

NAG Fortran Library Routine Document

C05ZAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

C05ZAF checks the user-provided gradients of a set of non-linear functions in several variables, for consistency with the functions themselves. The routine must be called twice.

2 Specification

```
SUBROUTINE C05ZAF(M, N, X, FVEC, FJAC, LDFJAC, XP, FVECP, MODE, ERR)
INTEGER          M, N, LDFJAC, MODE
real           X(N), FVEC(M), FJAC(LDFJAC,N), XP(N), FVECP(M), ERR(M)
```

3 Description

C05ZAF is based upon the MINPACK routine CHKDER (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 C05ZAF is its invariance under changes in scale of the variables or functions.

4 References

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

5 Parameters

- 1: M – INTEGER *Input*
On entry: the number of functions.
- 2: N – INTEGER *Input*
On entry: the number of variables. For use with C05PBF and C05PCF, $M = N$.
- 3: X(N) – **real** array *Input*
On entry: the components of a point x , at which the consistency check is to be made. (See Section 8.)
- 4: FVEC(M) – **real** array *Input*
On entry: when $MODE = 2$, FVEC must contain the functions evaluated at x .
- 5: FJAC(LDFJAC,N) – **real** array *Input*
On entry: 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 .)
- 6: LDFJAC – INTEGER *Input*
On entry: the first dimension of the array FJAC as declared in the (sub)program from which C05ZAF is called.

Constraint: $LDFJAC \geq M$.

- 7: XP(N) – *real* array *Output*
On exit: when $MODE = 1$, XP is set to a neighbouring point to X.
- 8: FVECP(M) – *real* array *Input*
On entry: when $MODE = 2$, FVECP must contain the functions evaluated at XP.
- 9: MODE – INTEGER *Input*
On entry: the value 1 on the first call and the value 2 on the second call of C05ZAF.
- 10: ERR(M) – *real* array *Output*
On exit: 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 i th user-supplied gradient is correct, whilst if $ERR(i)$ is 0.0 the i th 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 i th gradient is probably correct.

6 Error Indicators and Warnings

None.

7 Accuracy

See below.

8 Further Comments

The time required by C05ZAF increases with M and N.

C05ZAF 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

This example checks the Jacobian matrix for a problem with 15 functions of 3 variables. The results indicate that the first 7 gradients are probably incorrect (this is caused by a deliberate error in the code to calculate the Jacobian).

9.1 Program Text

Note: the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      C05ZAF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          M, N, LDFJAC
      PARAMETER        (M=15,N=3,LDFJAC=M)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*      .. Local Scalars ..
      INTEGER          I, MODE
*      .. Local Arrays ..
      real              ERR(M), FJAC(LDFJAC,N), FVEC(M), FVECP(M), X(N),
```

```

+          XP(N)
*  .. External Subroutines ..
EXTERNAL      C05ZAF, FCN
*  .. Executable Statements ..
WRITE (NOUT,*) 'C05ZAF Example Program Results'
X(1) = 9.2e-1
X(2) = 1.3e-1
X(3) = 5.4e-1
MODE = 1

*
CALL C05ZAF(M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR)

*
CALL FCN(M,N,X,FVEC,FJAC,LDFJAC,1)
CALL FCN(M,N,X,FVEC,FJAC,LDFJAC,2)
CALL FCN(M,N,XP,FVECP,FJAC,LDFJAC,1)

*
MODE = 2

*
CALL C05ZAF(M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR)

*
WRITE (NOUT,*)
WRITE (NOUT,99999) '      FVEC at X = ', (X(I),I=1,N)
WRITE (NOUT,*)
WRITE (NOUT,99998) (FVEC(I),I=1,M)
WRITE (NOUT,*)
WRITE (NOUT,99999) '      FVECP at XP = ', (XP(I),I=1,N)
WRITE (NOUT,*)
WRITE (NOUT,99998) (FVECP(I),I=1,M)
WRITE (NOUT,*)
WRITE (NOUT,*) '      ERR'
WRITE (NOUT,*)
WRITE (NOUT,99998) (ERR(I),I=1,M)
STOP

*
99999 FORMAT (1X,A,3F12.7)
99998 FORMAT (5X,3F12.4)
END

*
SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
*  .. Parameters ..
INTEGER      M1
PARAMETER    (M1=15)
*  .. Scalar Arguments ..
INTEGER      IFLAG, LDFJAC, M, N
*  .. Array Arguments ..
real        FJAC(LDFJAC,N), FVEC(M), X(N)
*  .. Local Scalars ..
real        TMP1, TMP2, TMP3, TMP4
INTEGER      I
*  .. Local Arrays ..
real        Y(M1)
*  .. Data statements ..
DATA
+          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,
+          1.34e0, 2.1e0, 4.39e0/
*  .. Executable Statements ..
IF (IFLAG.NE.2) THEN
  DO 20 I = 1, M
    TMP1 = I
    TMP2 = M + 1 - I
    TMP3 = TMP1
    IF (I.GT.(M+1)/2) TMP3 = TMP2
    FVEC(I) = Y(I) - (X(1)+TMP1/(X(2)*TMP2+X(3)*TMP3))
20  CONTINUE
  ELSE
    DO 40 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.GT.(M+1)/2) TMP3 = TMP2
      TMP4 = (X(2)*TMP2+X(3)*TMP3)**2
      FJAC(I,1) = -1.0e0
      FJAC(I,2) = TMP1*TMP2/TMP4
      FJAC(I,3) = TMP1*TMP3/TMP4
40    CONTINUE
      END IF
      RETURN
      END

```

9.2 Program Data

None.

9.3 Program Results

C05ZAF Example Program Results

FVEC at X =	0.9200000	0.1300000	0.5400000
-1.1816	-1.4297	-1.6063	
-1.7453	-1.8407	-1.9216	
-1.9841	-2.0225	-2.4690	
-2.8276	-3.4736	-4.4376	
-6.0477	-9.2678	-18.9181	
FVECP at XP =	0.9200000	0.1300000	0.5400000
-1.1816	-1.4297	-1.6063	
-1.7453	-1.8407	-1.9216	
-1.9841	-2.0225	-2.4690	
-2.8276	-3.4736	-4.4376	
-6.0477	-9.2678	-18.9181	
ERR			
0.1120	0.0976	0.0949	
0.0979	0.1053	0.1197	
0.1498	1.0000	1.0000	
1.0000	1.0000	1.0000	
1.0000	0.9947	0.9871	
