NAG Toolbox for MATLAB

e04vg

1 Purpose

e04vg is used to initialize the function e04vh.

2 Syntax

```
[cw, iw, rw, ifail] = e04vg()
```

3 Description

e04vg initializes the arrays **cw**, **iw** and **rw** for the function e04vh.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

None.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

lencw, leniw, lenrw

5.4 Output Parameters

1: **cw**(**lencw**) – **string array**

2: **iw**(**leniw**) – **int32 array**

3: **rw**(**lenrw**) – **double array**

4: **ifail** – **int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

One or more of the communication array lengths **lencw**, **leniw** or **lenrw** is less than 600.

7 Accuracy

Not applicable.

8 Further Comments

The time taken by e04vg is negligible.

9 Example

e04vh_usrfun.m

```
function [status, f, g, user] = usrfun(status, n, x, needf, nf, f,
needg, leng, g, user)

  if (needf > 0)
%   The nonlinear components of f_i(x) need to be assigned,
%   for i = 1 to NF
    f(1) = 1000*sin(-x(1)-0.25) + 1000*sin(-x(2) -0.25);
    f(2) = 1000*sin(x(1)-0.25) + 1000*sin(x(1)-x(2) -0.25);
    f(3) = 1000*sin(x(2)-x(1)-0.25) + 1000*sin(x(2) -0.25);
%   n.b. in this example there is no need to assign for the wholly
%   linear components f_4(x) and f_5(x).
    f(6) = 1.0d-6*x(3)^3 + 2.0d-6*x(4)^3/3;
  end

  if (needg > 0)
%   The derivatives of the function f_i(x) need to be assigned.
%   G(k) should be set to partial derivative df_i(x)/dx_j where
%   i = IGFUN(k) and j = IGVAR(k), for k = 1 to LENG.
    g(1) = -1000*cos(-x(1)-0.25);
    g(2) = -1000*cos(-x(2)-0.25);
    g(3) = 1000*cos(x(1)-0.25) + 1000*cos(x(1)-x(2) -0.25);
    g(4) = -1000*cos(x(1)-x(2)-0.25);
    g(5) = -1000*cos(x(2)-x(1)-0.25);
    g(6) = 1000*cos(x(2)-x(1)-0.25) + 1000*cos(x(2) -0.25);
    g(7) = 3.0d-6*x(3)^2;
    g(8) = 2.0d-6*x(4)^2;
  end
```

```
start = int32(0);
objadd = 0;
objrow = int32(6);
prob = '        ';
iafun = [int32(1);
     int32(2);
     int32(4);
     int32(4);
     int32(5);
     int32(5);
     int32(6);
     int32(6)];
javar = [int32(3);
     int32(4);
     int32(1);
     int32(2);
     int32(1);
     int32(2);
     int32(3);
     int32(4)];
nea = int32(8);
a = [-1;
     -1;
     -1;
     1;
     1;
     -1;
     3;
     2];
```

```
igfun = [int32(1);
     int32(1);
     int32(2);
     int32(2);
     int32(3);
     int32(3);
     int32(6);
     int32(6)];
jgvar = [int32(1);
     int32(2);
     int32(1);
     int32(2);
     int32(1);
     int32(2);
     int32(3);
     int32(4)];
neg = int32(8);
xlow = [-0.55;
     -0.55;
     0;
     0];
xupp = [0.55;
     0.55;
     1200;
     1200];
xnames = {'X1      '; 'X2      '; 'X3      '; 'X4      '};
flow = [-894.8;
     -894.8;
     -1294.8;
     -0.55;
     -0.55;
     -9.999999999999999e+24];
fupp = [-894.8;
     -894.8;
     -1294.8;
     9.999999999999999e+24;
     9.999999999999999e+24;
     9.999999999999999e+24];
fnames = {'NlnCon 1'; 'NlnCon 2'; 'NlnCon 3'; 'LinCon 1'; 'LinCon 2';
'Objectiv'};
x = [0;
     0;
     0;
     0];
xstate = zeros(4, 1, 'int32');
f = [0;
     0;
     0;
     0;
     0;
     0];
fstate = zeros(6, 1, 'int32');
fmul = [0;
     0;
     0;
     0;
     0;
     0];
ns = int32(0);
[cw, iw, rw, ifail] = e04vg();
[xOut, xstateOut, xmul, fOut, fstateOut, fmulOut, nsOut, ninf, sinf, ...
 cwOut, iwOut, rwOut, cuser, ifail] = ...
     e04vh(start, objadd, objrow, prob, 'e04vh_usrfun', iafun, javar, ...
           nea, a, igfun, jgvar, neg, xlow, xupp, xnames, flow, fupp, ...
           fnames, x, xstate, f, fstate, fmul, ns, cw, iw, rw)
```

```
xOut =
   1.0e+03 *
    0.0001
   -0.0004
```

```
    0.6799
    1.0261
xstateOut =
           3
           3
           2
           3
xmul =
   1.0e-07 *
   -0.4568
    0.0139
    0.0008
   -0.0008
fOut =
   1.0e+03 *
   -0.8948
   -0.8948
   -1.2948
   -0.0005
    0.0005
    5.1265
fstateOut =
           0
           0
           0
           0
           0
           0
fmulOut =
   -4.3870
   -4.1056
   -5.4633
         0
         0
         0
nsOut =
           1
ninf =
           0
sinf =
     0
cwOut =
     array elided
iwOut =
     array elided
rwOut =
     array elided
cuser =
     0
ifail =
           0
```