NAG Toolbox for MATLAB

e04cc

1 Purpose

e04cc minimizes a general function $F(x)$ of n independent variables $x = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ by the Simplex method. Derivatives of the function need not be supplied.

2 Syntax

```
[x, f, user, ifail] = e04cc(x, tol, funct, monit, maxcal, 'n', n, 'user', user)
```

3 Description

e04cc finds an approximation to a minimum of a function of n variables. You must supply a (sub)program to calculate the value of the function for any set of values of the variables.

The method is iterative. A simplex of $n + 1$ points is set up in the n-dimensional space of the variables (for example in 2 dimensions the simplex is a triangle) under the assumption that the problem has been scaled so that the values of the independent variables at the minimum are of order unity. The starting point you have provided is the first vertex of the simplex, the remaining n vertices are generated by the function. The vertex of the simplex with the largest function value is reflected in the centre of gravity of the remaining vertices and the function value at this new point is compared with the remaining function values. Depending on the outcome of this test the new point is accepted or rejected, a further expansion move may be made, or a contraction may be carried out. See Nelder and Mead 1965 and Parkinson and Hutchinson 1972 for more details. When no further progress can be made the sides of the simplex are reduced in length and the method is repeated.

The method tends to be slow, but it is robust and therefore very useful for functions that are subject to inaccuracies.

4 References

Nelder J A and Mead R 1965 A simplex method for function minimization *Comput. J.* **7** 308–313

Parkinson J M and Hutchinson D 1972 An investigation into the efficiency of variants of the simplex method *Numerical Methods for Nonlinear Optimization* (ed F A Lootsma) Academic Press

5 Parameters

5.1 Compulsory Input Parameters

1: **x(n) – double array**

A guess at the position of the minimum. Note that the problem should be scaled so that the values of the $\mathbf{x}(i)$ are of order unity.

2: **tol – double scalar**

The error tolerable in the result. If f_i, for $i = 1, 2, \ldots, n + 1$, are the individual function values at the vertices of a simplex and f_m is the mean of these values, then the function will terminate when

$$\sqrt{\frac{1}{n+1}\sum_{i=1}^{n+1}(f_i - f_m)^2} < \mathbf{tol}.$$

Constraint: **tol** must be greater than or equal to the ***machine precision*** (see Chapter X02).

3: **funct – string containing name of m-file**

funct must calculate the value of the function at **xc** and assign this value to **fc**. It should be tested separately before being used in conjunction with e04cc.

```
[fc, user] = funct(n, xc, user)
```

Input Parameters

1: **n – int32 scalar**

The number n of variables.

2: **xc(n) – double array**

The point x at which the function is required.

3: **user – Any MATLAB object**

funct is called from e04cc with **user** as supplied to e04cc

Output Parameters

1: **fc – double scalar**

The value of the function $F(x)$ at the current point x.

2: **user – Any MATLAB object**

funct is called from e04cc with **user** as supplied to e04cc

4: **monit – string containing name of m-file**

monit is called once every iteration in e04cc. It can be used to print out the current values of any selection of its parameters but must not be used to change the values of the parameters.

```
[user] = monit(fmin, fmax, sim, n, is, ncall, user)
```

Input Parameters

1: **fmin – double scalar**

The smallest function value in the current simplex.

2: **fmax – double scalar**

The largest function value in the current simplex.

3: **sim(is,n) – double array**

The rows of **sim** contain the position vectors of the vertices of the current simplex.

4: **n – int32 scalar**

The number of variables.

5: **is – int32 scalar**

The first dimension of the array **sim**.

6: **ncall – int32 scalar**

The number of times that user-supplied (sub)program **funct** has been called so far.

7: **user – Any MATLAB object**

monit is called from e04cc with **user** as supplied to e04cc

Output Parameters

1: **user – Any MATLAB object**

monit is called from e04cc with **user** as supplied to e04cc

5: **maxcal – int32 scalar**

The maximum number of function evaluations to be allowed.

Constraint: **maxcal** ≥ 1.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the array **x**.

the number n of independent variables.

Constraint: **n** > 0.

2: **user – Any MATLAB object**

user is not used by e04cc, but is passed to **funct** and **monit**. Note that for large objects it may be more efficient to use a global variable which is accessible from the m-files than to use **user**.

5.3 Input Parameters Omitted from the MATLAB Interface

iw, w1, w2, w3, w4, w5, w6

5.4 Output Parameters

1: **x**(**n**) **– double array**

The value of x corresponding to the function value in **f**.

2: **f – double scalar**

The lowest function value found.

3: **user – Any MATLAB object**

user is not used by e04cc, but is passed to **funct** and **monit**. Note that for large objects it may be more efficient to use a global variable which is accessible from the m-files than to use **user**.

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 $= 1$

On entry, **n** < 1,
or **tol** $<$ ***machine precision*** (see Chapter X02),
or **iw** $\neq$ **n** $+ 1$,
or **maxcal** < 1.

ifail $= 2$

maxcal function evaluations have been completed, e04cc has been terminated prematurely. Check the coding of user-supplied (sub)program **funct** before increasing the value of **maxcal**.

7 Accuracy

On a successful exit the accuracy will be as defined by **tol**.

8 Further Comments

The time taken by e04cc depends on the number of variables, the behaviour of the function and the distance of the starting point from the minimum. Each iteration consists of 1 or 2 function evaluations unless the size of the simplex is reduced, in which case $n + 1$ function evaluations are required.

9 Example

e04cc_funct.m

```
function [fc,user] = e04cc_funct(n,xc,user)
                                                                fc          =
exp(xc(1))*(4.0*xc(1)*(xc(1)+xc(2))+2.0*xc(2)*(xc(2)+1.0)+1.0);
```

e04cc_monit.m

```
function [user] = e04cc_monit(fmin,fmax,sim,n,n1,ncall,user)
    fprintf('After  %d  function  calls,  the  value  is  %12.8g  with
simplex\n', ncall, fmin);
  fprintf('\n%12.8g.\n', sim);
```

```
x = [-1;
     1];
tol = 1.05418557512311e-08;
maxcal = int32(100);
[xOut, f, user, ifail] = e04cc(x, tol, 'e04cc_funct', 'e04cc_monit',
maxcal)
```

```
After 8 function calls, the value is    1.6554575 with simplex

          -1.

 -0.61629671.

          -1.

           1.

           1.

         0.5.
After 10 function calls, the value is    1.6554575 with simplex

          -1.

  -1.1918516.

          -1.

           1.

       0.625.
```

```
        0.5.
After 12 function calls, the value is    1.6554575 with simplex
  -1.0479629.
  -1.1918516.
         -1.
     0.78125.
       0.625.
        0.5.
After 14 function calls, the value is    1.6390231 with simplex
  -1.0479629.
 -0.85611127.
         -1.
     0.78125.
     0.65625.
        0.5.
After 16 function calls, the value is    1.3042983 with simplex
  -0.5683338.
 -0.85611127.
         -1.
    -0.03125.
     0.65625.
        0.5.
After 18 function calls, the value is   0.65742922 with simplex
  -0.5683338.
 -0.85611127.
  0.15110987.
    -0.03125.
     0.65625.
       -0.25.
After 20 function calls, the value is  0.011591044 with simplex
  -0.5683338.
  0.43888734.
  0.15110987.
    -0.03125.
     -0.9375.
       -0.25.
After 22 function calls, the value is  0.011591044 with simplex
  -0.1366676.
```

```
  0.43888734.

  0.15110987.

     -0.3125.

     -0.9375.

       -0.25.
After 24 function calls, the value is  0.011591044 with simplex

 0.079165502.

  0.43888734.

  0.15110987.

   -0.453125.

     -0.9375.

       -0.25.
After 25 function calls, the value is  0.011591044 with simplex

 0.079165502.

  0.43888734.

  0.36694297.

   -0.453125.

     -0.9375.

    -1.140625.
After 27 function calls, the value is  0.011591044 with simplex

  0.24104033.

  0.43888734.

  0.36694297.

 -0.74609375.

     -0.9375.

    -1.140625.
After 29 function calls, the value is  0.011591044 with simplex

  0.24104033.

  0.43888734.

   0.3534534.

 -0.74609375.

     -0.9375.

 -0.99121094.
After 31 function calls, the value is  0.011591044 with simplex

  0.47373539.

  0.43888734.

   0.3534534.
```

```
  -1.0734863.

     -0.9375.

 -0.99121094.
After 32 function calls, the value is  0.011591044 with simplex

  0.47373539.

  0.43888734.

  0.55916932.

  -1.0734863.

     -0.9375.

  -1.0197754.
After 34 function calls, the value is 0.0057248986 with simplex

  0.48638186.

  0.43888734.

  0.55916932.

   -1.026062.

     -0.9375.

  -1.0197754.
After 36 function calls, the value is 0.00073786732 with simplex

  0.48638186.

  0.43888734.

  0.51090196.

   -1.026062.

     -0.9375.

  -1.0007782.
After 38 function calls, the value is 0.00073786732 with simplex

  0.48638186.

  0.46876462.

  0.51090196.

   -1.026062.

 -0.97546005.

  -1.0007782.
After 40 function calls, the value is 0.00073786732 with simplex

  0.48810758.

  0.46876462.

  0.51090196.

  -1.0070906.

 -0.97546005.

  -1.0007782.
```

```
After 42 function calls, the value is 0.00073786732 with simplex

  0.48810758.

  0.51487484.

  0.51090196.

  -1.0070906.

  -1.0181715.

  -1.0007782.
After 44 function calls, the value is 0.00020074954 with simplex

  0.50049799.

  0.51487484.

  0.51090196.

  -1.0082827.

  -1.0181715.

  -1.0007782.
After 46 function calls, the value is 0.00014404663 with simplex

  0.50049799.

  0.49652511.

  0.51090196.

  -1.0082827.

 -0.99088937.

  -1.0007782.
After 48 function calls, the value is 0.00014121934 with simplex

  0.50049799.

  0.49652511.

  0.50470675.

  -1.0082827.

 -0.99088937.

  -1.0001821.
After 50 function calls, the value is 7.0566626e-06 with simplex

  0.50055696.

  0.49652511.

  0.50470675.

  -1.0019092.

 -0.99088937.

  -1.0001821.
After 52 function calls, the value is 7.0566626e-06 with simplex

  0.50055696.

  0.49957848.
```

```
  0.50470675.

  -1.0019092.

 -0.99596752.

  -1.0001821.
After 54 function calls, the value is 7.0566626e-06 with simplex

  0.50055696.

  0.49957848.

   0.4977482.

  -1.0019092.

 -0.99596752.

 -0.99831651.
After 56 function calls, the value is 7.0566626e-06 with simplex

  0.50055696.

  0.49936553.

   0.4977482.

  -1.0019092.

  -0.9980402.

 -0.99831651.
After 58 function calls, the value is 3.9934847e-06 with simplex

  0.50055696.

  0.49936553.

  0.50106777.

  -1.0019092.

  -0.9980402.

  -1.0008038.
After 60 function calls, the value is 5.2918317e-07 with simplex

  0.50055696.

  0.50008895.

  0.50106777.

  -1.0019092.

 -0.99969836.

  -1.0008038.
After 62 function calls, the value is 5.2918317e-07 with simplex

  0.50056766.

  0.50008895.

  0.50106777.

  -1.0010802.
```

```
 -0.99969836.

  -1.0008038.
After 63 function calls, the value is 5.2918317e-07 with simplex

  0.50056766.

  0.50008895.

  0.49958884.

  -1.0010802.

 -0.99969836.

  -0.9999747.
After 65 function calls, the value is 3.5086232e-07 with simplex

  0.50020328.

  0.50008895.

  0.49958884.

  -1.0004583.

 -0.99969836.

  -0.9999747.
After 67 function calls, the value is 1.4131325e-07 with simplex

  0.50020328.

  0.50008895.

  0.49986748.

  -1.0004583.

 -0.99969836.

  -1.0000265.
After 69 function calls, the value is 4.050879e-08 with simplex

  0.50020328.

  0.50006216.

  0.49986748.

  -1.0004583.

  -0.9999704.

  -1.0000265.
After 71 function calls, the value is 4.050879e-08 with simplex

  0.50008405.

  0.50006216.

  0.49986748.

  -1.0002284.

  -0.9999704.

  -1.0000265.
After 73 function calls, the value is 3.1231385e-08 with simplex
```

```
  0.50008405.

  0.50006216.

  0.49997029.

  -1.0002284.

  -0.9999704.

   -1.000063.
After 75 function calls, the value is 1.7885024e-08 with simplex

  0.49998231.

  0.50006216.

  0.49997029.

 -0.99991082.

  -0.9999704.

   -1.000063.
xOut =
    0.5000
   -0.9999
f =
   1.7885e-08
user =
     0
ifail =
           0
```