA,  IN,     IOPT,   IOUT,   IPRINT, ISEED1,
     4                   ISEED2, ISEED3, ISEED4, ITOUT,  L,      LQL,
     5                   LSAVE,  MAXASUM,        MAXFUN, MAXIT,  MAXNM,
     6                   MI,     MINASUM,        MODE,   MSAVE0, MULT,
     7                   NSAVE0, NX0,    NZ0,    OPTEND, PHASE0,
     8                   PHSPRV0,        RHOB,   STPMIN, T0,     WDT0,
     9                   WDX,    WX,     WZ
C
C
C   Set some constants and initial values
C
      DEG    = ONE80/PI
      MODE   = 0
      RAD    = PI/ONE80
C
  100 READ(IN,CDATA)
C
```

```
      IFAIL  = 0
      NFUNC  = 0
      ME     = 0
      MG     = ME + MI
      M      = MG
C
      DO 101  II = 1, NZ0/2
      II2 = 2*II
      II1 = II2 - 1
      MSAVE(II1) = MSAVE0(II)
      MSAVE(II2) = MSAVE0(II)
  101 CONTINUE
C
      DO 102  II = 1, NX0/2
      II2 = 2*II
      II1 = II2 - 1
      NSAVE(II1) = NSAVE0(II)
      NSAVE(II2) = NSAVE0(II)
  102 CONTINUE
C
      WRITE(IOUT,1902)  ICASE
      WRITE(IOUT,1914)  ICASE
      WRITE(IOUT,CDATA)
      WRITE(IOUT,1915)  ICASE
C
      GO TO (10,10,20,20,30,30,998,998,998), IDATA
C
   10 WRITE(IOUT,1926)
      WRITE(IOUT,1913)
      WRITE(IOUT,1905) CRAN1, CRAN2, CRAN3, CRAN4
C
C  *****   Randomly Define the Initial T-Matrix (T0) and
C              the Initial Previous Control Amplitude (APRV0)
C                  and Phase Angle (PHSPRV0) Vectors Before Compression   *****
C
      DO 11 II = 1, NX0
      DO 12 JJ = 1, NZ0
      T0(JJ,II) = CRAN1*(TWO*RANQ(ISEED1) - ONE)
   12 CONTINUE
   11 CONTINUE
C
      DO 13 II = 1, NX0, 2
      II2          =  II + 1
      II1          =  II2/2
      APRV0(II1)    =  A00(II1)   +  CRAN2*(TWO*RANQ(ISEED2) - ONE)
      PHSPRV0(II1)  =  PHASE0(II1)  +  CRAN3*(TWO*RANQ(ISEED3) - ONE)
      ICYCL = ICYCL0
   81 IF (PHSPRV0(II1)) 82, 91, 84
   82 IF (ICYCL)  87, 87, 83
   83 ICYCL = ICYCL - 1
      PHSPRV0(II1) = PHSPRV0(II1) + THREE60
      GO TO 81
   84 IF (PHSPRV0(II1) - THREE60) 91, 85, 85
   85 IF (ICYCL)  87, 87, 86
   86 ICYCL = ICYCL - 1
      PHSPRV0(II1) = PHSPRV0(II1) - THREE60
      GO TO 84
C
C  *****   Unable to move PHSPRV0(II1) to the Principal Cycle [0,360)   *****
C
   87 WRITE(IOUT,1951)
      WRITE(IOUT,1949)
      WRITE(IOUT,1928) ICYCL0, II1, APRV0(II1), PHSPRV0(II1)
C
   91 CVPRV0(II)    = APRV0(II1)*DSIN(RAD*PHSPRV0(II1))
      CVPRV0(II2)   = APRV0(II1)*DCOS(RAD*PHSPRV0(II1))
C
   13 CONTINUE
C
C   *****    Determine the Previous End Conditions Vector ECPRV0
C                 from the Initial T0-Matrix including the Random
C                        Component Vector ECRAN0 Before Compression       *****
      WRITE(IOUT,1925)
      DO 40 JJ = 1, NZ0
```

```
      SUMZ = ZERO
      DO 41 II = 1, NX0
      SUMZ   =  SUMZ + T0(JJ,II)*CVPRV0(II)
   41 CONTINUE
      ECPRV00(JJ) = SUMZ
      ECRAN0(JJ)  = + CRAN4*(TWO*RANQ(ISEED4) - ONE)
      ECPRV0(JJ)  = SUMZ + ECRAN0(JJ)
      WRITE(IOUT,1904) JJ, ECPRV00(JJ),  ECRAN0(JJ),  ECPRV0(JJ)
   40 CONTINUE
      GO TO 60
C
C  *****   Input the Initial Previous Actual NLP Control Vector
C                 (CVPRV0) Directly Via NAMELIST Input CDATA and then
C                        Compute the Previous Control Amplitude (APRV0) and
C                                Control Phase Angle (PHSPRV0) Vectors   *****
C
   20 WRITE(IOUT,1930)
      WRITE(IOUT,1952)
      DO 49 II = 1, NX0, 2
      II2 = II + 1
      II1 = II2/2
      CALL AMPHSE(1,EPS,DEG,RAD,ITC,APRV0(II1),PHSPRV0(II1),CVPRV0(II),
     1CVPRV0(II2))
   49 CONTINUE
      GO TO 60
C
C  *****   The Initial Previous Control Amplitude (APRV0)
C                   and Phase Angle (PHSPRV0) Vectors have been
C                           Directly Input Via the NAMELIST Data CDATA   *****
C
   30 WRITE(IOUT,1953)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
C   *****   Write the Initial Previous Control Amplitude Vector
C                APRV0 and its Limits AL0 and AU0 Before Compression      *****
C
   60 WRITE(IOUT,1939)  ICASE
      WRITE(IOUT,1948)
      WRITE(IOUT,1922)
      DO 89 II = 1, NX0, 2
      II2          =  II + 1
      II1          =  II2/2
      WRITE(IOUT,1923)  II1,  AL0(II1),  APRV0(II1),  AU0(II1),
     1                  PHSPRV0(II1)
C     IF (APRV0(II1) - EPS - AL0(II1)) 14, 15, 15
C  14 APRV0(II1) = AL0(II1) + EPS
C     GO TO 17
C  15 IF (APRV0(II1) + EPS - AU0(II1)) 89, 89, 16
C  16 APRV0(II1) = AU0(II1) - EPS
C  17 WRITE(IOUT,1923)  II1,  AL0(II1),  APRV0(II1),  AU0(II1),
C    1                  PHSPRV0(II1)
   89 CONTINUE
C
C   *****   Define the Previous Actual NLP Control Vector CVPRV0 and
C                           Its Limits  CVL0 & CVU0 Before Compression   *****
C
      WRITE(IOUT,1907)
      WRITE(IOUT,1911)
      DO 50 II = 1, NX0, 2
      II2          =  II + 1
      II1          =  II2/2
      CVPRV0(II)    = APRV0(II1)*DSIN(RAD*PHSPRV0(II1))
      CVPRV0(II2)   = APRV0(II1)*DCOS(RAD*PHSPRV0(II1))
      CVL0(II)   = -AU0(II1)
      CVL0(II2) = -AU0(II1)
      CVU0(II)   =  AU0(II1)
      CVU0(II2) =  AU0(II1)
      WRITE(IOUT,1904)   II,   CVL0(II),   CVPRV0(II),   CVU0(II)
      WRITE(IOUT,1904)  II2,  CVL0(II2),  CVPRV0(II2),  CVU0(II2)
   50 CONTINUE
C
C   *****   Either the BEFORE Compression Initial Actual NLP Control Vector
C             Estimates (CV00), OR the BEFORE Compression Initial Control
```

```
C               Vector Amplitudes (A00) and Phase Angles (PHASE0) Estimates
C                 are Directly Input Via NAMELIST Data CDATA               *****
C
      WRITE(IOUT,1933)
C
      GO TO (59,69,59,69,59,69,998,998,998), IDATA
C
C  *****   Input the Initial Actual NLP Control Vector Estimates (CV00)
C              Directly Via NAMELIST Input CDATA and then Compute
C                  the Initial Control Amplitude (A00) and Initial
C                       Control Phase Angle (PHASE0) Estimated Vectors     *****
C
   59 WRITE(IOUT,1954)
C
      DO 48 II = 1, NX0, 2
      II2 = II + 1
      II1 = II2/2
      CALL AMPHSE(1,EPS,DEG,RAD,ITC,A00(II1),PHASE0(II1),CV00(II),
     1 CV00(II2))
   48 CONTINUE
C
C  *****   The Initial Control Amplitude Estimates Vector (A00) and
C                Initial Phase Angle Estimates Vector (PHASE0) have
C                      been Directly Input Via NAMELIST Data CDATA        *****
C
   69 WRITE(IOUT,1939)  ICASE
      WRITE(IOUT,1955)
C
C   *****   Adjust the Initial Control Amplitude Estimates Vector (A00)
C              to Define the Initial Control Amplitude Estimates Vector
C                  (A0) to Within Limits If Required Before Compression   *****
C
      WRITE(IOUT,1935)
      WRITE(IOUT,1927)
      DO 32 II = 1, NX0, 2
      II2         =  II + 1
      II1         =  II2/2
      A0(II1) = A00(II1)
      WRITE(IOUT,1923)  II1,  AL0(II1),  A0(II1),  AU0(II1), PHASE0(II1)
      IF (A00(II1) - EPS - AL0(II1)) 33, 34, 34
   33 A0(II1) = AL0(II1) + EPS
      GO TO 36
   34 IF (A00(II1) + EPS - AU0(II1)) 32, 32, 35
   35 A0(II1) = AU0(II1) - EPS
   36 WRITE(IOUT,1923)  II1,  AL0(II1),  A0(II1),  AU0(II1), PHASE0(II1)
   32 CONTINUE
C
C   *****   Define the Initial Actual NLP Control Vector CV0
C                Estimates and Its Limits CVL0 & CVU0 Before Compression   *****
C
      WRITE(IOUT,1934)
      WRITE(IOUT,1929)
      DO 78  II = 1, NX0, 2
      II2       =  II + 1
      II1       =  II2/2
      CV00(II)  = A00(II1)*DSIN(RAD*PHASE0(II1))
      CV00(II2) = A00(II1)*DCOS(RAD*PHASE0(II1))
   78 CONTINUE

      DO 37  II = 1, NX0
      CV0(II)   = CV00(II)
      WRITE(IOUT,1904)   II,   CVL0(II),   CV0(II),   CVU0(II)
      IF (CV0(II) - EPS - CVL0(II)) 43, 44, 44
   43 CV0(II)   = CVL0(II) + EPS
      GO TO 46
   44 IF (CV0(II) + EPS - CVU0(II)) 37, 37, 45
   45 CV0(II)   = CVU0(II) - EPS
   46 WRITE(IOUT,1904)  II,  CVL0(II),  CV0(II),  CVU0(II)
   37 CONTINUE
C
C   *****    Determine the Initial End Conditions Vector EC0 from the
C               Initial T0-Matrix and the Initial Actual NLP Control
C                    Vector Estimates (CV0) Before Before Compression;
C                        and then Write this Vector EC0, ECPRV0, and
```

```
C                             the End Conditions Weighting Coefficient
C                                  Vector WDT0 Before Compression.        *****
C
      WRITE(IOUT,1920)
      WRITE(IOUT,1956)
      DO 31 JJ = 1, NZ0
      SUMZ = ZERO
      DO 42 II = 1, NX0
      SUMZ     =  SUMZ + T0(JJ,II)*CV0(II)
   42 CONTINUE
      EC0(JJ)  = SUMZ + ECPRV0(JJ)
      WRITE(IOUT,1904)  JJ, EC0(JJ), ECPRV0(JJ), WDT0(JJ)
   31 CONTINUE
C
      WRITE(IOUT,1939)  ICASE

C
C
C  *****   Compression of the T-Matrix and the NLP Control Vector CV and
C                            Its Limits as Required and Compution of the
C                                        Ideal End Conditions Vector EC   *****
C
      WRITE(IOUT,1942)
      WRITE(IOUT,1943)   (MSAVE0(JJ),  JJ=1,NZ0/2)
      WRITE(IOUT,1941)   (MSAVE(JJ),   JJ=1,NZ0)
      WRITE(IOUT,1944)
      WRITE(IOUT,1943)   (NSAVE0(II),  II=1,NX0/2)
      WRITE(IOUT,1941)   (NSAVE(II),   II=1,NX0)
C
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C
      CALL CMPRSS(IOUT,ITOUT,NZ0,NX0,MSAVE,NSAVE,CVPRV0,CV0,CVL0,CVU0,
     1            AL0,AU0,T0,ECPRV0,EC0,WDT0,NZ,NX,CVPRV,CV,CVL,CVU,T,
     2            ECPRV,EC,WDT,AL,AU)
C
      N      = NX
      MNN2   = M + N + N + 2
C
      GO TO (99,92,92), OPTEND
C
C  *****   Constrain the Initial Estimate of the Compressed (Actual)
C                                    NLP Control Vector  CV as Required   *****
C
   92 CONTINUE
C
C   *****   Evalution of the Performance Index   *****
C
      SUMF = ZERO
C
      DO 51 JJ = 1, NZ
      SUMZ = ZERO
      DO 52 II = 1, NX
      SUMZ   =  SUMZ + T(JJ,II)*(CV(II) - CVPRV(II))
   52 CONTINUE
      EC(JJ)  = SUMZ + ECPRV(JJ)
      SUMF = SUMF + EC(JJ)*EC(JJ)*WDT(JJ)
   51 CONTINUE
      WRITE(IOUT,1921)
      WRITE(IOUT,1905)  (EC(JJ),  JJ=1,NZ)
C
      F = SUMF
      WRITE(IOUT,1909) F
C
C  *****   Define the Compressed (Actual) Control Amplitude
C                                     Vector A and Its Limits AL and AU   *****
C
      DO 53 II = 1, NX, 2
      II2 = II + 1
      II1 = II2/2
      CALL AMPHSE(1,EPS,DEG,RAD,ITC,A(II1),PHASE(II1),CV(II),CV(II2))
   53 CONTINUE
C
      WRITE(IOUT,1947)
      WRITE(IOUT,1938)
```

```
      DO 54 II = 1, NX, 2
      II2         =  II + 1
      II1         =  II2/2
      WRITE(IOUT,1923)  II1,  AL(II1),  A(II1),  AU(II1), PHASE(II1)
   54 CONTINUE
C
C   *****   Evaluation of the Constraint Functions   *****
C
      CALL CONSTR(IOUT,ITOUT,NX,MI,LSAVE,A,AL,AU,MAXASUM,MINASUM,G,G0,
     1            GMAX)
C
      IF (MI) 999, 55, 56
   55 WRITE(IOUT,1939)  ICASE
      WRITE(IOUT,1940)
      IIC  =  1 + NX/2
      WRITE(IOUT,1941)  (LSAVE(II),  II=1,IIC)
      GO TO 58
   56 WRITE(IOUT,1917)  ICASE
      WRITE(IOUT,1940)
      IIC  =  1 + NX/2
      WRITE(IOUT,1941)  (LSAVE(II),  II=1,IIC)
      WRITE(IOUT,1916)
      DO 57 II = 1, MI
      WRITE(IOUT,1904)  II,  G(II),  G0(II),  GMAX(II)
   57 CONTINUE
   58 CONTINUE
C
      GO TO (99,99,93), OPTEND
C
   93 CONTINUE
C
      GO TO (96,96,200), IOPT
C
C  *****   NLPQLP Optimisation   *****
C
   96 WRITE(IOUT,1950)  ICASE
C
      I      = 0
C
    1 CONTINUE
C
C============================================================
C
C   This is the main block to compute all function values.
C   The block is executed either for computing a steplength
C   sequentially or for approximating gradients by forward
C   differences.
C
C
C   *****   Evalution of the Performance Index   *****
C
      SUMF = ZERO
C
      DO 61 JJ = 1, NZ
      SUMZ = ZERO
      DO 62 II = 1, NX
      SUMZ   =  SUMZ + T(JJ,II)*(CV(II) - CVPRV(II))
   62 CONTINUE
      EC(JJ)  = SUMZ + ECPRV(JJ)
      SUMF = SUMF + EC(JJ)*EC(JJ)*WDT(JJ)
   61 CONTINUE
C
      F = SUMF
C
C  *****   Define the Compressed (Actual) Control Amplitude Vector A and
C                                                Its Limits AL and AU   *****
C
      DO 63 II = 1, NX, 2
      II2 = II + 1
      II1 = II2/2
      CALL AMPHSE(1,EPS,DEG,RAD,ITC,A(II1),PHASE(II1),CV(II),CV(II2))
   63 CONTINUE
C
C   *****   Evaluation of the Constraint Functions   *****
```

```
C
      CALL CONSTR(IOUT,ITOUT,NX,MI,LSAVE,A,AL,AU,MAXASUM,MINASUM,G,G0,
     1            GMAX)
C
C
C============================================================
C
C
      NFUNC = NFUNC + 1
      IF (IFAIL.EQ.-1) GOTO 4
      IF (I.GT.0) GOTO 3
    2 CONTINUE
      FBCK = F
      DO J=1,M
         GBCK(J) = G(J)
      ENDDO
      I = 0
    5 I = I + 1
      EPSREL = EPS*DMAX1(1.0D0,DABS(CV(I)))
      CV(I) = CV(I) + EPSREL
      GOTO 1
    3 CONTINUE
      DF(I) = (F - FBCK)/EPSREL
      DO J=1,M
         DG(J,I) = (G(J) - GBCK(J))/EPSREL
      ENDDO
      CV(I) = CV(I) - EPSREL
   IF (I.LT.N) GOTO 5
      F = FBCK
      DO J=1,M
         G(J) = GBCK(J)
      ENDDO
C
    4 CONTINUE
C
C
      CALL NLPQLP (     L,      M,     ME,   MMAX,      N,
     /              NMAX,   MNN2,     CV,      F,      G,
     /                DF,     DG,      U,    CVL,    CVU,
     /                 C,      D,    ACC,  ACCQP, STPMIN,
     /            MAXFUN,  MAXIT,  MAXNM,   RHOB, IPRINT,
     /              MODE,   IOUT,  IFAIL,     WA,    LWA,
     /               KWA,   LKWA, ACTIVE, LACTIV,    LQL,
     /                QL)
C
C
      WRITE(IOUT,1912)
C
C
      IF (IFAIL.EQ.-1) GOTO 1
      IF (IFAIL.EQ.-2) GOTO 2
C
C   *****   Write Number of Function Evaluations   *****
C
      WRITE(IOUT,1000) NFUNC
C
C  *****   Write the Solution Control Vector   *****
C
      WRITE(IOUT,1903)  ICASE
      WRITE(IOUT,1937)
      DO 71 II = 1, NX
      DCV(II) = CV(II) - CV0(II)
      WRITE(IOUT,1923)  II,  CVL(II),  CV(II),  CVU(II), DCV(II)
   71 CONTINUE
C
C   *****   Evalution of the Performance Index   *****
C
C
C
C   ********************
C
      WRITE(IOUT,1901)
      WRITE(IOUT,1945)
      WRITE(IOUT,1905)  (EC(JJ),  JJ=1,NZ)
```

```
      WRITE(IOUT,1906)  F
      WRITE(IOUT,1901)
C
C   *******************
C
C
      SUMF   = ZERO
      WRITE(IOUT,1945)
C
      DO 72 JJ = 1, NZ
      SUMZ   = ZERO
      DO 73 II = 1, NX
      SUMZ   =  SUMZ + T(JJ,II)*(CV(II) - CVPRV(II))
   73 CONTINUE
      EC(JJ) = SUMZ + ECPRV(JJ)
      SUMF   = SUMF + EC(JJ)*EC(JJ)*WDT(JJ)
   72 CONTINUE
      WRITE(IOUT,1905)  (EC(JJ),  JJ=1,NZ)
C
      F = SUMF
      WRITE(IOUT,1906)  F
      WRITE(IOUT,1901)
C
C   *******************
C
C
C
C  *****   Define the Compressed (Actual) Control Amplitude
C                                    Vector A and Its Limits AL and AU   *****
C
      DO 74 II = 1, NX, 2
      II2 = II + 1
      II1 = II2/2
      CALL AMPHSE(1,EPS,DEG,RAD,ITC,A(II1),PHASE(II1),CV(II),CV(II2))
   74 CONTINUE
C
      WRITE(IOUT,1936)
      WRITE(IOUT,1938)
      DO 75 II = 1, NX, 2
      II2 = II + 1
      II1 = II2/2
      WRITE(IOUT,1923)  II1,  AL(II1),  A(II1),  AU(II1), PHASE(II1)
   75 CONTINUE
C
C   *****   Evaluation of the Constraint Functions   *****
C
      CALL CONSTR(IOUT,ITOUT,NX,MI,LSAVE,A,AL,AU,MAXASUM,MINASUM,G,G0,
     1            GMAX)
C
      IF (MI) 999, 76, 77
   76 WRITE(IOUT,1946)  ICASE
      GO TO 79
   77 WRITE(IOUT,1918)  ICASE
      WRITE(IOUT,1940)
      WRITE(IOUT,1941)  (LSAVE(II),  II=1,IIC)
      WRITE(IOUT,1916)
      DO 80 II = 1, MI
      WRITE(IOUT,1904)  II,  G(II),  G0(II),  GMAX(II)
   80 CONTINUE
   79 CONTINUE
C
C
      GO TO (97,200,200), IOPT
C
C
C  *****   Solve the Regulator Problem   *****
C
  200 CONTINUE
C
      WRITE(IOUT,2950)  ICASE
C
C   *****   Write Alpha and the Weighting Vectors   *****
C
      NXXX  =  NX
```

```
      NZZZ  =  NZ
C
      WRITE(IOUT,2964)  ALPHA
      WRITE(IOUT,2965)
      WRITE(IOUT,2928)  NZZZ, (WZ(I), I=1,NZZZ)
      WRITE(IOUT,2966)
      WRITE(IOUT,2928)  NXXX, (WX(I), I=1,NXXX)
      WRITE(IOUT,2967)
      WRITE(IOUT,2928)  NXXX, (WDX(I), I=1,NXXX)
C
C  *****   Compute Regulator Problem Solution Control Vector   *****
C
      CALL DIAGW(NZZZ,WZ,WZZ)
C
C  *****   061   *****   061   *****   061   *****   061   *****
C
C     WRITE(IOUT,3961)
C     DO 301 I = 1, NZZZ
C     WRITE(IOUT,2928)  I ,  (WZZ(I,J), J=1,NZZZ)
C 301 CONTINUE
C
      CALL DIAGW(NXXX,WX,WXX)
C
C  *****   062   *****   062   *****   062   *****   062   *****
C
C     WRITE(IOUT,3962)
C     DO 302 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (WXX(I,J), J=1,NXXX)
C 302 CONTINUE
C
      CALL DIAGW(NXXX,WDX,WDXX)
C
C  *****   063   *****   063   *****   063   *****   063   *****
C
C     WRITE(IOUT,3963)
C     DO 303 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (WDXX(I,J), J=1,NXXX)
C 303 CONTINUE
C
      DO 202 I = 1, NZZZ
      DO 201 J = 1, NXXX
      TT(I,J)  = T(I,J)
  201 CONTINUE
  202 CONTINUE
C
C  *****   064   *****   064   *****   064   *****   064   *****
C
      WRITE(IOUT,3964)
      DO 304 I = 1, NZZZ
      WRITE(IOUT,2928)  I ,  (TT(I,J), J=1,NXXX)
  304 CONTINUE
C
      CALL TRNSP(NZZZ,NXXX,TT,TTT)
C
C  *****   065   *****   065   *****   065   *****   065   *****
C
C     WRITE(IOUT,3965)
C     DO 305 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (TTT(I,J), J=1,NZZZ)
C 305 CONTINUE
C
      CALL MMULT(NXXX,NZZZ,NZZZ,TTT,WZZ,DUMXZ)
C
C  *****   066   *****   066   *****   066   *****   066   *****
C
C     WRITE(IOUT,3966)
C     DO 306 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (DUMXZ(I,J), J=1,NZZZ)
C 306 CONTINUE
C
      CALL MMULT(NXXX,NZZZ,NXXX,DUMXZ,TT,DUMXX)
C
C  *****   067   *****   067   *****   067   *****   067   *****
C
```

```
C     WRITE(IOUT,3967)
C     DO 307 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (DUMXX(I,J), J=1,NXXX)
C 307 CONTINUE
C
      CALL SMDFF(NXXX,NXXX,1,DUMXX,WDXX,DUMTT)
C
C  *****   068   *****   068   *****   068   *****   068   *****
C
      DO 321 I = 1,NXXX
      DO 320 J = 1,NXXX
      DUMXX(I,J) = DUMTT(I,J)
  320 CONTINUE
  321 CONTINUE
C
C     WRITE(IOUT,3968)
C     DO 308 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (DUMXX(I,J), J=1,NXXX)
C 308 CONTINUE
C
C  *****   Compute the Matrix [DUMXX1] to be Inverted   *****
C
      CALL SMDFF(NXXX,NXXX,1,DUMXX,WXX,DUMXX1)
C
      WRITE(IOUT,2960)
      DO 203 I = 1, NXXX
      WRITE(IOUT,2928)  I ,  (DUMXX1(I,J), J=1,NXXX)
  203 CONTINUE
C
C  *****   Initialise Matrices [DD]. [EE], and [FF] to Zero   *****
C
      DO 220 I = 1, NXX
      IPIV(I)  = IZERO
      DO 219 J = 1, NXX
      DD(I,J)  = ZERO
      EE(I,J)  = ZERO
      FF(I,J)  = ZERO
  219 CONTINUE
  220 CONTINUE
C
C  *****   Define Matrix [DD] to be equal to Matrix [DUMXX1]   *****
C
      DO 210 I = 1, NXXX
      DO 209 J = 1, NXXX
      DD(I,J)  = DUMXX1(I,J)
  209 CONTINUE
  210 CONTINUE
C
      CALL DGETRF(NXXX,NXXX,DD,NXXX,IPIV,INFO)
      CALL DGETRI(NXXX,DD,NXXX,IPIV,WORK,LWORK,INFO)
C
       IF(INFO) 412, 411, 413
C
C  *****   Matrix Inversion was Successful   *****
C
  411 WRITE(IOUT,2970)
      GO TO 414
C
C  *****   Matrix Inversion Failed. An Element had an Illegal Value.
C                                                                 *****
C
  412 INFO = -INFO
      WRITE(IOUT,2973)  INFO
      GO TO 98
C
C  *****   Matrix Inversion Failed. An Element on the Diagonal is Equal
C          to Zero and correspondingly the Matrix is Singular and its
C          Inverse could not be computed.
C                                                                 *****
C
  413 WRITE(IOUT,2974) INFO
      GO TO 98
C
  414 CONTINUE
```

```
C
C  *****   Matrix [DD] = The Inverted Matrix  =  [DUMXX1]-1   *****
C
      WRITE(IOUT,2961)
      DO 204 I = 1, NXXX
      WRITE(IOUT,2928)  I ,  (DD(I,J), J=1,NXXX)
  204 CONTINUE
C
C   *****   Matrix [EE] = The Identity Matrix = [DUMXX1][DD]   *****
C
      CALL MMULT(NXXX,NXXX,NXXX,DUMXX1,DD,EE)
C
      WRITE(IOUT,2962)
      DO 205 I = 1, NXX
      WRITE(IOUT,2928)  I ,  (EE(I,J), J=1,NXX)
  205 CONTINUE
C
C   *****   Matrix [FF] = The Identity Matrix = [DD][DUMXX1]   *****
C
      CALL MMULT(NXXX,NXXX,NXXX,DD,DUMXX1,FF)
C
      WRITE(IOUT,2963)
      DO 206 I = 1, NXX
      WRITE(IOUT,2928)  I ,  (FF(I,J), J=1,NXX)
  206 CONTINUE
C
C  *****   Compute the Solution Control Vector (the Theta Vector)   *****
C
      CALL MMULT(NXXX,NXXX,NXXX,DD,DUMXX,DUMTT)
      DO 323 I = 1,NXXX
      DO 322 J = 1,NXXX
      DUMXX(I,J) = DUMTT(I,J)
  322 CONTINUE
  323 CONTINUE
C
C  *****   069   *****   069   *****   069   *****   069   *****
C
C     WRITE(IOUT,3969)
C     DO 309 I = 1, NXXX
C     WRITE(IOUT,2928)  I ,  (DUMXX(I,J), J=1,NXXX)
C 309 CONTINUE
C
      DO 207 J = 1, NXXX
      DUMX(J,1) = CVPRV(J)
  207 CONTINUE
C
C  *****   070   *****   070   *****   070   *****   070   *****
C
C     WRITE(IOUT,3970)
C     DO 310 J = 1, NXXX
C     WRITE(IOUT,2928)  J ,  DUMX(J,1)
C 310 CONTINUE
C
      CALL MMULT(NXXX,NXXX,1,DUMXX,DUMX,DUMT)
C
C  *****   071   *****   071   *****   071   *****   071   *****
C
      DO 324 J = 1,NXXX
      DUMX(J,1) = DUMT(J,1)
  324 CONTINUE
C
C     WRITE(IOUT,3971)
C     DO 311 J = 1, NXXX
C     WRITE(IOUT,2928)  J ,  DUMX(J,1)
C 311 CONTINUE
C
      DO 208 I = 1, NZZZ
      DUMZ(I,1) = ECPRV(I)
  208 CONTINUE
C
C  *****   072   *****   072   *****   072   *****   072   *****
C
C     WRITE(IOUT,3972)
C     DO 312 I = 1, NZZZ
```

```
C     WRITE(IOUT,2928)  I ,  DUMZ(I,1)
C 312 CONTINUE
C
      CALL MMULT(NXXX,NZZZ,1,DUMXZ,DUMZ,DUMX1)
C
C  *****   073   *****   073   *****   073   *****   073   *****
C
C     WRITE(IOUT,3973)
C     DO 313 J = 1, NXXX
C     WRITE(IOUT,2928)  J ,  DUMX1(J,1)
C 313 CONTINUE
C
      CALL MMULT(NXXX,NXXX,1,DD,DUMX1,DUMT1)
C
C  *****   074   *****   074   *****   074   *****   074   *****
      DO 325 J = 1,NXXX
      DUMX1(J,1) = DUMT1(J,1)
  325 CONTINUE
C
C     WRITE(IOUT,3974)
C     DO 314 J = 1, NXXX
C     WRITE(IOUT,2928)  J ,  DUMX1(J,1)
C 314 CONTINUE
C
      CALL SLMULT(ALPHA,NXXX,1,DUMX1,DUMT1)
C
C  *****   075   *****   075   *****   075   *****   075   *****
C
      DO 326 J = 1,NXXX
      DUMX1(J,1) = DUMT1(J,1)
  326 CONTINUE
C
C     WRITE(IOUT,3975)
C     DO 315 J = 1, NXXX
C     WRITE(IOUT,2928)  J ,  DUMX1(J,1)
C 315 CONTINUE
C
      CALL SMDFF(NXXX,1,0,DUMX,DUMX1,DUMT)
C
C  *****   Write the Solution Control Vector (the Theta-Vector)   *****
C
      DO 327 J = 1,NXXX
      DUMX(J,1) = DUMT(J,1)
      THETA(J) = DUMX(J,1)
  327 CONTINUE
C
      WRITE(IOUT,2968)
      DO 318 II = 1, NXXX, 2
      II2 = II + 1
      II1 = II2/2
      CALL AMPHSE(1,EPS,DEG,RAD,ITC,AMP(II1),PHS(II1),THETA(II),
     1            THETA(II2))
      WRITE(IOUT,2975)   II,  THETA(II)
      WRITE(IOUT,2976)  II2,  THETA(II2), II1, AMP(II1), PHS(II1)
  318 CONTINUE
      GO TO (998,332,334), IOPT
  332 WRITE(IOUT,2969)
      DO 333 II = 1, NXXX, 2
      II2 = II + 1
      II1 = II2/2
      WRITE(IOUT,2975)   II,  CV(II)
      WRITE(IOUT,2976)  II2,  CV(II2), II1, A(II1), PHASE(II1)
  333 CONTINUE
C
C  *****   Compute the Solution Measurement Vector (the Z-Vector)   *****
C
  334 DO 330 J = 1,NXXX
      DUMX(J,1) = THETA(J) - CVPRV(J)
  330 CONTINUE
C
      CALL MMULT(NZZZ,NXXX,1,TT,DUMX,DUMZ)
C
C  *****   076   *****   076   *****   076   *****   076   *****
C
```

```
      DO 331 J = 1, NZZZ
      DUMX(J,1) = ECPRV(J)
  331 CONTINUE
C
C     WRITE(IOUT,3976)
C     DO 316 J = 1, NZZZ
C     WRITE(IOUT,2928)  J ,  DUMZ(J,1)
C 316 CONTINUE
C
      CALL SMDFF(NZZZ,NZZZ,1,DUMZ,DUMX,DUMT)
C
C  *****   Write the Solution Measurement Vector (the Z-Vector)   *****
C
      DO 328 J = 1,NZZZ
      ZZ(J) = DUMT(J,1)
  328 CONTINUE
C
      WRITE(IOUT,2977)
      DO 335  I = 1, NZZZ
      WRITE(IOUT,2975)  I, ZZ(I)
  335 CONTINUE
C
C   *****   Compute the Corresponding Performance Index   *****
C
      DO 319 J = 1, NZZZ
      DUMZ(J,1) = ZZ(J)
  319 CONTINUE
C
      CALL MMULT(NZZZ,NZZZ,1,WZZ,DUMZ,DUMQ)
C
C  *****   077   *****   077   *****   077   *****   077   *****
C
      DO 329 J = 1,NZZZ
      DUMZ(J,1) = DUMQ(J,1)
  329 CONTINUE
C
C     WRITE(IOUT,3977)
C     DO 317 J = 1, NXXX
C     WRITE(IOUT,2928)  J ,  DUMZ(J,1)
C 317 CONTINUE
C
      CALL TRNSP(NZZZ,1,DUMZ,DUMZT)
C
C  *****   078   *****   078   *****   078   *****   078   *****
C
C     WRITE(IOUT,3978)
C     I = 1
C     WRITE(IOUT,2928)  I ,  (DUMZT(1,J), J=1,NZZZ)
C
      CALL MMULT(1,NZZZ,1,DUMZT,DUMZ,JJJ)
C
C   *****   Write the Corresponding Performance Index   *****
C
      WRITE(IOUT,2906) JJJ(1,1)
C
C   *****   End of Regulator Problem   *****
C
      GO TO 97
C
C   *****   ERROR End of Case   *****
C
  998 WRITE(IOUT,1924), IDATA
      GO TO 98
C
C   *****   End of Case   *****
C
C
   97 IF (CVOUT .LE. 0) GO TO 98
      GO TO (94,94,95), IOPT
   94 WRITE(IOUT,1931) NXXX
      WRITE(IOUT,1932) (CV(II), II=1,NXXX)
      GO TO (98,95,95), IOPT
   95 WRITE(IOUT,2931) NXXX
      WRITE(IOUT,1932) (THETA(II), II=1,NXXX)
```

```
      GO TO (98,371,98), IOPT
  371 RSSDCV = ZERO
      DO 372 II = 1, NXXX
      DELTCV(II) = CV(II) - THETA(II)
      RSSDCV = RSSDCV + DELTCV(II)*DELTCV(II)
  372 CONTINUE
      RSSDCV = DSQRT(RSSDCV)
      WRITE(IOUT,2971) RSSDCV
      WRITE(IOUT,2972) (DELTCV(II), II=1,NXXX)
   98 CVOUT = 0
   99 WRITE(IOUT,1908)  ICASE
C
      IF (MULT .LE. 0)  GO TO 999
      MULT  = 0
      WRITE(IOUT,1901)
      ICASE = ICASE + 1
      GO TO 100
  999 STOP
C
      END
C
C
C
C
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C
```

D.2 CMPRSS Subroutine Code

The purpose of the CMPRSS Subroutine Code is to reduce the dimension of the control vector and/or end conditions vector if and as required.

```
C
      SUBROUTINE CMPRSS(IOUT,ITOUT,NZ0,NX0,MSAVE,NSAVE,CVPRV0,CV0,CVL0,
     1                  CVU0,AL0,AU0,T0,ECPRV0,EC0,WDT0,NZ,NX,CVPRV,CV,
     2                  CVL,CVU,T,ECPRV,EC,WDT,AL,AU)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
C
C  ****   Control Vector, T-Matrix, and End Conditions Vector Compression   *****
C
C  *****   These statements establish and define the Dimensions and
C                Type of some of the Non-linear Programming Parameters.   *****
C
      INTEGER           NMAX, MMAX, MNN2X, LWA, LKWA, LACTIV
C
      PARAMETER (       NMAX   = 14,
     /                  MMAX   = 10,
     /                  MNN2X  = MMAX + NMAX + NMAX + 2,
     /                  LWA    = 1.5*NMAX*NMAX + 33*NMAX + 9*MMAX + 200,
     /                  LKWA   = NMAX + 20,
     /                  LACTIV = 2*MMAX + 10)
C
C
C  *****   These statements establish and define the Dimensions and
C                Type of some of the NLP10x10 Main Driver Parameters.   *****
C
      INTEGER*4  NCC,   NXX,   NXZ,   NZNX,   NZZ
C
      PARAMETER  (NCC=6,NXX=10,NZZ=10,NZNX=NZZ*NXX,NXZ=MAX(NXX,NZZ))
C
C
C  *****   These statements establish and define the Dimensions and
C          Type of some of the CMPRSS Compression Routine Parameters.   *****
C
      INTEGER*4  I,       IOUT,    ITOUT,   J,       JJ,      JRM,
     1           K        L,       LL,      MSAVE(NZZ),
     2           NSAVE(NXX),       NX,      NX0,     NZ,      NZ0
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      REAL*8  AL(NCC),       AL0(NCC),      AU(NCC),       AU0(NCC),
     1        CV(NMAX),      CV0(NMAX),     CVL(NMAX),     CVL0(NMAX),
     2        CVPRV(NMAX),   CVPRV0(NMAX),  CVU(NMAX),     CVU0(NMAX),
     3        EC(NZZ),       EC0(NZZ),      ECPRV(NZZ),    ECPRV0(NZZ),
     4        ECPRVT(NZZ),   ECPRVTT(NZZ),  ECT(NZZ),      SUM,
     5        T(NZZ,NXX),    T0(NZZ,NXX),   TT(NZZ,NXX),   WDT(NZZ),
     6        WDT0(NZZ),     WDTT(NZZ)
C
C
C  *****   Data Values Statement   *****
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      DATA    ECT,               TT                                  /
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
     o        NZZ*0.0D+00,       NZNX*0.0D+00                        /
C
C
C
C  *****   Format Statements   *****
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 1010 FORMAT(//2X,53H *****   Initial Control Vector (CV0), T-Matrix (T0
     1),/43X,37H and Measurement Vector (EC0)   *****/)
 1011 FORMAT(/2X,45H *****   Initial Control Vector (CV0)   *****)
 1012 FORMAT(/3X,4H CV0,1X,4D18.8/(8X,4D18.8))
 1013 FORMAT(/2X,53H *****   Initial Greatest Least Bounds (CVL0) for th
     1e/51X,29H Control Vector (CV0)   *****)
 1014 FORMAT(/2X,5H CVL0,1X,4D18.8/(8X,4D18.8))
 1015 FORMAT(/2X,50H *****   Initial Least Upper Bounds (CVU0) for the/
     1 51X,29H Control Vector (CV0)   *****)
```

```
 1016 FORMAT(/2X,5H CVU0,1X,4D18.8/(8X,4D18.8))
 1017 FORMAT(//4X,4H Row,9X,38H *****   Initial T-Matrix (T0)   *****)
 1018 FORMAT(/1X,3H T0,I3,1X,4D18.8/(8X,4D18.8))
 1019 FORMAT(//2X,58H *****   Initial T-Matrix (T0) Output is Supressed
     1  *****)
 1020 FORMAT(//2X,52H *****   Initial End Conditions Vector (EC0)   ****
     1*)
 1021 FORMAT(/3X,4H EC0,1X,4D18.8/(8X,4D18.8))
 1022 FORMAT(//2X,56H *****   Initial Control Vector (CVPRV0) and Measur
     1ement/27X,53H Vector (ECPRV0) from the Previous Duty Cycle   *****
     2/)
 1023 FORMAT(/2X,63H *****   Initial Previous Cycle Control Vector (CVPR
     1V0)   *****)
 1024 FORMAT(/7H CVPRV0,1X,4D18.8/(8X,4D18.8))
 1025 FORMAT(//2X,70H *****   Initial Previous Cycle End Conditions Vect
     1or (ECPRV0)   *****)
 1026 FORMAT(/7H ECPRV0,1X,4D18.8/(8X,4D18.8))
 1027 FORMAT(//2X,56H *****   Initial CV0, T0, EC0, CVPRV0, and ECPRV0 O
     1utput/58X,22H is Suppressed   *****/)
 1071 FORMAT(/2X,60H *****   Initial End Conditions Weighting Coefficien
     1t Vector/36X,44H (WDT0) for the Performance Index (F)  *****)
 1072 FORMAT(/2X,5H WDT0,1X,4D18.8/(8X,4D18.8))
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 1030 FORMAT(//2X,57H *****   Intermediate Control Vector (CV), T-Matrix
     1 (TT),/15X,65H and Measurement Vector (ECT) after the First Compre
     2ssion   *****/)
 1031 FORMAT(/2X,77H *****   Intermediate Control Vector (CV) after the
     1First Compression   *****)
 1032 FORMAT(/4X,3H CV,1X,4D18.8/(8X,4D18.8))
 1033 FORMAT(/2X,57H *****   Intermediate Greatest Least Bounds (CVL) fo
     1r the/52X,28H Control Vector (CV)   *****)
 1034 FORMAT(/3X,4H CVL,1X,4D18.8/(8X,4D18.8))
 1035 FORMAT(/2X,54H *****   Intermediate Least Upper Bounds (CVU) for t
     1he/52X,28H Control Vector (CV)   *****)
 1036 FORMAT(/3X,4H CVU,1X,4D18.8/(8X,4D18.8))
 1037 FORMAT(//4X,4H Row,7X,43H *****   Intermediate T-Matrix (TT)   ***
     1**)
 1038 FORMAT(/1X,3H TT,I3,1X,4D18.8/(8X,4D18.8))
 1039 FORMAT(//2X,63H *****   Intermediate T-Matrix (TT) Output is Supre
     1ssed   *****)
 1040 FORMAT(//2X,57H *****   Intermediate End Conditions Vector (ECT)
     1 *****)
 1041 FORMAT(/3X,4H ECT,1X,4D18.8/(8X,4D18.8))
 1042 FORMAT(//2X,67H *****   Intermediate Control Vector (CVPRV) and Me
     1asurement Vector/26X,44H (ECPRV0) from the Previous Duty Cycle aft
     2er/50X,30H the First Compression   *****/)
 1043 FORMAT(/2X,65H *****   Intermediate Previous Cycle Control Vector
     1(CVPRV) after/50X,30H the First Compression   *****)
 1044 FORMAT(/1X,6H CVPRV,1X,4D18.8/(8X,4D18.8))
 1045 FORMAT(//2X,68H *****   Intermediate Previous Cycle End Conditions
     1 Vector (ECPRVTT)/44X,36H after the First Compression   *****)
 1046 FORMAT(/7H ECPRVT,1X,4D18.8/(8X,4D18.8))
 1047 FORMAT(//2X,58H *****   Intermediate CV, TT, ECT, CVPRV, and ECPRV
     1T after/29X,51H the First Compression Output is Suppressed   *****
     2/)
 1081 FORMAT(/2X,65H *****   Intermediate End Conditions Weighting Coeff
     1icient Vector/36X,44H (WDTT) for the Performance Index (F)  *****)
 1082 FORMAT(/2X,5H WDTT,1X,4D18.8/(8X,4D18.8))
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 1050 FORMAT(//2X,53H *****   Final Control Vector (CV), T-Matrix (T), a
     1nd/19X,61H Measurement Vector (EC) after the Second Compression
     2*****/)
 1051 FORMAT(/2X,71H *****   Final Control Vector (CV) after the Second
     1Compression   *****)
 1052 FORMAT(/4X,3H CV,1X,4D18.8/(8X,4D18.8))
 1053 FORMAT(/2X,51H *****   Greatest Least Bounds (CVL) Vector for the/
     152X,28H Control Vector (CV)   *****)
 1054 FORMAT(/3X,4H CVL,1X,4D18.8/(8X,4D18.8))
 1055 FORMAT(/2X,76H *****   Least Upper Bounds (CVU) Vector for the Con
     1trol Vector (CV)   *****)
```

```
 1056 FORMAT(/3X,4H CVU,1X,4D18.8/(8X,4D18.8))
 1057 FORMAT(//4X,4H Row,11X,35H *****   Final T-Matrix (T)   *****)
 1058 FORMAT(/2X,2H T,I3,1X,4D18.8/(8X,4D18.8))
 1059 FORMAT(//2X,55H *****   Final T-Matrix (T) Output is Supressed   *
     1****)
 1060 FORMAT(//2X,78H *****   Final End Conditions Vector (EC) after the
     1 Second Compression   *****)
 1061 FORMAT(/4X,3H EC,1X,4D18.8/(8X,4D18.8))
 1062 FORMAT(//2X,60H *****   Final Control Vector (CVPRV) and Measureme
     1nt Vector/24X,43H (ECPRV) from the Previous Duty Cycle after/49X
     2,31H the Second Compression   *****/)
 1063 FORMAT(/2X,58H *****   Final Previous Cycle Control Vector (CVPRV)
     1 after/49X,31H the Second Compression   *****)
 1064 FORMAT(/1X,6H CVPRV,1X,4D18.8/(8X,4D18.8))
 1065 FORMAT(//2X,59H *****   Final Previous Cycle End Conditions Vector
     1 (ECPRV)/43X,37H after the Second Compression   *****)
 1066 FORMAT(/1X,6H ECPRV,1X,4D18.8/(8X,4D18.8))
 1067 FORMAT(//2X,48H *****   Final CV, T, EC, CVPRV, and ECPRV after/
     1 28X,52H the Second Compression Output is Suppressed   *****/)
 1091 FORMAT(/2X,58H *****   Final End Conditions Weighting Coefficient
     1Vector/37X,43H (WDT) for the Performance Index (F)  *****)
 1092 FORMAT(/3X,4H WDT,1X,4D18.8/(8X,4D18.8))
C
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
C
 1990 FORMAT(//1X,79H *******************   T-Matrix Compression is Init
     1iated   ********************/)
 1998 FORMAT(19X,39H   *****   ERROR STOP in CMPRSS   *****/)
 1999 FORMAT(//1X,79H *******************   T-Matrix Compression is Comp
     1leted   ********************)
C
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
C
C
C  *****   Execute CMPRSS   *****
C
C  *****   Initial Control Vector (CV0), T-Matrix (T0), and Measurement
C                Vector (EC0); and Initial Control Vector (CVPRV0) and
C                      Measurement Vector from the Previous Duty Cycle   *****
C
      WRITE(IOUT,1990)
C
      DO 11 K = 1, NZ0
      SUM        = 0.0
      DO 12    L = 1, NX0
      SUM        = SUM    + T0(K,L)*CV0(L)
   12 CONTINUE
      EC0(K)     = SUM + ECPRV0(K)
   11 CONTINUE
C
      IF (IABS(ITOUT) - 2) 18, 13, 13
C
   13 WRITE(IOUT,1010)
      WRITE(IOUT,1011)
      WRITE(IOUT,1012) (CV0(L), L=1,NX0)
      WRITE(IOUT,1013)
      WRITE(IOUT,1014) (CVL0(L), L=1,NX0)
      WRITE(IOUT,1015)
      WRITE(IOUT,1016) (CVU0(L), L=1,NX0)
      IF (ITOUT) 16, 14, 14
   14 WRITE(IOUT,1017)
      DO 15 K = 1, NZ0
      WRITE(IOUT,1018) K, (T0(K,L), L=1,NX0)
   15 CONTINUE
      GO TO 17
   16 WRITE(IOUT,1019)
   17 WRITE(IOUT,1020)
      WRITE(IOUT,1021) (EC0(K), K=1,NZ0)
      WRITE(IOUT,1071)
      WRITE(IOUT,1072) (WDT0(K), K=1,NZ0)
C
      WRITE(IOUT,1022)
      WRITE(IOUT,1023)
      WRITE(IOUT,1024) (CVPRV0(L), L=1,NX0)
```

```
      WRITE(IOUT,1025)
      WRITE(IOUT,1026) (ECPRV0(K), K=1,NZ0)
C
      GO TO 20
C
   18 WRITE(IOUT,1027)
C
C  *****   Compression of the Control Vector and the T-Matrix
C                                              (First Compression)   *****
C
   20 NX         = 0
      J          = 1
C
      DO 21   L = 1, NX0
      LL         = (L + 1)/2
      IF (NSAVE(L)) 21, 21, 23
   23 NX         = J
      CV(J)      = CV0(L)
      CVPRV(J)   = CVPRV0(L)
      CVL(J)     = CVL0(L)
      CVU(J)     = CVU0(L)
C
      JRM        = MOD(J,2)
      IF (JRM) 998, 24, 25
   24 JJ         = (J + 1)/2
      AL(JJ)     = AL0(LL)
      AU(JJ)     = AU0(LL)
C
   25 DO 22   K = 1, NZ0
      TT(K,J)    = T0(K,L)
   22 CONTINUE
C
      J          = J + 1
C
   21 CONTINUE
C
C  *****   Measurement Vector, T-Matrix, and Control Vector
C                                       after the First Compression   *****
C
      DO 31    K = 1, NZ0
      SUM         = 0.0
      DO 32    J = 1, NX
      SUM         = SUM    + TT(K,J)*CV(J)
   32 CONTINUE
      ECPRVTT(K) = ECPRV0(K)
      ECT(K)      = SUM + ECPRVTT(K)
      WDTT(K)     = WDT0(K)
   31 CONTINUE
C
      IF (IABS(ITOUT) - 3) 38, 33, 33
C
   33 WRITE(IOUT,1030)
      WRITE(IOUT,1031)
      WRITE(IOUT,1032) (CV(J), J=1,NX)
      WRITE(IOUT,1033)
      WRITE(IOUT,1034) (CVL(J), J=1,NX)
      WRITE(IOUT,1035)
      WRITE(IOUT,1036) (CVU(J), J=1,NX)
      IF (ITOUT) 36, 34, 34
   34 WRITE(IOUT,1037)
      DO 35 K = 1, NZ0
      WRITE(IOUT,1038) K, (TT(K,J), J=1,NX)
   35 CONTINUE
      GO TO 37
   36 WRITE(IOUT,1039)
   37 WRITE(IOUT,1040)
      WRITE(IOUT,1041) (ECT(K), K=1,NZ0)
      WRITE(IOUT,1081)
      WRITE(IOUT,1082) (WDTT(K), K=1,NZ0)
C
      WRITE(IOUT,1042)
      WRITE(IOUT,1043)
      WRITE(IOUT,1044) (CVPRV(J), J=1,NX)
      WRITE(IOUT,1045)
```

```
      WRITE(IOUT,1046) (ECPRVTT(K), K=1,NZ0)
C
      GO TO 40
C
   38 WRITE(IOUT,1047)
C
C
C  *****   Compression of the Measurement Vector and the T-Matrix
C                                  (Second and Final Compression)   *****
C
   40 NZ         = 0
         I       = 1
C
      DO 41   K = 1, NZ0
      IF (MSAVE(K)) 41, 41, 43
   43 NZ         = I
      WDT(I)     = WDTT(K)
      ECPRVT(I) = ECPRVTT(K)
C
      DO 42  J  = 1, NX
      T(I,J)     = TT(K,J)
   42 CONTINUE
C
      I          = I + 1
C
   41 CONTINUE
C
C  *****   Measurement Vector, T-Matrix, and Control Vector
C                          after the Second and Final Compression   *****
C
      DO 51    I = 1, NZ
      SUM        = 0.0
      DO 52    J = 1, NX
      SUM        = SUM    + T(I,J)*CV(J)
   52 CONTINUE
      ECPRV(I)   = ECPRVT(I)
      EC(I)      = SUM + ECPRV(I)
   51 CONTINUE
C
      IF (IABS(ITOUT) - 1) 58, 53, 53
C
   53 WRITE(IOUT,1050)
      WRITE(IOUT,1051)
      WRITE(IOUT,1052) (CV(J), J=1,NX)
      WRITE(IOUT,1053)
      WRITE(IOUT,1054) (CVL(J), J=1,NX)
      WRITE(IOUT,1055)
      WRITE(IOUT,1056) (CVU(J), J=1,NX)
      IF (ITOUT) 56, 54, 54
   54 WRITE(IOUT,1057)
      DO 55 I= 1, NZ
      WRITE(IOUT,1058) I, (T(I,J), J=1,NX)
   55 CONTINUE
      GO TO 57
   56 WRITE(IOUT,1059)
   57 WRITE(IOUT,1060)
      WRITE(IOUT,1061) (EC(I), I=1,NZ)
      WRITE(IOUT,1091)
      WRITE(IOUT,1092) (WDT(I), I=1,NZ)
C
      WRITE(IOUT,1062)
      WRITE(IOUT,1063)
      WRITE(IOUT,1064) (CVPRV(J), J=1,NX)
      WRITE(IOUT,1065)
      WRITE(IOUT,1066) (ECPRV(I), I=1,NZ)
      GO TO 999
C
   58 WRITE(IOUT,1067)
      GO TO 999
C
C  *****   Error Exit   *****
C
  998 WRITE(IOUT,1998)
      STOP
```

```
C
C  *****   Normal Exit   *****
C
  999 WRITE(IOUT,1999)
      RETURN
C
      END
C
C
C
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789
C234567890123456789012345678901234567890123456789012345678901234567890123456789
C234567890123456789012345678901234567890123456789012345678901234567890123456789
C
C
C
```

D.3 CONSTR Subroutine Code

The purpose of the CONSTR Subroutine Code is to define the constraint functions if any, in a format compatible with the NLPQLP requirements.

```
C
      SUBROUTINE CONSTR(IOUT,ITOUT,NX,MI,LSAVE,A,AL,AU,MAXASUM,MINASUM,
     1           G,G0,GMAX)
C
C
C  *****   Definition of Constraints   *****
C
C  *****   These statements establish and define the Dimensions and
C                Type of some of the Non-linear Programming Parameters.   *****
C
      INTEGER           NMAX, MMAX, MNN2X, LWA, LKWA, LACTIV
C
      PARAMETER (       NMAX   = 14,
     /                  MMAX   = 10,
     /                  MNN2X  = MMAX + NMAX + NMAX + 2,
     /                  LWA    = 1.5*NMAX*NMAX + 33*NMAX + 9*MMAX + 200,
     /                  LKWA   = NMAX + 20,
     /                  LACTIV = 2*MMAX + 10)
C
C
C  *****   These statements establish and define the Dimensions and
C                Type of some of the NLP10x10 Main Driver Parameters.   *****
C
      INTEGER*4  NCC,   NXX,   NXZ,   NZNX,   NZZ
C
      PARAMETER  (NCC=6,NXX=10,NZZ=10,NZNX=NZZ*NXX,NXZ=MAX(NXX,NZZ))
C
C
C  *****   These statements establish and define the Dimensions and
C           Type of some of the CONSTR Constraint Routine Parameters.   *****
C
      INTEGER*4  IOUT,     ITOUT,     JJ,         LL,        LMAX,
     1           LSAVE(NCC),          MI,         NX
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      REAL*8  A(NCC),     AL(NCC),    ASUM,       AU(NCC),    G(MMAX),
     1        G0(MMAX),   GMAX(MMAX), MAXASUM,    MINASUM,    ZERO
C
C
C  *****   Data Values Statement   *****
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      DATA    ZERO                                                     /
     o        0.0D+00                                                  /
C
C
C  *****   Format Statements   *****
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
 1001 FORMAT(//2H  )
 1004 FORMAT(5X,I2,3X,3D20.8)
 1005 FORMAT(/37X,6H LSAVE/)
 1006 FORMAT((10I7/))
 1007 FORMAT(//2X,8H Element,7X,11H Constraint,9X,10H Amplitude,10X,
     1        8H Max Amp/)
 1008 FORMAT(///17X,45H *****   No Constraints are Specified   *****//)
 1998 FORMAT(19X,39H   *****   ERROR STOP in CONSTR   *****/)
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
C
C  *****   Execute CONSTR   *****
C
C  *****   Set constants and initial values   *****
C
      IF (MI) 998, 30, 10
C
   10 LMAX  = NX/2
      JJ    = 0
      ASUM  = ZERO
C
```

```
C  *****   Define Constraint Vector   *****
C
      DO 11 LL = 1, LMAX
      IF (LSAVE(LL)) 998, 11, 12
   12 JJ  = JJ + 1
      G0(JJ)   = A(LL)
      GMAX(JJ) = AU(LL)
      G(JJ)    = GMAX(JJ) - G0(JJ)
      ASUM     = ASUM + G0(JJ)
   11 CONTINUE
C
      IF (JJ - MI) 21, 30, 998
C
   21 JJ  =  JJ + 1
      A(JJ)    = ASUM
      AL(JJ)   = MINASUM
      AU(JJ)   = MAXASUM
      G0(JJ)   = ASUM
      GMAX(JJ) = MAXASUM
      G(JJ)    = GMAX(JJ) - G0(JJ)
C
C  *****   Normal Exit   *****
C
   30 IF (IABS(ITOUT) - 4) 999, 31, 31
   31 IF (JJ) 998, 34, 32
   32 WRITE(IOUT,1005)
      WRITE(IOUT,1006) (LSAVE(LL), LL=1,LMAX)
      WRITE(IOUT,1007)
      DO 33 JJ = 1, MI
      WRITE(IOUT,1004)  JJ, G(JJ), G0(JJ), GMAX(JJ)
   33 CONTINUE
      GO TO 999
   34 WRITE(IOUT,1008)
      GO TO 999
C
C  *****   Error Exit   *****
C
  998 WRITE(IOUT,1998)
      STOP
C
C  *****   Normal Exit   *****
C
  999 RETURN
C
      END
C
C
C
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
C
C
```

D.4 AMPHSE Subroutine Code

The purpose of the AMPHSE Subroutine Code is to convert angular quantities from Polar to Cartesian co-ordinates or vice versa.

```
C
      SUBROUTINE AMPHSE(IDIR,EPS,DEG,RAD,ITC,A,PSIDEG,X,Y)
C
C
C   *****   Conversion between Polar and Cartesian Co-ordinates   *****
C
C
      CHARACTER*2  ITC
C
      INTEGER*4    IDIR
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      REAL*8     A,          ABSX,       ABSY,       DEG,        EPS,
     1           ONE,        PSIDEG,     PSIRAD,     RAD,        V,
     2           W,          X,          Y,          ZERO
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
      DATA       ONE,      ZERO                                     /
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
     o           1.0D+00,  0.0D+00                                  /
C
C234567890123456789012345678901234567890123456789012345678901234567890
C
C
C  *****   Execute AMPHSE   *****
C
      IF (IDIR) 10, 10, 20
C
C   *****   Given A and PSI,  Compute X and Y   *****
C
   10 ITC       = '  '
      PSIRAD    =  PSIDEG*RAD
      X         =  A*DSIN(PSIRAD)
      Y         =  A*DCOS(PSIRAD)
      GO TO 999
C
C   *****   Given X and Y,  Compute A and PSI   *****
C
   20 ABSX      =  DABS(X)
      ABSY      =  DABS(Y)
      IF (ABSX - EPS) 21, 21, 22
   21 IF (ABSY - EPS) 26, 26, 22
C
C  *****   Amplitude A is greater than EPS   *****
C
   22 PSIDEG    =  DEG*DATAN2(X,Y)
      A         =  DSQRT(X*X + Y*Y)
      GO TO 30
C
C  *****   Amplitude A is less than EPS   *****
C
   26 ITC       = ' U'
      A         = ZERO
      PSIDEG    = ZERO
C
   30 CONTINUE
C
C   *****   End of Case   *****
C
  999 RETURN
C
      END
C
C
C
C
C
C234567890123456789012345678901234567890123456789012345678901234567890
C234567890123456789012345678901234567890123456789012345678901234567890
C234567890123456789012345678901234567890123456789012345678901234567890
```

C
C
C

D.5 RANQ Real Function Code

The purpose of the RANQ Real Function Code is to define a Random number.

```
C
      REAL*8 FUNCTION RANQ(ISEED)
C
      INTEGER*4  IDIST,  ISEED(4),  N
C
      REAL*8     XX
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C
      DATA       IDIST,      N                                        /
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C
     o           1,          1                                        /
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C
      CALL DLARNV( IDIST, ISEED, N, XX )
C
      RANQ   =  XX
C
      RETURN
      END
C
C
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C
C
C
```