Stanford University, ICME

CME 338 Large-Scale Numerical Optimization

Instructor: Michael Saunders Spring 2019

Notes 14: **SNOPT Example Optimizations**

1 Input and output listings

`snOptB` is the original interface to SNOPT. It treats a nonlinear objective function separately from any nonlinear constraints. Here are two examples of its use.

The Weapon Assignment problem has 12 linear constraints and 100 variables (all appearing in the nonlinear objective). The constraint data is contained in an MPS file (a standard for linear optimization data). A Fortran 77 main program defines the objective function and calls library routines for setting runtime options, inputting the MPS file, and solving the problem.

The Spring optimal control problem can be of any size. Here a 200×302 example is specified at runtime. The F77 main program generates the corresponding problem and solves it. The problem has 100 nonlinear constraints (which must come first) and 100 linear constraints. The first 101 variables appear nonlinearly in the constraints. All 202 variables are involved in the nonlinear objective.

2 The Weapon Assignment problem

```
!     ------------------------------------------------------------------
!     File t5weapon.f
!     Illustrates using SNOPT on a linearly constrained problem.
!
!     15 May 1998: First   version.
!     27 Oct 2003: Current version.
!     24 Oct 2010: Comments updated
!     ------------------------------------------------------------------
      program
     &     t5main

      implicit
     &     none
      integer
     &     maxm, maxn, maxne
      parameter
     &   ( maxm   = 1000,
     &     maxn   = 1000,
     &     maxne  = 3000 )
      character
     &     PrbNms(5)*8, Names(maxm+maxn)*8
      integer
     &     indA(maxne), hs(maxn+maxm)
      integer
     &     locA(maxn+1)
      double precision
     &     Acol(maxne), bl(maxn+maxm), bu(maxn+maxm),
     &     x(maxn+maxm), pi(maxm), rc(maxn+maxm)

      !SNOPT workspace-----------------------------------------------
      integer
     &     lencw, leniw, lenrw
      parameter
     &   ( lencw = 500, leniw = 10000, lenrw = 20000 )
      character
     &     cw(lencw)*8
      integer
     &     iw(leniw)
      double precision
     &     rw(lenrw)
      !------------------------------------------------------------------
      logical
     &     byname
      character
     &     lfile*20
      integer
     &     Errors, iSpecs, iPrint, iSumm, i1, i2, INFO, iMPS, inewB,
     &     iObj, itnlim, m, mincw, miniw, minrw, n, ne, nInf, nName,
```

```
     &     nnCon, nnJac, nOut, nnObj, nS
      double precision
     &     Obj, ObjAdd, sInf
      external
     &     dummy, t5obj
      !-------------------------------------------------------------------
      ! Specify some of the SNOPT files.
      !  iSpecs  is the Specs file   (0 if none).
      !  iPrint  is the Print file   (0 if none).
      !  iSumm   is the Summary file (0 if none).
      !
      !  nOut    is an output file used here by t5weapon.

      iSpecs =  4
      iPrint =  9
      iNewB  = 11
      iSumm  =  6
      nOut   =  6

      byname = .true.

      if ( byname ) then

         ! Unix and DOS systems.  Open the Specs and print files.

         lfile = 't5weapon.spc'
         open( iSpecs, file=lfile, status='OLD',     err=800 )

         lfile = 't5weapon.out'
         open( iPrint, file=lfile, status='UNKNOWN', err=800 )

         lfile = 't5weapon.newbasis'
         open( iNewB, file=lfile, status='UNKNOWN', err=800 )
      end if

      !-------------------------------------------------------------------
      ! First,  snInit MUST be called to initialize optional parameters
      ! to their default values.
      !-------------------------------------------------------------------
      call snInit
     &   ( iPrint, iSumm, cw, lencw, iw, leniw, rw, lenrw )

      !-------------------------------------------------------------------
      ! Read a Specs file (Optional).
      !-------------------------------------------------------------------
      call snSpec
     &   ( iSpecs, INFO, cw, lencw, iw, leniw, rw, lenrw )

      if (INFO .ne. 101  .and.  INFO .ne. 107) then
         go to 990
      end if

      !-------------------------------------------------------------------
      ! Set up the data structure for the linear constraints.
      ! MPSinp needs to know the number of nonlinear variables, etc.
      ! The following calls fetch values set in the SPECS file.
      ! Optionally, these values can be set in-line.
      !-------------------------------------------------------------------
      Errors = 0

      call snGeti
     &   ( 'Nonlinear constraints        ', nnCon, Errors,
     &     cw, lencw, iw, leniw, rw, lenrw )
      call snGeti
     &   ( 'Nonlinear Jacobian  variables', nnJac, Errors,
     &     cw, lencw, iw, leniw, rw, lenrw )
      call snGeti
     &   ( 'Nonlinear Objective variables', nnObj, Errors,
     &     cw, lencw, iw, leniw, rw, lenrw )
      call snGeti
     &   ( 'MPS file                     ',  iMPS, Errors,
     &     cw, lencw, iw, leniw, rw, lenrw )

      ! The problem name is not needed---it is set by MPSinp.
      ! Specify the OBJECTIVE, RHS, RANGES and BOUNDS to be selected
      ! from the MPS file.  Blank names mean "select the first one".

*     PrbNms(1) = '        '    ! PROBLEM   name
      PrbNms(2) = '        '    ! OBJECTIVE name
      PrbNms(3) = '        '    ! RHS       name
      PrbNms(4) = '        '    ! RANGES    name
```

```
 PrbNms(5) = '        '    ! BOUNDS    name

 if ( byname ) then

    ! Unix and DOS systems.  Open the MPS file.

    lfile = 't5weapon.mps'
    open( iMPS, file=lfile, status='OLD', err=800 )
 end if

 call MPSinp
&   ( iMPS, maxm, maxn, maxne,
&     nnCon, nnJac, nnObj,
&     m, n, ne,
&     iObj, ObjAdd, PrbNms,
&     Acol, indA, locA, bl, bu, Names,
&     hs, x, pi,
&     INFO, mincw, miniw, minrw, nS,
&     cw, lencw, iw, leniw, rw, lenrw )
 close( iMPS )

 if (INFO .ne. 103) go to 990
 nName = m + n

 !-------------------------------------------------------------------
 ! Specify any options not set in the Specs file.
 ! i1 and i2 may refer to the Print and Summary file respectively.
 ! Setting them to 0 suppresses printing.
 !-------------------------------------------------------------------
 itnlim = 1000
 i1     =    0
 i2     =    0
 call snSeti
&   ( 'Iterations       ', itnlim, i1, i2, Errors,
&     cw, lencw, iw, leniw, rw, lenrw )

 !-------------------------------------------------------------------
 ! Go for it, using a Cold start.
 ! hs    need not be set if a basis file is to be input.
 !       Otherwise, each hs(1:n) should be 0, 1, 2, 3, 4, or 5.
 !       The values are used by the Crash procedure s2crsh
 !       to choose an initial basis B.
 !       If hs(j) = 0 or 1, column j is eligible for B.
 !       If hs(j) = 2, column j is initially superbasic (not in B).
 !       If hs(j) = 3, column j is eligible for B and is given
 !                     preference over columns with hs(j) = 0 or 1.
 !       If hs(j) = 4 or 5, column j is initially nonbasic.
 !-------------------------------------------------------------------
 call snOptB
&   ( 'Cold', m, n, ne, nName,
&     nnCon, nnObj, nnJac,
&     iObj, ObjAdd, PrbNms(1),
&     dummy, t5obj,
&     Acol, indA, locA, bl, bu, Names,
&     hs, x, pi, rc,
&     INFO, mincw, miniw, minrw,
&     nS, nInf, sInf, Obj,
&     cw, lencw, iw, leniw, rw, lenrw,
&     cw, lencw, iw, leniw, rw, lenrw )

 if (INFO .eq. 82 .or. INFO .eq. 83 .or. INFO .eq. 84) then
    go to 910
 end if

 write(nOut, *) ' '
 write(nOut, *) 'snOpt finished.'
 write(nOut, *) 'Input errors  =', Errors
 write(nOut, *) 'snOptB INFO   =', INFO
 write(nOut, *) 'nInf          =', nInf
 write(nOut, *) 'sInf          =', sInf
 if (iObj .gt. 0) then
    write(nOut, *)
&               'Obj           =', ObjAdd + x(n+iObj) + Obj
 else
    write(nOut, *)
&               'Obj           =', ObjAdd + Obj
 end if
 if (INFO .gt. 30) go to 910

 stop
```

```
      !-------------------------------------------------------------
      ! Error exit.
      !-------------------------------------------------------------
  800 write(nOut, 4000) 'Error while opening file', lfile
      stop

  910 write(nOut, *) ' '
      write(nOut, *) 'STOPPING because of error condition'

  990 stop

 4000 format(/  a, 2x, a  )

      end ! program t5main

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      subroutine t5obj
     &   ( mode, n, x, f, g, nState,
     &     cw, lencw, iw, leniw, rw, lenrw )

      implicit
     &     none
      integer
     &     mode, n, nState, lencw, leniw, lenrw, iw(leniw)
      character
     &     cw(lencw)*8
      double precision
     &     f, x(n), g(n), rw(lenrw)

      !-------------------------------------------------------------
      ! t5obj  is funobj for the Weapon Assignment problem t5weapon.
      ! It assumes the Specs file contains data after the End card.
      !-------------------------------------------------------------
      intrinsic          log
      double precision   zero
      integer            nweapn,      ntargt
      parameter        ( nweapn = 5,  ntargt = 20,  zero   = 0.0d+0 )
      double precision   q(nweapn,ntargt), ql(nweapn,ntargt), u(ntargt)
      save               q               , ql               , u
      !-------------------------------------------------------------
      integer
     &     i, iSpecs, j,k
      double precision
     &     xk, t
      !-------------------------------------------------------------
      if (nState .eq. 1) then
         !-----------------------------------------------------
         ! First entry.  Read some data defining the objective.
         !-----------------------------------------------------
         ! Weapon data follows the SPECS data.
         ! The SPECS file unit number is defined by the call to snInit
         ! and saved in SNOPT workspace.

         iSpecs = iw( 11) ! Specs (options) file

         do i = 1, nweapn
            read (iSpecs, '(18f4.2)') (q(i,j), j = 1, ntargt)
            do j = 1, ntargt
               ql(i,j) = log( q(i,j) )
            end do
         end do
         read (iSpecs, '(18f4.0)') u
      end if

      !-------------
      ! Normal entry.
      !-------------
      k      = 0
      f      = zero

      do j = 1, ntargt
         t     = u(j)
         do i  = 1, nweapn
            xk = x(k+i)
            if (xk .gt. zero) t = t * q(i,j)**xk
         end do

         if (mode .eq. 2) then
            do i = 1, nweapn
               g(k+i) = t * ql(i,j)
```

```
            end do
         end if

         k     = k + nweapn
         f     = f + (t - u(j))
      end do

      end ! subroutine t50bj

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      subroutine dummy
     &   ( mode, nnCon, nnJac, neJac, x, fCon, gCon,
     &     nState, cu, lencu, iu, leniu, ru, lenru )

      implicit
     &     none

      integer
     &     mode, nnCon, nnJac, neJac, nState, lencu, leniu, lenru,
     &     iu(leniu)
      double precision
     &     x(nnJac), fCon(nnCon), gCon(neJac), ru(lenru)
      character
     &     cu(lencu)*8

      !==================================================================
      ! Problem t5weapon.
      !==================================================================

      ! Relax, no nonlinear constraints.

      end ! subroutine dummy
```

3 Print file for the Weapon Assignment problem

```
        ==============================
        S N O P T  7.4-1.3  (May 2015)
        ==============================
1


 SPECS file
 ----------

      Begin t5weapon   (Weapon Assignment Problem)

      *  MPS   input data----------------------

         Problem number               1115
         MPS file                       10
         Rows                           20
         Columns                       200
         Elements                      200
         Upper bound                   100.0
         Nonlinear Objective variables  100

      *  SNOPT input data----------------------

         Minimize
         Superbasics limit               80
         NEW BASIS file                  11
         Iterations                    1000
         Print level                      0
         Solution                       Yes
      End
1

 SNSPEC EXIT 100 -- finished successfully
 SNSPEC INFO 101 -- SPECS file read
1

 MPS Input Data
 ==============
 MPS file ..............        10
 Row limit..............        20       Problem Number.........      1115       Lower bound default....  0.00E+00
 Column limit...........       200       List limit.............         0       Upper bound default....  1.00E+02
 Elements limit ........       200       Error message limit....        10       Aij tolerance..........  1.00E-10
1

 MPS file
 --------
      1    NAME          WEAPON ASSIGNMENT PROBLEM
      2    ROWS
     15    COLUMNS
 ===>  Warning - no linear objective selected
    116    RHS
    125    BOUNDS
    126    ENDATA

 Names selected
 --------------
 Objective               (Min)       0
 RHS            B                   12
 RANGES                              0
 BOUNDS                              0

 Length of row-name hash table          2003
 Collisions during table lookup            0
 No. of INITIAL  bounds  specified         0
 No. of superbasics specified              0
1

 MPSinp EXIT 100 -- finished successfully
 MPSinp INFO 103 -- MPS file read
1
```

```
 Parameters
 ==========

 Files
 -----
 Solution file.....         0      Old basis file ........         0      Standard input.........         5
 Insert file.......         0      New basis file ........        11      (Printer)..............         9
 Punch file........         0      Backup basis file......         0      (Specs file)...........         4
 Load file.........         0      Dump file..............         0      Standard output........         6

 Frequencies
 -----------
 Print frequency.....       100      Check frequency........        60      Save new basis map.....       100
 Summary frequency...       100      Factorization frequency        50      Expand frequency.......     10000

 QP subproblems
 --------------
 QPsolver Cholesky......
 Scale tolerance.....     0.900      Minor feasibility tol..  1.00E-06      Iteration limit........      1000
 Scale option........         0      Minor optimality  tol..  1.00E-06      Minor print level......         0
 Crash tolerance.....     0.100      Pivot tolerance........  3.25E-11      New superbasics........        99
 Crash option........         3      Elastic weight.........  1.00E+00

 Partial pricing
 ---------------
 LP Partial price....         1      Prtl price section ( A)       100      Prtl price section (-I)        12
 QP Partial price....         1      Prtl price section ( A)       100      Prtl price section (-I)        12

 The SQP Method
 --------------
 Minimize............                Cold start.............                Proximal Point method..         1
 Nonlinear objectiv vars       100      Major optimality tol...  2.00E-06      Function precision.....  3.00E-13
 Unbounded step size....  1.00E+20      Superbasics limit......        80      Difference interval....  5.48E-07
 Unbounded objective....  1.00E+14      Reduced Hessian dim....        80      Central difference int.  6.70E-05
 Major step limit.......  2.00E+00      Derivative linesearch..                Derivative level.......         3
 Major iterations limit.      1000      Linesearch tolerance...   0.90000      Verify level...........         0
 Minor iterations limit.       500      Penalty parameter......  0.00E+00      Major Print Level......         1
 Time limit (secs)...... 9999999.0

 Hessian Approximation
 ---------------------
 Limited-Memory Hessian.            Hessian updates........        10      Hessian frequency......  99999999
                                                                           Hessian flush..........  99999999

 Miscellaneous
 -------------
 LU factor tolerance....      3.99      LU singularity tol.....  3.25E-11      Timing level...........         3
 LU update tolerance....      3.99      LU swap tolerance......  1.22E-04      Debug level............         0
 LU partial  pivoting...                eps (machine precision)  2.22E-16      System information.....        No
                                                                           Sticky parameters......        No
1



 Matrix statistics
 -----------------
                Total      Normal        Free       Fixed     Bounded
 Rows              12          12           0           0           0
 Columns          100           0           0           0         100

 No. of matrix elements                 135     Density      11.250
 Biggest                        1.0000E+00  (excluding fixed columns,
 Smallest                       1.0000E+00   free rows, and RHS)

 No. of objective coefficients            0

 Nonlinear constraints       0     Linear constraints      12
 Nonlinear variables       100     Linear variables         0
 Jacobian  variables         0     Objective variables    100
 Total constraints          12     Total variables        100
1
```

```
 The user has defined     100   out of     100   objective  gradients.

 Cheap test of user-supplied problem derivatives...

 The objective  gradients seem to be OK.

 Gradient projected in one direction  -4.03125705174E+00
 Difference approximation             -4.03125603289E+00
1


   Itns Major Minors    Step   nObj Feasible  Optimal      Objective     L+U BSwap     nS condZHZ
     63     0     57             1           9.0E-01 -1.8180449E+02      21           55 2.8E+01 _  r
     64     1      1 2.4E+00      3           8.6E-01 -2.8406633E+02      21           55 2.8E+01 _  rl
     75     2     11 4.6E-01      4           3.3E+00 -3.8062657E+02      21           61 3.0E+01 _s  l
     81     3      6 1.0E+00      5           8.9E+00 -1.2172351E+03      21           62 1.8E+01 _
     82     4      1 1.0E+00      6           4.7E+00 -1.4502744E+03      21           62 1.9E+01 _
     85     5      3 1.0E+00      7           1.5E+00 -1.6013318E+03      23           62 1.6E+01 _
     87     6      2 1.0E+00      8           7.1E-01 -1.6477764E+03      25           61 2.2E+01 _

 NEW BASIS file saved on file  11    itn =    100
    105     7     18 1.0E+00     10           8.2E-01 -1.6534815E+03      24           70 1.3E+02 _
    121     8     16 1.0E+00     11           3.0E-01 -1.6810936E+03      26           56 1.9E+01 _
    126     9      5 1.0E+00     12           3.9E-01 -1.6923324E+03      26           60 2.0E+01 _
    132    10      6 2.9E+00     14           3.0E-01 -1.6958623E+03      26           55 1.5E+01 _
    134    11      2 1.0E+00     15           4.3E-01 -1.7055542E+03      26           56 1.5E+01 _
    136    12      2 1.1E+00     17           1.9E-01 -1.7079470E+03      26           55 1.6E+01 _  R
    144    13      8 2.8E+00     19           1.8E-01 -1.7086365E+03      26           48 1.5E+01 _s
    149    14      5 1.0E+00     20           2.6E-01 -1.7202353E+03      28           44 1.1E+01 _
    153    15      4 1.0E+00     21           1.3E-01 -1.7239194E+03      28           41 1.1E+01 _
    157    16      4 1.0E+00     22           9.0E-02 -1.7273741E+03      30           39 7.2E+00 _
    165    17      8 1.0E+00     23           1.0E-01 -1.7294954E+03      30           34 5.8E+00 _
    167    18      2 1.0E+00     24           5.4E-02 -1.7314014E+03      32           33 5.5E+00 _
    170    19      3 1.0E+00     25           4.5E-02 -1.7319010E+03      32           31 5.5E+00 _

   Itns Major Minors    Step   nObj Feasible  Optimal      Objective     L+U BSwap     nS condZHZ
    171    20      1 1.0E+00     26           2.4E-02 -1.7325320E+03      32           31 5.5E+00 _
    173    21      2 1.0E+00     27           2.6E-02 -1.7328439E+03      32           30 5.3E+00 _
    174    22      1 1.0E+00     28           5.3E-02 -1.7333788E+03      32           30 5.3E+00 _
    177    23      3 1.0E+00     29           5.7E-02 -1.7337435E+03      32           30 6.4E+00 _  R
    181    24      4 1.0E+00     30           4.2E-02 -1.7340416E+03      34           27 3.5E+00 _s
    182    25      1 1.0E+00     31           1.7E-02 -1.7344998E+03      34           27 3.5E+00 _
    184    26      2 1.0E+00     32           2.5E-02 -1.7346638E+03      36           26 3.3E+00 _
    187    27      3 1.0E+00     33           2.9E-02 -1.7348568E+03      36           24 3.3E+00 _
    189    28      2 1.0E+00     34           2.6E-02 -1.7350334E+03      36           23 3.2E+00 _
    191    29      2 1.0E+00     35           2.2E-02 -1.7351636E+03      36           22 3.3E+00 _
    193    30      2 1.0E+00     36           1.2E-02 -1.7351936E+03      36           21 3.3E+00 _
    194    31      1 1.0E+00     37           1.1E-02 -1.7352055E+03      36           21 4.0E+00 _
    195    32      1 1.0E+00     38           4.9E-03 -1.7352142E+03      36           21 4.9E+00 _
    196    33      1 1.0E+00     39           7.7E-03 -1.7352221E+03      36           21 1.0E+01 _
    197    34      1 1.0E+00     40           7.9E-03 -1.7352534E+03      36           21 5.6E+00 _  R
    198    35      1 1.0E+00     41           8.5E-03 -1.7352662E+03      36           21 6.4E+00 _s
    199    36      1 1.0E+00     42           1.5E-02 -1.7352802E+03      36           21 9.9E+00 _

 NEW BASIS file saved on file  11    itn =    200
    200    37      1 1.0E+00     43           9.3E-03 -1.7352980E+03      36           21 1.1E+01 _
    201    38      1 1.0E+00     44           2.2E-02 -1.7353138E+03      36           21 9.8E+00 _
    205    39      4 1.0E+00     45           2.4E-02 -1.7353732E+03      36           20 6.8E+00 _

   Itns Major Minors    Step   nObj Feasible  Optimal      Objective     L+U BSwap     nS condZHZ
    208    40      3 1.0E+00     46           1.7E-02 -1.7355300E+03      36           18 7.3E+00 _
    209    41      1 1.0E+00     47           1.1E-02 -1.7355618E+03      36           18 8.2E+00 _
    210    42      1 5.1E-01     49           8.9E-03 -1.7355672E+03      36           18 1.0E+01 _
    211    43      1 1.0E+00     50           5.9E-03 -1.7355678E+03      36           18 8.9E+00 _
    212    44      1 1.0E+00     51           5.7E-04 -1.7355695E+03      36           18 9.5E+00 _
    213    45      1 1.0E+00     52           3.6E-04 -1.7355695E+03      36           18 1.0E+01 _  R
    214    46      1 1.0E+00     53           3.1E-04 -1.7355696E+03      36           18 1.0E+01 _s
    215    47      1 1.0E+00     54           1.5E-04 -1.7355696E+03      36           18 9.4E+00 _
    216    48      1 1.0E+00     55           1.1E-04 -1.7355696E+03      36           18 1.0E+01 _
    217    49      1 1.0E+00     56           1.1E-04 -1.7355696E+03      36           18 1.1E+01 _
    218    50      1 1.0E+00     57           4.2E-05 -1.7355696E+03      36           18 1.1E+01 _
    219    51      1 1.0E+00     58           2.2E-05 -1.7355696E+03      36           18 1.0E+01 _
    220    52      1 1.0E+00     59           2.5E-05 -1.7355696E+03      36           18 1.2E+01 _
    221    53      1 1.0E+00     60           1.1E-05 -1.7355696E+03      36           18 1.4E+01 _
    222    54      1 1.0E+00     61           3.9E-06 -1.7355696E+03      36           18 1.3E+01 _
    223    55      1 1.0E+00     62           2.4E-06 -1.7355696E+03      36           18 1.2E+01 _
    224    56      1 1.0E+00     63           2.5E-06 -1.7355696E+03      36           18 1.0E+01 _  R
    225    57      1 1.0E+00     64         (1.2E-06)-1.7355696E+03      36           18 9.5E+00 _s
1
```

```
 SNOPTB EXIT   0 -- finished successfully
 SNOPTB INFO   1 -- optimality conditions satisfied

 NEW BASIS file saved on file  11    itn =    225

 Problem name                WEAPON A
 No. of iterations               225   Objective          -1.7355695799E+03
 No. of major iterations          57   Linear    obj. term  0.0000000000E+00
                                       Nonlinear obj. term -1.7355695799E+03
 No. of calls to funobj           65   No. of calls to funcon             0
 No. of superbasics               18   No. of basic nonlinears           11
 No. of degenerate steps          15   Percentage                      6.67
 Max x                     26 1.0E+02   Max pi                   2 2.2E-01
 Max Primal infeas         26 4.3E-14   Max Dual infeas          72 1.2E-06
1

 Name           WEAPON A                Objective Value     -1.7355695799E+03

 Status         Optimal Soln            Iteration    225    Superbasics    18

 Objective              (Min)
 RHS            B
 Ranges
 Bounds

 Section 1 - Rows

  Number  ...Row.. State  ...Activity...  Slack Activity  ..Lower Limit.  ..Upper Limit.  .Dual Activity    ..i

     101  L1          UL       200.00000               .            None       200.00000        -0.05993      1
     102  L2          UL       100.00000               .            None       100.00000        -0.21769      2
     103  L3          UL       300.00000               .            None       300.00000        -0.06871      3
     104  L4          UL       150.00000               .            None       150.00000        -0.12359      4
     105  L5          UL       250.00000               .            None       250.00000        -0.07229      5
     106  G6         SBS        50.81562        20.81562        30.00000            None               .      6
     107  G7          LL       100.00000               .       100.00000            None         0.05993      7
     108  G8         SBS        51.13242        11.13242        40.00000            None               .      8
     109  G9         SBS        58.82362         8.82362        50.00000            None               .      9
     110  G10         LL        70.00000               .        70.00000            None         0.02700     10
     111  G11         BS        41.28121         6.28121        35.00000            None               .     11
     112  G12        SBS        62.41423        52.41423        10.00000            None               .     12
1

 Section 2 - Columns

  Number  .Column. State  ...Activity...  .Obj Gradient.  ..Lower Limit.  ..Upper Limit.  Reduced Gradnt    m+j

       1  X011        LL               .               .               .       100.00000         0.05993     13
       2  X012        LL               .        -0.15116               .       100.00000         0.06653     14
       3  X013        LL               .        -0.03539               .       100.00000         0.03332     15
       4  X014        LL               .               .               .       100.00000         0.12359     16
       5  X015        BS        50.81562        -0.07229               .       100.00000         0.00000     17
       6  X021       SBS        13.51634        -0.05993               .       100.00000        -0.00000     18
       7  X022       SBS         1.36070        -0.21769               .       100.00000         0.00000     19
       8  X023        LL               .        -0.05993               .       100.00000         0.00878     20
       9  X024        LL               .               .               .       100.00000         0.12359     21
      10  X025       SBS        45.40759        -0.07229               .       100.00000        -0.00000     22
      11  X031        LL               .               .               .       100.00000         0.05993     23
      12  X032        LL               .        -0.14090               .       100.00000         0.07679     24
      13  X033        LL               .        -0.03539               .       100.00000         0.03332     25
      14  X034        LL               .               .               .       100.00000         0.12359     26
      15  X035       SBS        48.62903        -0.07229               .       100.00000         0.00000     27
      16  X041        LL               .               .               .       100.00000         0.05993     28
      17  X042       SBS        23.48954        -0.21769               .       100.00000         0.00000     29
      18  X043        LL               .        -0.05097               .       100.00000         0.01774     30
      19  X044        LL               .               .               .       100.00000         0.12359     31
      20  X045        LL               .        -0.06404               .       100.00000         0.00825     32
      21  X051        LL               .               .               .       100.00000         0.05993     33
      22  X052       SBS        20.89961        -0.21769               .       100.00000         0.00000     34
      23  X053        LL               .        -0.05468               .       100.00000         0.01403     35
      24  X054        LL               .               .               .       100.00000         0.12359     36
      25  X055        LL               .        -0.06871               .       100.00000         0.00358     37
      26  X061     D  BS       100.00000        -0.00000               .       100.00000        -0.00000     38
      27  X062        LL               .        -0.00000               .       100.00000         0.15777     39
      28  X063        LL               .        -0.00000               .       100.00000         0.00878     40
      29  X064        LL               .               .               .       100.00000         0.06366     41
      30  X065        LL               .        -0.00000               .       100.00000         0.01236     42
      31  X071        BS        39.10012        -0.05993               .       100.00000         0.00000     43
      32  X072        LL               .        -0.11985               .       100.00000         0.09784     44
      33  X073        LL               .        -0.04743               .       100.00000         0.02128     45
      34  X074        LL               .               .               .       100.00000         0.12359     46
```

```
     35  X075       LL          .          -0.01149          .          100.00000          0.06080     47
     36  X081       SBS        27.06677    -0.05993          .          100.00000          0.00000     48
     37  X082       LL          .          -0.07318          .          100.00000          0.14452     49
     38  X083       LL          .          -0.03478          .          100.00000          0.03393     50
     39  X084       LL          .            .               .          100.00000          0.12359     51
     40  X085       LL          .            .               .          100.00000          0.07229     52
     41  X091       SBS        20.31677    -0.05993          .          100.00000         -0.00000     53
     42  X092       LL          .          -0.05993          .          100.00000          0.15777     54
     43  X093       LL          .          -0.02239          .          100.00000          0.04631     55
     44  X094       LL          .            .               .          100.00000          0.12359     56
     45  X095       LL          .            .               .          100.00000          0.07229     57
     46  X101       LL          .            .               .          100.00000          0.05993     58
     47  X102       LL          .          -0.10348          .          100.00000          0.11421     59
     48  X103       LL          .          -0.03519          .          100.00000          0.03351     60
     49  X104       LL          .          -0.02801          .          100.00000          0.09558     61
     50  X105       BS         51.13242    -0.07229          .          100.00000          0.00000     62
     51  X111       LL          .            .               .          100.00000          0.05993     63
     52  X112       LL          .            .               .          100.00000          0.21769     64
     53  X113       LL          .          -0.01317          .          100.00000          0.05554     65
     54  X114       SBS        33.19727    -0.12359          .          100.00000         -0.00000     66
     55  X115       LL          .          -0.06722          .          100.00000          0.00507     67
     56  X121       LL          .            .               .          100.00000          0.05993     68
     57  X122       LL          .          -0.02994          .          100.00000          0.18775     69
     58  X123       LL          .          -0.02994          .          100.00000          0.03876     70
     59  X124       SBS        40.93413    -0.12359          .          100.00000         -0.00000     71
     60  X125       LL          .          -0.06051          .          100.00000          0.01178     72
     61  X131       LL          .            .               .          100.00000          0.05993     73
     62  X132       LL          .            .               .          100.00000          0.21769     74
     63  X133       LL          .          -0.00770          .          100.00000          0.06100     75
     64  X134       LL          .          -0.07229          .          100.00000          0.05130     76
     65  X135       BS         54.01534    -0.07229          .          100.00000          0.00000     77
     66  X141       LL          .            .               .          100.00000          0.05993     78
     67  X142       LL          .          -0.18947          .          100.00000          0.02822     79
     68  X143       LL          .          -0.02994          .          100.00000          0.03876     80
     69  X144       BS         58.82362    -0.12359          .          100.00000         -0.00000     81
     70  X145       LL          .          -0.02994          .          100.00000          0.04235     82
     71  X151       LL          .            .               .          100.00000          0.03293     83
     72  X152       BS         26.21127    -0.19069          .          100.00000          0.00000     84
     73  X153       BS         43.78873    -0.04171          .          100.00000         -0.00000     85
     74  X154       LL          .          -0.02766          .          100.00000          0.06892     86
     75  X155       LL          .          -0.01376          .          100.00000          0.03153     87
     76  X161       LL          .            .               .          100.00000          0.05993     88
     77  X162       SBS        24.23623    -0.21770          .          100.00000         -0.00000     89
     78  X163       LL          .          -0.03440          .          100.00000          0.03430     90
     79  X164       BS         17.04498    -0.12359          .          100.00000          0.00000     91
     80  X165       LL          .          -0.01712          .          100.00000          0.05517     92
     81  X171       LL          .            .               .          100.00000          0.05993     93
     82  X172       SBS         3.80265    -0.21769          .          100.00000         -0.00000     94
     83  X173       SBS        72.03416    -0.06871          .          100.00000          0.00000     95
     84  X174       LL          .            .               .          100.00000          0.12359     96
     85  X175       LL          .            .               .          100.00000          0.07229     97
     86  X181       LL          .          -0.04227          .          100.00000          0.01766     98
     87  X182       LL          .          -0.14367          .          100.00000          0.07402     99
     88  X183       BS         57.55146    -0.06871          .          100.00000         -0.00000    100
     89  X184       LL          .            .               .          100.00000          0.12359    101
     90  X185       LL          .            .               .          100.00000          0.07229    102
     91  X191       LL          .            .               .          100.00000          0.05993    103
     92  X192       LL          .          -0.15387          .          100.00000          0.06383    104
     93  X193       SBS        64.21142    -0.06871          .          100.00000         -0.00000    105
     94  X194       LL          .            .               .          100.00000          0.12359    106
     95  X195       LL          .            .               .          100.00000          0.07229    107
     96  X201       LL          .            .               .          100.00000          0.05993    108
     97  X202       LL          .          -0.13392          .          100.00000          0.08378    109
     98  X203       BS         62.41423    -0.06871          .          100.00000         -0.00000    110
     99  X204       LL          .            .               .          100.00000          0.12359    111
    100  X205       LL          .            .               .          100.00000          0.07229    112

Time for MPS input                      0.00 seconds
Time for solving problem                0.01 seconds
Time for solution output                0.00 seconds
Time for constraint functions           0.00 seconds
Time for objective function             0.00 seconds
```

4 The Spring optimal control problem

```
*     ------------------------------------------------------------------
*     File springb.f for snOptB
*
*     This is a main program to generate an optimal control problem
*     of arbitrary size and solve it by calling SNOPT as a subroutine.
*
*     The problem size depends on a parameter T.  There are
*     2T constraints and 3T + 2 variables, as well as bounds
*     on the variables.  The first T constraints are quadratic in
*     T + 1 variables, and the objective function is quadratic in
*     T + 1 other variables.
*
*     The control problem models a spring, mass and damper system.
*     It is of the form
*
*    ------------------------------------------------------------------
*    | minimize    1/2 sum x(t)**2   (t = 0 to T)                       |
*    |                                                                  |
*    | subject to                                                       |
*    |     y(t+1)  =  y(t)  -  0.01 y(t)**2  -  0.004 x(t)  +  0.2 u(t) |
*    |                                                                  |
*    |     x(t+1)  =  x(t)  +  0.2  y(t),                               |
*    |                                                                  |
*    |     y(t)   >=  -1,     -0.2  <=  u(t)  <=  0.2,                  |
*    |                                                                  |
*    |                (all for t = 0 to T-1)                            |
*    | and                                                              |
*    |     y(0)    =   0,      y(T)  =  0,       x(0) = 10.             |
*    ------------------------------------------------------------------
*
*     For large enough T (e.g. T >= 90), the optimal objective value
*     is about 1186.382.
*
*     This model with T = 100 was used as test problem 5.11 in
*     B. A. Murtagh and M. A. Saunders (1982), A projected Lagrangian
*     algorithm and its implementation for sparse nonlinear constraints,
*     Mathematical Programming Study 16, 84--117.
*
*     14 Nov 1994: First version of spring.f, derived from manne.f.
*     26 Sep 1997: Updated for SNOPT 5.3.
*     15 Dec 2004: Current version.
*     ------------------------------------------------------------------
      program            springb

      implicit
     &     none
      integer
     &     maxT, maxm, maxn, maxne, nName
      parameter
     &   ( maxT   = 2000,
     &     maxm   = 2*maxT,
     &     maxn   = 3*maxT + 2,
     &     maxne  = 7*maxT,
     &     nName  = 1 )

      character
     &     ProbNm*8, Names(nName)*8
      integer
     &     indA(maxne) , hs(maxm+maxn), locA(maxn+1)
      double precision
     &     Acol(maxne) , bl(maxm+maxn), bu(maxm+maxn),
     &     x(maxm+maxn), pi(maxm)     , rc(maxm+maxn)

*     ------------------------------------------------------------------
*     USER workspace (none required)

      integer
     &     lenru, leniu, lencu
      parameter
     &   (  lenru = 1,
     &      leniu = 1,
     &      lencu = 1)
      integer
     &     iu(leniu)
      double precision
     &     ru(lenru)
      character
     &     cu(lencu)*8
```

```
*     ------------------------------------------------------------------
*     SNOPT workspace

      integer
     &     lenrw, leniw, lencw
      parameter
     &   (  lenrw = 1000000,
     &      leniw =  500000,
     &      lencw =     500)
      integer
     &     iw(leniw)
      double precision
     &     rw(lenrw)
      character
     &     cw(lencw)*8
*     ------------------------------------------------------------------
      logical
     &     byname
      character
     &     lfile*20
      integer
     &     Errors, INFO, iObj, iSpecs, iPrint, iSumm, m, mincw, miniw,
     &     minrw, n, ne, nnCon, nnObj, nnJac, nOut, nS, ninf, T
      double precision
     &     ObjAdd, sinf, Obj
      external
     &     SprCon, SprObj
*     ------------------------------------------------------------------
*     Specify some of the SNOPT files.
*     iSpecs  is the Specs file   (0 if none).
*     iPrint  is the Print file   (0 if none).
*     iSumm   is the Summary file (0 if none).
*     nOut    is an output file used here by the main program.

      iSpecs = 4
      iPrint = 9
      iSumm  = 6
      nOut   = 6

      byname = .true.

      if ( byname ) then

*        Unix and DOS systems.  Open the Specs and print files.

         lfile = 'springb.spc'
         open( iSpecs, file=lfile, status='OLD',     err=800 )

         lfile = 'springb.out'
         open( iPrint, file=lfile, status='UNKNOWN', err=800 )
      end if

*     ------------------------------------------------------------------
*     Set options to their default values.
*     ------------------------------------------------------------------
      call snInit
     &   ( iPrint, iSumm, cw, lencw, iw, leniw, rw, lenrw )

*     ------------------------------------------------------------------
*     Read a Specs file.  This must include "Problem number  T"
*     for some integer T.
*     ------------------------------------------------------------------
      call snSpec
     &   ( iSpecs, INFO, cw, lencw, iw, leniw, rw, lenrw )

      if (INFO .ne. 101  .and.  INFO .ne. 107) then
         stop
      end if

*     ------------------------------------------------------------------
*     The following call fetches T, which defines the number of
*     nonlinear constraints.
*     It is specified at runtime in the SPECS file.
*     ------------------------------------------------------------------
      Errors = 0

      call snGeti
     &   ( 'Problem number', T, Errors,
     &     cw, lencw, iw, leniw, rw, lenrw )

      if (T .le. 1  .or.  T .gt. maxm/2) then
```

```
         write(nOut, *) 'Invalid no. of Nonlinear constraints:', T
         stop
      end if

*     Write T into the problem name.

      write(ProbNm, '(i8)') T
      if      (T .lt.   100) then
         ProbNm(1:6) = 'Spring'
      else if (T .lt.  1000) then
         ProbNm(1:5) = 'Sprng'
      else if (T .lt. 10000) then
         ProbNm(1:4) = 'Spri'
      else
         ProbNm(1:3) = 'Spr'
      end if

      write(nOut, *) 'Spring optimal control problem.    T =', T

*     ------------------------------------------------------------------
*     Generate an T-period problem.
*     ------------------------------------------------------------------
      call spdata
     &   ( T, maxm, maxn, maxne, Errors,
     &     m, n, ne, nnCon, nnObj, nnJac,
     &     Acol, indA, locA, bl, bu, hs, x, pi )

      if (Errors .gt. 0) then
         write(nOut, *) 'Not enough storage to generate a problem ',
     &                  'with  Nonlinear constraints =', T
         stop
      end if

*     ------------------------------------------------------------------
*     Go for it, using a Cold start.
*     iObj   = 0 means there is no linear objective row in Acol(*).
*     Objadd = 0.0 means there is no constant to be added to the
*              objective.
*     hs       need not be set if a basis file is to be input.
*              Otherwise, each hs(1:n) should be 0, 1, 2, 3, 4, or 5.
*              The values are used by the Crash procedure
*              to choose an initial basis B.
*              If hs(j) = 0 or 1, column j is eligible for B.
*              If hs(j) = 2, column j is initially superbasic (not in B).
*              If hs(j) = 3, column j is eligible for B and is given
*                            preference over columns with hs(j) = 0 or 1.
*              If hs(j) = 4 or 5, column j is initially nonbasic.
*     ------------------------------------------------------------------
      iObj   = 0
      ObjAdd = 0.0d+0

      call snOptB
     &   ( 'Cold', m, n, ne, nName,
     &     nnCon, nnObj, nnJac,
     &     iObj, ObjAdd, ProbNm,
     &     SprCon, SprObj,
     &     Acol, indA, locA, bl, bu, Names,
     &     hs, x, pi, rc,
     &     INFO, mincw, miniw, minrw,
     &     nS, ninf, sinf, Obj,
     &     cu, lencu, iu, leniu, ru, lenru,
     &     cw, lencw, iw, leniw, rw, lenrw )

      if (INFO .eq. 82 .or. INFO .eq. 83 .or. INFO .eq. 84) then
         go to 900
      end if

      write(nOut, *) ' '
      write(nOut, *) 'snOpt finished.'
      write(nOut, *) 'Input  errors =', Errors
      write(nOut, *) 'snOptB INFO   =', INFO
      write(nOut, *) 'nInf          =', nInf
      write(nOut, *) 'sInf          =', sInf
      if (iObj .gt. 0) then
         write(nOut, *)
     &              'Obj           =', ObjAdd + x(n+iObj) + Obj
      else
         write(nOut, *)
     &              'Obj           =', ObjAdd + Obj
      end if
      if (INFO .ge. 30) go to 900
```

```
      stop

*     ------------------------------------------------------------------
*     Error exit.
*     ------------------------------------------------------------------
  800 write(nOut, 4000) 'Error while opening file', lfile
      stop

  900 write(nOut, *) ' '
      write(nOut, *) 'STOPPING because of error condition'
      stop

 4000 format(/  a, 2x, a  )

      end ! program springb

*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      subroutine spdata
     &   ( T, maxm, maxn, maxne, Errors,
     &     m, n, ne, nnCon, nnObj, nnJac,
     &     Acol, indA, locA, bl, bu, hs, x, pi )

      implicit
     &     none
      integer
     &     Errors, m, maxm, maxn, maxne, n, ne, nnCon, nnObj, nnJac, T,
     &     indA(maxne), hs(maxn+maxm), locA(maxn+1)
      double precision
     &     Acol(maxne)    , bl(maxn+maxm), bu(maxn+maxm),
     &     x(maxn+maxm), pi(maxm)

*     ------------------------------------------------------------------
*     spdata generates data for the "Spring" optimal control problem.
*     The constraints take the form
*              c(x) + A*x - s = 0,
*     where the Jacobian for c(x) + Ax is stored in Acol(*), and any
*     terms coming from c(x) are in the TOP LEFT-HAND CORNER of Acol(*),
*     with dimensions  nnCon x nnJac.
*     Note that the right-hand side is zero.
*     s is a set of slack variables whose bounds contain any constants
*     that might have formed a right-hand side.
*
*     The objective function is
*             f(x) + d'x
*     where d would be row iobj of A (but there is no such row in
*     this example).  f(x) involves only the FIRST nnObj variables.
*
*     On entry,
*     T       is the number of time periods.
*     maxm, maxn, maxne are upper limits on m, n, ne.
*
*     On exit,
*     Errors  is 0 if there is enough storage, 1 otherwise.
*     m       is the number of nonlinear and linear constraints.
*     n       is the number of variables.
*     ne      is the number of nonzeros in Acol(*).
*     nnCon   is the number of nonlinear constraints (they come first).
*     nnObj   is the number of nonlinear objective variables.
*     nnJac   is the number of nonlinear Jacobian variables.
*     a       is the constraint matrix (Jacobian), stored column-wise.
*     indA      is the list of row indices for each nonzero in Acol(*).
*     locA      is a set of pointers to the beginning of each column of a.
*     bl      is the lower bounds on x and s.
*     bu      is the upper bounds on x and s.
*     hs(1:n) is a set of initial states for each x  (0,1,2,3,4,5).
*     x (1:n) is a set of initial values for x.
*     pi(1:m) is a set of initial values for the dual variables pi.
*
*     14 Nov 1994: First version of spdata.
*     ------------------------------------------------------------------
      integer
     &     i, j, k, nb
*     ------------------------------------------------------------------
      double precision zero,             one
      parameter      ( zero   = 0.0d+0,  one    = 1.0d+0 )
      double precision bplus,            dummy
      parameter      ( bplus  = 1.0d+20, dummy  = 0.111111d+0 )
*     ------------------------------------------------------------------
*     T defines the dimension of the problem.
```

```
      m      = T*2
      n      = T*3 + 2
      nb     = n   + m
      nnCon  = T
      nnObj  = T*2 + 2  ! y(0:T) and x(0:T)
      nnJac  = T   + 1  ! y(0:T)
      ne     = T*7

*     Check if there is enough storage.

      Errors = 0
      if (m      .gt. maxm ) Errors = 1
      if (n      .gt. maxn ) Errors = 1
      if (ne     .gt. maxne) Errors = 1
      if (Errors .gt.   0  ) return

*     ------------------------------------------------------------------
*     Generate columns for y(t), t = 0 to T.
*     The first T rows are nonlinear, and the next T are linear.
*     The Jacobian is T x (T+1) upper bidiagonal.
*     We generate the sparsity pattern here.
*     We put in dummy numerical values for the nonlinear gradients.
*     The true non-constant values are computed by funcon.
*     ------------------------------------------------------------------
      j      = 0   ! counts the variables
      ne     = 0   ! counts the Jacobian and linear constraint entries

      do   k = 0, T
         j       =   j  + 1
         locA(j) =   ne + 1
         bl(j)   = - one
         bu(j)   =   bplus
         x (j)   = - one
         hs(j)   =   0      ! Make the y(t) nonbasic.

*        There are two Jacobian nonzeros per column,
*        except in the first and last column.

         if (k .gt. 0) then    !  Aij = 1
            ne       =   ne + 1
            indA(ne) =   k
            Acol(ne) =   one
         end if

         if (k .lt. T) then    !  Aij = .02y - 1  (nonlinear)
            ne       =   ne + 1
            indA(ne) =   k  + 1
            Acol(ne) =   dummy
         end if

*        Below the Jacobian the linear constraints are diagonal.

         if (k .lt. T) then
            ne       =   ne + 1
            indA(ne) =   T  + k + 1
            Acol(ne) = - 0.2d+0
         end if
      end do

*     ------------------------------------------------------------------
*     Generate columns for x(t), t = 0 to T.
*     They form 0.004*I in the first T rows,
*     and an upper-bidiagonal in the last T rows.
*     ------------------------------------------------------------------
      do       k = 0, T
         j       =   j  + 1
         locA(j) =   ne + 1
         bl(j)   = - bplus
         bu(j)   =   bplus
         x (j)   =   zero
         hs(j)   =   3     ! Make the x(t) basic.

*        Part of 0.004*I.

         if (k .lt. T) then
            ne       =   ne + 1
            indA(ne) =   k  + 1
            Acol(ne) =   0.004d+0
         end if

*        The bidiagonal parts have two entries
```

```
*        except in the first and last columns.

         if (k .gt. 0) then    !  Aij = 1
            ne       =   ne + 1
            indA(ne) =   T  + k
            Acol(ne) =   one
         end if

         if (k .lt. T) then    !  Aij = - 1
            ne       =   ne + 1
            indA(ne) =   T  + k + 1
            Acol(ne) = - one
         end if
      end do

*     ------------------------------------------------------------------
*     Generate columns for u(t), t = 0 to T-1.
*     They form -0.2I in the first T rows.
*     ------------------------------------------------------------------
      do k = 0, T - 1
         j       =   j  + 1
         locA(j) =   ne + 1
         bl(j)   = - 0.2d+0
         bu(j)   =   0.2d+0
         x (j)   =   zero
         hs(j)   =   3     ! Make the u(t) basic.

         ne       =   ne + 1
         indA(ne) =   k  + 1
         Acol(ne) = - 0.2d+0
      end do

*     locA(*) has one extra element.
*     Some of the variables are fixed.

      locA(n+1) = ne + 1
      bl(1)   = zero      ! y(0) = 0
      bu(1)   = zero
      bl(T+1) = zero      ! y(T) = 0
      bu(T+1) = zero
      bl(T+2) = 10.0d+0   ! x(0) = 10
      bu(T+2) = 10.0d+0

*     ------------------------------------------------------------------
*     Set bounds on the slacks.
*     We don't need to set initial values and states for slacks
*     (assuming SNOPT does a cold start).
*     ------------------------------------------------------------------
      do      j = n + 1, nb
         bl(j)  = zero
         bu(j)  = zero
      end do

*     Initialize pi.
*     SNOPT requires only pi(1:nnCon) to be initialized.
*     We initialize all of pi just in case.

      do       i = 1, T
         pi(i)   =   zero
         pi(T+i) =   zero
      end do

      end ! subroutine spdata

*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      subroutine SprObj
     &   ( mode, n, x, f, g, nState,
     &     cu, lencu, iu, leniu, ru, lenru )

      implicit
     &     none
      integer
     &     lencu, leniu, lenru, mode, n, nState, iu(leniu)
      double precision
     &     f, x(n), g(n), ru(lenru)
      character
     &     cu(lencu)*8
*     ==================================================================
*     This is funobj for problem Spring  (an optimal control problem).
*     ==================================================================
```

```
      integer
     &     jy, jx, k, T
      double precision
     &     u
*     ----------------------------------------------------------------
      double precision   zero
      parameter        ( zero = 0.0d+0 )
*     ----------------------------------------------------------------
      T      = (n - 2)/2
      f      = zero
      jy     = 0
      jx     = T + 1

      do k = 0, T
         jy    = jy + 1
         jx    = jx + 1
         u     = x(jx)
         f     = f  +  u**2
         g(jy) = zero
         g(jx) = u
      end do

      f = f / 2.0d+0

      end ! subroutine SprObj

*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      subroutine SprCon
     &   ( mode, m, n, njac, x, f, g, nState,
     &     cu, lencu, iu, leniu, ru, lenru )

      implicit
     &     none
      integer
     &     lencu, leniu, lenru, mode, m, n, njac, nState, iu(leniu)
      double precision
     &     x(n), f(m), g(njac), ru(lenru)
      character
     &     cu(lencu)*8

*     ----------------------------------------------------------------
*     This is funcon for problem Spring  (Optimal Control Problem).
*     The Jacobian is upper bidiagonal,
*     and only the diagonal terms are nonlinear.
*     The constant 1's in the Jacobian are not regenerated here.
*     ----------------------------------------------------------------
      integer
     &     i, jg, jy, T
      double precision
     &     yt, ytp1
*     ----------------------------------------------------------------
      double precision   one
      parameter        ( one = 1.0d+0 )
*     ----------------------------------------------------------------
      T     = n - 1
      jy    =     0    ! Counts y(t) variables
      jg    =   - 1    ! Counts nonlinear Jacobian elements

      do i = 1, T
         jy     = jy + 1
         jg     = jg + 2
         yt     = x(jy)
         ytp1   = x(jy + 1)
         f(i)   = 0.01d+0 * yt**2  +  (ytp1  -  yt)
         g(jg)  = 0.02d+0 * yt               -  one
*--      g(jg+1)= one      ! Constant element set by spdata.
      end do

      end ! subroutine SprCon
```

5 Print file for the Spring problem

```
        ==============================
        S N O P T  7.4-1.3  (May 2015)
        ==============================
1


 SPECS file
 ----------

      Begin Spring  (optimal control problem)
         Problem number          100  *  The generator uses this to set T

         Major iterations        500
         Minor iterations      10000

      *  Superbasics limit       200  * Set this in spring.f
      *  Iterations             1000  *     ditto

         Major Print level     000001
      *                       (JFLXBT)
         Minor Print level      00001
         Solution                 No
      End   Spring
1

 SNSPEC EXIT 100 -- finished successfully
 SNSPEC INFO 101 -- SPECS file read
1

 Parameters
 ==========

 Files
 -----
 Solution file.........        0       Old basis file ........        0       Standard input.........        5
 Insert file...........        0       New basis file ........        0       (Printer)..............        9
 Punch file............        0       Backup basis file......        0       (Specs file)...........        4
 Load file.............        0       Dump file..............        0       Standard output........        6

 Frequencies
 -----------
 Print frequency........      100       Check frequency........       60       Save new basis map.....      100
 Summary frequency......      100       Factorization frequency       50       Expand frequency.......    10000

 QP subproblems
 --------------
 QPsolver Cholesky......
 Scale tolerance........    0.900       Minor feasibility tol..  1.00E-06       Iteration limit........    10000
 Scale option...........        0       Minor optimality  tol..  1.00E-06       Minor print level......        1
 Crash tolerance........    0.100       Pivot tolerance........  3.25E-11       New superbasics........       99
 Crash option...........        3       Elastic weight.........  1.00E+05

 Partial pricing
 ---------------
 LP Partial price.......        1       Prtl price section ( A)       302       Prtl price section (-I)       200
 QP Partial price.......        1       Prtl price section ( A)       302       Prtl price section (-I)       200

 The SQP Method
 --------------
 Minimize...............                Cold start.............                 Proximal Point method..        1
 Nonlinear objectiv vars       202       Major optimality tol...  2.00E-06       Function precision.....  3.00E-13
 Unbounded step size....  1.00E+20       Superbasics limit......       203       Difference interval....  5.48E-07
 Unbounded objective....  1.00E+14       Reduced Hessian dim....       203       Central difference int.  6.70E-05
 Major step limit.......  2.00E+00       Derivative linesearch..                 Derivative level.......        3
 Major iterations limit.      500       Linesearch tolerance...   0.90000       Verify level...........        0
 Minor iterations limit.    10000       Penalty parameter......  0.00E+00       Major Print Level......        1
 Time limit (secs)...... 9999999.0

 Hessian Approximation
 ---------------------
 Limited-Memory Hessian.               Hessian updates........       10       Hessian frequency......  99999999
                                                                              Hessian flush..........  99999999

 Nonlinear constraints
 ---------------------
 Nonlinear constraints..       100       Major feasibility tol..  1.00E-06       Violation limit........  1.00E+01
 Nonlinear Jacobian vars       101
```

```
 Miscellaneous
 -------------
 LU factor tolerance....      3.99       LU singularity tol.....  3.25E-11       Timing level...........         3
 LU update tolerance....      3.99       LU swap tolerance......  1.22E-04       Debug level............         0
 LU partial  pivoting...                 eps (machine precision)  2.22E-16       System information.....        No
                                                                                 Sticky parameters......        No
1



 Matrix statistics
 -----------------
                Total      Normal        Free       Fixed     Bounded
 Rows             200           0           0         200           0
 Columns          302           0         100           3         199

 No. of matrix elements                 700     Density       1.159
 Biggest  constant element        1.0000E+00  (excluding fixed columns,
 Smallest constant element        2.0000E-01   free rows, and RHS)

 No. of objective coefficients             0

 Nonlinear constraints     100     Linear constraints     100
 Nonlinear variables       202     Linear variables       100
 Jacobian  variables       101     Objective variables    202
 Total constraints         200     Total variables        302
1


    Itn     LP mult  LP step           SumInfE NonOpt Elastic LP obj   +SBS   -SBS    -BS    Pivot     L+U ncp
    100     5.0E+00  2.0E-01           2.8E+02     78  2.8240000E+02    128    128    127  1.0E+00     398
    200    -6.0E-01  1.0E+00           1.7E+02     68  1.6640000E+02     21     21    123 -2.0E-01     398
    200                                            1  3.8000000E+01                                    398


 The user has defined     100   out of     200   constraint gradients.
 ==>  Some constraint derivatives are missing,  assumed constant.

 The user has defined     202   out of     202   objective  gradients.

 Cheap test of user-supplied problem derivatives...

 The constraint gradients seem to be OK.

 -->  The largest discrepancy was    5.93E-11  in constraint   360


 The objective  gradients seem to be OK.

 Gradient projected in one direction   2.47265191794E-02
 Difference approximation              2.47267859846E-02
1


    Itn     QP mult  QP step   rgNorm          NonOpt   QP Objective   +SBS   -SBS    -BS    Pivot     L+U ncp    nS condZHZ
    309     1.6E+03  2.9E-02                     38  4.5206601E+03    203    203    292 -6.8E-01    2220   2


   Itns Major Minors    Step   nCon Feasible  Optimal  MeritFunction     L+U BSwap     nS condZHZ Penalty
    358     0    149             1  5.5E-01  6.7E-02  9.1446000E+02    1163            2 5.4E+02         _  r
    409     1     51 1.9E-01      2  4.4E-01  3.7E-02  5.3664206E+03     780     1      1 4.0E+03 9.6E+01 _  rl
    458     2     49 2.4E-01      4  3.4E-01  2.8E-02  7.8786922E+03     672            2 1.4E+03 2.1E+02 _sM l
    511     3     53 1.0E+00      5  8.0E-03  2.9E-03  1.1847610E+03     771            3 4.2E+02 2.9E+01 _
    514     4      3 1.0E+00      6  2.9E-05  8.3E-05  1.1864107E+03     593            3 2.2E+02 4.1E+02 _
    519     5      5 1.0E+00      7 (9.4E-07) 4.8E-05  1.1863950E+03     693            3 7.2E+02 6.9E+01 _
    524     6      5 1.0E+00      8 (6.6E-07) 1.9E-05  1.1863858E+03     674            5 3.3E+03 6.9E+01 _
    526     7      2 1.0E+00      9 (3.5E-07) 6.9E-06  1.1863825E+03     597            6 3.4E+03 6.9E+01 _
    527     8      1 1.0E+00     10 (4.3E-08) 4.5E-06  1.1863823E+03     597            6 3.8E+03 6.9E+01 _
    528     9      1 1.0E+00     11 (4.1E-08) 3.3E-06  1.1863822E+03     597            6 5.2E+03 6.9E+01 _
    530    10      2 1.0E+00     12 (6.0E-08) 2.5E-06  1.1863821E+03     621     2      5 6.5E+02 6.9E+01 _
    531    11      1 1.0E+00     13 (1.1E-08) 3.2E-06  1.1863821E+03     621            5 1.7E+03 6.9E+01 _
    532    12      1 1.0E+00     14 (3.2E-08)(1.2E-06) 1.1863821E+03     621            5 7.9E+02 6.9E+01 _  R
1
```

```
SNOPTB EXIT   0 -- finished successfully
SNOPTB INFO   1 -- optimality conditions satisfied

Problem name                 Sprng100
No. of iterations                 532   Objective            1.1863820785E+03
No. of major iterations            12   Linear    obj. term  0.0000000000E+00
Penalty parameter           6.864E+01   Nonlinear obj. term  1.1863820785E+03
No. of calls to funobj             15   No. of calls to funcon             15
No. of superbasics                  5   No. of basic nonlinears           177
No. of degenerate steps             1   Percentage                       0.19
Max x                     102 1.0E+01   Max pi                      1 5.8E+02
Max Primal infeas           0 0.0E+00   Max Dual infeas            99 6.7E-04
Nonlinear constraint violn    3.5E-07

Time for MPS input                            0.00 seconds
Time for solving problem                      0.01 seconds
Time for solution output                      0.00 seconds
Time for constraint functions                 0.00 seconds
Time for objective function                   0.00 seconds
```