Solving an Optimization Problem using the NAG Library for .NET from F#

NAG has just released their latest numerical library; the NAG Library for .NET (http://www.nag.co.uk/numeric/DT/DTdescription.asp). This is the first release of the library and includes over 400 methods for key mathematical and statistical areas, including Wavelet Transforms, Integration, Interpolation and Approximation, Random Number Generators, Time Series Analysis, and Optimization. The Optimization chapter contains methods for solving LP-, QP-, LS- and NLP-problems without constraints or with constraints. A global optimizer is also included, solving problems without constraints but with bounds on the variables.

The example below illustrates how to solve a LS optimization problem using the NAG Library for .NET from F#. The problem is stated as follows: min 1/2 || b - Ax || ^ 2, where A is a (10 x 9)-matrix and b is a (10 x 1)-vector. The variables (x1,...,x9) are bounded and there are 3 general constraints.

```
open System
open NagLibrary

 //number of variables
 let n: int = 9;;

//number of rows in the matrix A
 let m: int = 10;;

//number of general constraints
 let nclin: int = 3;;

//the vector of observations
 let b =  [| 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0|];;

//lower bounds on the variables and on the constraints
 let bl = [| 0.0; 0.0; -1.0e25; 0.0; 0.0; 0.0; 0.0; 0.0; 0.0; 2.0; -1.0e25; 1.0|];;

//upper bounds on the variables and on the constraints
 let bu = [| 2.0; 2.0; 2.0; 2.0; 2.0; 2.0; 2.0; 2.0; 2.0; 1.0e25; 2.0; 4.0 |];;

//starting point
let x =  [|0.0; 0.0; 0.0; 0.0; 0.0; 0.0; 0.0; 0.0; 0.0|];;

//matrix A
let a =    array2D  [ [1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0];
                      [1.0; 2.0; 1.0; 1.0; 1.0; 1.0; 2.0; 0.0; 0.0];
                      [1.0; 1.0; 3.0; 1.0; 1.0; 1.0; -1.0; -1.0; -3.0];
                      [1.0; 1.0; 1.0; 4.0; 1.0; 1.0; 1.0; 1.0; 1.0];
                      [1.0; 1.0; 1.0; 3.0; 1.0; 1.0; 1.0; 1.0; 1.0];
                      [1.0; 1.0; 2.0; 1.0; 1.0; 0.0; 0.0; 0.0; -1.0];
                      [1.0; 1.0; 1.0; 1.0; 0.0; 1.0; 1.0; 1.0; 1.0];
                      [1.0; 1.0; 1.0; 0.0; 1.0; 1.0; 1.0; 1.0; 1.0];
                      [1.0; 1.0; 0.0; 1.0; 1.0; 1.0; 2.0; 2.0; 3.0];
                      [1.0; 0.0; 1.0; 1.0; 1.0; 1.0; 0.0; 2.0; 2.0]]

// 3 general constraints
let c  =   array2D  [[1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 1.0; 4.0];
                    [1.0; 2.0; 3.0; 4.0; -2.0; 1.0; 1.0; 1.0; 1.0];
                    [1.0; -1.0; 1.0; -1.0; 1.0; 1.0; 1.0; 1.0; 1.0]]

//on exit: it includes the Lagrange multipliers for each constraint
let clamda = Array.create (n+nclin) 0.0;;
```

```
//includes the explicit linear term of the objective function. Not referenced in this
//example
let cvec = Array.create n 0.0;;

//on exit: defines the order of the columns of a with respect to the ordering of x,.
//Not referenced in this problem
let kx = Array.create n 0;;
//specifies the status of the constraints at the start of the feasibility phase. only
//referenced for a warm start
let istate = Array.create (n+nclin) 0;;

//on exit: Errors or warnings detected by the method
let ifail = ref -1;;

//on exit:  the value of the objective function at x
let objf = ref 0.0;;

//on exit: the total number of iterations performed
let iter = ref 0;;

//an object used to configure optional parameters to this method
let options = new E04.e04ncOptions();;
options.Set("List");;
options.Set("Print level = 10");;

//method used to solve the problem
E04.e04nc(m, n, nclin, c, bl, bu, cvec, istate, kx, x, a, b, iter, objf, clamda,
options, ifail)
```

Result:

```
Calls to E04NEF
 ---------------

      List
      Print level = 10

 *** E04NCF

 Parameters
 ----------

 Problem type..........       LS1      Hessian...............        NO

 Linear constraints.....        3      Feasibility tolerance..  1.05E-08
 Variables.............         9      Crash tolerance........  1.00E-02
 Objective matrix rows..       10      Rank tolerance.........  1.11E-14

 Infinite bound size....  1.00E+20      COLD start............
 Infinite step size.....  1.00E+20      EPS (machine precision)  1.11E-16

 Print level...........        10      Feasibility phase itns.       60
 Monitoring file.......        -1      Optimality  phase itns.       60

 Workspace provided is     IWORK(       9),  WORK(     261).
 To solve problem we need  IWORK(       9),  WORK(     261).

 Rank of the objective function data matrix =     6


 Itn     Step Ninf Sinf/Objective  Norm Gz
```

```
  0  0.0E+00    2   3.000000E+00  0.0E+00
  1  1.0E+00    0   1.300000E+01  2.7E+01
  2  7.4E-01    0   1.000000E+00  0.0E+00
  3  8.1E-01    0   1.266667E-01  0.0E+00
  4  1.0E+00    0   1.105244E-01  5.6E-16
  5  1.3E-01    0   9.820996E-02  3.0E-01
  6  1.0E+00    0   9.629363E-02  1.1E-16
  7  1.0E+00    0   8.134082E-02  3.5E-16

Exit from LS problem after     7 iterations.


Varbl State     Value       Lower Bound   Upper Bound    Lagr Mult   Slack

V   1    LL    0.00000          .          2.00000      0.1572         .
V   2    FR   4.152607E-02      .          2.00000        .        4.1526E-02
V   3    FR   0.587176         None        2.00000        .         1.413
V   4    LL    0.00000          .          2.00000      0.8782         .
V   5    FR   9.964323E-02      .          2.00000        .        9.9643E-02
V   6    LL    0.00000          .          2.00000      0.1473         .
V   7    FR   4.905781E-02      .          2.00000        .        4.9058E-02
V   8    LL    0.00000          .          2.00000      0.8603         .
V   9    FR   0.305649          .          2.00000        .        0.3056


L Con State     Value       Lower Bound   Upper Bound    Lagr Mult   Slack

L   1    LL    2.00000       2.00000         None       0.3777     -4.4409E-16
L   2    UL    2.00000         None        2.00000    -5.7914E-02      .
L   3    LL    1.00000       1.00000       4.00000      0.1075         .

Exit E04NCF - Optimal LS solution.

Final LS objective value =   0.8134082E-01
```

Some more examples of calling the NAG .NET library from F# can be found under the following link: http://www.nag.co.uk/numeric/DT/fsharp/#naglibrary .

For further questions regarding the use of the .NET library we - the technical team at NAG (support@nag.co.uk) and myself (serban@nag.com) - would be happy to give anyone help using the .NET Library for the first time.