NAG Toolbox for MATLAB

c05za

1 Purpose

c05za checks the user-supplied gradients of a set of non-linear functions in several variables, for consistency with the functions themselves. The function must be called twice.

2 Syntax

```
[xp, err] = c05za(x, fvec, fjac, fvecp, mode, 'm', m, 'n', n)
```

3 Description

c05za is based on the MINPACK routine CHKDER (see Moré *et al.* 1980). It checks the *i*th gradient for consistency with the *i*th function by computing a forward-difference approximation along a suitably chosen direction and comparing this approximation with the user-supplied gradient along the same direction. The principal characteristic of c05za is its invariance under changes in scale of the variables or functions.

4 References

Moré J J, Garbow B S and Hillstrom K E 1980 User guide for MINPACK-1 *Technical Report ANL-80-74* Argonne National Laboratory

5 Parameters

5.1 Compulsory Input Parameters

1: **x**(**n**) – **double array**

The components of a point x, at which the consistency check is to be made. (See Section 8.)

2: **fvec**(**m**) – **double array**

When **mode** = 2, **fvec** must contain the functions evaluated at x.

3: **fjac**(**ldfjac,n**) – **double array**

ldfjac, the first dimension of the array, must be at least **m**.

When **mode** = 2, **fjac** must contain the user-supplied gradients. (The *i*th row of **fjac** must contain the gradient of the *i*th function evaluated at the point x.)

4: **fvecp**(**m**) – **double array**

When **mode** = 2, **fvecp** must contain the functions evaluated at **xp**.

5: **mode** – **int32 scalar**

The value 1 on the first call and the value 2 on the second call of c05za.

5.2 Optional Input Parameters

1: **m** – **int32 scalar**

Default: The dimension of the arrays **fvec**, **fvecp**. (An error is raised if these dimensions are not equal.)

the number of functions.

2: **n – int32 scalar**

Default: The dimension of the arrays **x**, **fjac**. (An error is raised if these dimensions are not equal.)

the number of variables. For use with c05pb and c05pc, $\mathbf{m} = \mathbf{n}$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldfjac

5.4 Output Parameters

1: **xp**$(*)$ **– double array**

Note: the dimension of the array **xp** must be at least **n** if **mode** $= 1$, and at least 1 otherwise.

When **mode** $= 1$, **xp** is set to a neighbouring point to **x**.

2: **err**$(*)$ **– double array**

Note: the dimension of the array **err** must be at least **m** if **mode** $= 2$, and at least 1 otherwise.

When **mode** $= 2$, **err** contains measures of correctness of the respective gradients. If there is no loss of significance (see Section 8), then if **err**(i) is 1.0 the ith user-supplied gradient is correct, whilst if **err**(i) is 0.0 the ith gradient is incorrect. For values of **err**(i) between 0.0 and 1.0 the categorisation is less certain. In general, a value of **err**$(i) > 0.5$ indicates that the ith gradient is probably correct.

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **m** < 1,
or **n** < 1,
or **ldfjac** $<$ **m**,
or **mode** $\neq 1$ or 2.

7 Accuracy

See Section 8.

8 Further Comments

The time required by c05za increases with **m** and **n**.

c05za does not perform reliably if cancellation or rounding errors cause a severe loss of significance in the evaluation of a function. Therefore, none of the components of x should be unusually small (in particular, zero) or any other value which may cause loss of significance. The relative differences between corresponding elements of **fvecp** and **fvec** should be at least two orders of magnitude greater than the ***machine precision***.

9 Example

c05za_fcn.m

```
function [fvec, fjac] = c05pc_fcn(x, iflag)

  m = 15;
  n = 3;
```

```
  y = [1.4e-1, 1.8e-1, 2.2e-1, 2.5e-1, 2.9e-1, 3.2e-1, 3.5e-1, 3.9e-1,
...
       3.7e-1, 5.8e-1, 7.3e-1, 9.6e-1, 1.34,   2.1,    4.39];


  fvec = zeros(m, 1);
  fjac = zeros(m, n);

  if (iflag ~= 2)
    for i = 1:m
      tmp1 = i;
      tmp2 = m + 1 - i;
      tmp3 = tmp1;
      if (i > (m+1)/2)
        tmp3 = tmp2;
      end
      fvec(i) = y(i) - (x(1)+tmp1/(x(2)*tmp2+x(3)*tmp3));
    end
  end

  if (iflag ~= 1)
    for i = 1:m
      tmp1 = i;
      tmp2 = m + 1 - i;
%
%     error introduced into next statement for illustration.
%     corrected statement should read    tmp3 = tmp1 .
%
      tmp3 = tmp2;
      if (i > (m+1)/2)
        tmp3 = tmp2;
      end
      tmp4 = (x(2)*tmp2+x(3)*tmp3)^2;
      fjac(i,1) = -1;
      fjac(i,2) = tmp1*tmp2/tmp4;
      fjac(i,3) = tmp1*tmp3/tmp4;
    end
  end
```

```
x = [0.92;
     0.13;
     0.54];
fvec = zeros(15,1);
fjac = zeros(15,3);
fvecp = zeros(15,1);
[xp, err] = c05za(x, fvec, fjac, fvecp, int32(1));
[fvec, fjac] = c05za_fcn(x, 3);
fvecp = c05za_fcn(xp, 1);
[tmp, err] = c05za(x, fvec, fjac, fvecp, int32(2));
fprintf(' fvec at x = %f, %f, %f\n ', x(1), x(2), x(3));
fprintf(' %f \n', fvec);
fprintf('\n fvecp at xp = %f, %f, %f\n ', xp(1), xp(2), xp(3));
fprintf(' %f \n', fvecp);
fprintf(' err \n');
fprintf(' %f \n', err);
```

```
fvec at x = 0.920000, 0.130000, 0.540000
 -1.181606
 -1.429655
 -1.606344
 -1.745269
 -1.840654
 -1.921586
 -1.984141
 -2.022537
 -2.468977
 -2.827562
 -3.473582
```

```
 -4.437612
 -6.047662
 -9.267761
 -18.918060
 fvecp at xp = 0.920000, 0.130000, 0.540000
 -1.181606
 -1.429655
 -1.606344
 -1.745269
 -1.840654
 -1.921586
 -1.984141
 -2.022537
 -2.468977
 -2.827562
 -3.473582
 -4.437612
 -6.047662
 -9.267761
 -18.918059
err
0.111989
 0.097562
 0.094922
 0.097924
 0.105290
 0.119745
 0.149805
 1.000000
 1.000000
 1.000000
 0.963773
 1.000000
 1.000000
 1.000000
 1.000000
```