NAG Toolbox

nag_roots_sys_deriv_check (c05zd)

1 Purpose

nag_roots_sys_deriv_check (c05zd) checks the user-supplied gradients of a set of nonlinear functions in several variables, for consistency with the functions themselves. The function must be called twice.

2 Syntax

```
[xp, err, ifail] = nag_roots_sys_deriv_check(mode, x, fvec, fjac, fvecp, 'm', m, 'n', n)
[xp, err, ifail] = c05zd(mode, x, fvec, fjac, fvecp, 'm', m, 'n', n)
```

3 Description

nag_roots_sys_deriv_check (c05zd) is based on the MINPACK routine CHKDER (see Moré *et al.* (1980)). It checks the ith gradient for consistency with the ith 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 nag_roots_sys_deriv_check (c05zd) 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: **mode** – INTEGER

The value 1 on the first call and the value 2 on the second call of nag_roots_sys_deriv_check (c05zd).

Constraint: **mode** $= 1$ or 2.

2: **x**(**n**) – REAL (KIND=nag_wp) array

The components of a point x, at which the consistency check is to be made. (See Section 7.)

3: **fvec**(**m**) – REAL (KIND=nag_wp) array

If **mode** $= 2$, **fvec** must contain the value of the functions evaluated at x. If **mode** $= 1$, **fvec** is not referenced.

4: **fjac**(**m**, **n**) – REAL (KIND=nag_wp) array

If **mode** $= 2$, **fjac** must contain the value of $\dfrac{\partial f_i}{\partial x_j}$ at the point x, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. If **mode** $= 1$, **fjac** is not referenced.

5: **fvecp**(**m**) – REAL (KIND=nag_wp) array

If **mode** $= 2$, **fvecp** must contain the value of the functions evaluated at **xp** (as output by a preceding call to nag_roots_sys_deriv_check (c05zd) with **mode** $= 1$). If **mode** $= 1$, **fvecp** is not referenced.

5.2 Optional Input Parameters

1: **m** – INTEGER

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

m, the number of functions.

Constraint: **m** ≥ 1.

2: **n** – INTEGER

Default: the dimension of the array **x** and the second dimension of the array **fjac**. (An error is raised if these dimensions are not equal.)

n, the number of variables. For use with nag_roots_sys_deriv_easy (c05rb), nag_roots_sys_deriv_expert (c05rc) and nag_roots_sys_deriv_rcomm (c05rd), **m** $=$ **n**.

Constraint: **n** ≥ 1.

5.3 Output Parameters

1: **xp**(**n**) – REAL (KIND=nag_wp) array

If **mode** $= 1$, **xp** is set to a point neighbouring **x**. If **mode** $= 2$, **xp** is undefined.

2: **err**(**m**) – REAL (KIND=nag_wp) array

If **mode** $= 2$, **err** contains measures of correctness of the respective gradients. If **mode** $= 1$, **err** is undefined. If there is no loss of significance (see Section 7), then if **err**(i) is 1.0 the ith user-supplied gradient $\dfrac{\partial f_i}{\partial x_j}$, for $j = 1, 2, \ldots, n$ 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.

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

Constraint: **mode** $= 1$ or 2.

ifail $= 2$

Constraint: **m** ≥ 1.

ifail $= 3$

Constraint: **n** ≥ 1.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

nag_roots_sys_deriv_check (c05zd) 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*** returned by nag_machine_precision (x02aj).

8 Further Comments

The time required by nag_roots_sys_deriv_check (c05zd) increases with **m** and **n**.

9 Example

This example checks the Jacobian matrix for a problem with 15 functions of 3 variables (sometimes referred to as the Bard problem).

9.1 Program Text

```
function c05zd_example

fprintf('c05zd example results\n\n');

% Point at which to check gradients:
x = [0.92, 0.13, 0.54];

fvec  = zeros(15, 1);
fjac  = zeros(15, 3);
fvecp = zeros(15, 1);

y = 0.01*[14, 18, 22, 25, 29, 32, 35, 39, 47, 58, 73, 96, 134, 210, 439];

[xp, err, ifail] = c05zd(nag_int(1), x, fvec, fjac, fvecp);

for i=1:15
  u = i;
  v = 16 - i;
  w = min(u, v);
  fvec(i)  = y(i) - (x(1)+u/(v*x(2)+w*x(3)));
  fvecp(i) = y(i) - (xp(1)+u/(v*xp(2)+w*xp(3)));
  denom = (v*x(2)+w*x(3))^(-2);
  fjac(i,:) = [-1, u*v*denom, u*w*denom];
end

[xp, err, ifail] = c05zd(nag_int(2), x, fvec, fjac, fvecp);

fprintf('\nAt point %12.4f %12.4f %12.4f\n', x);
if any(err <= 0.5)
  for i=1:15
    if err(i) <= 0.5
      fprintf('Suspicious gradient number %d with error measure %12.4f\n', ...
              i, err(i));
    end
  end
else
  fprintf('Gradients appear correct\n');
end
```

9.2 Program Results

```
     c05zd example results

At point       0.9200       0.1300       0.5400
Gradients appear correct
```