NAG Fortran Library Routine Document

F01CKF

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

F01CKF returns with the result of the multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C.

2 Specification

```
SUBROUTINE F01CKF(A, B, C, N, P, M, Z, IZ, OPT, IFAIL)
INTEGER           N, P, M, IZ, OPT, IFAIL
real              A(N,P), B(N,M), C(M,P), Z(IZ)
```

3 Description

The n by m matrix B is post-multiplied by the m by p matrix C. If OPT $= 1$ the result is formed in the n by p matrix A. If OPT $= 2$, m must equal p, and the result is written back to B. If OPT $= 3$, n must equal m, and the result is written back to C.

4 References

None.

5 Parameters

1: A(N,P) – ***real*** array *Output*

On entry: if OPT $= 1$, A contains the result of the matrix multiplication.

2: B(N,M) – ***real*** array *Input/Output*

On entry: the n by m matrix B.

On exit: if OPT $= 2$, B contains the result of the multiplication.

3: C(M,P) – ***real*** array *Input/Output*

On entry: the m by p matrix C.

On exit: if OPT $= 3$, C contains the result of the multiplication.

4: N – INTEGER *Input*

On entry: n, the dimension of the arrays A and B as declared in the (sub)program from which F01CKF is called.

Constraint: if OPT $= 3$, N $=$ M.

5: P – INTEGER *Input*

On entry: p, the second dimension of the arrays A and C as declared in the (sub)program from which F01CKF is called.

Constraint: if OPT $= 2$, P $=$ M.

6: M – INTEGER *Input*

On entry: the second dimension of the array B and the first dimension of the array C as declared in the (sub)program from which F01CKF is called.

7: Z(IZ) – ***real*** array *Workspace*

8: IZ – INTEGER *Input*

On entry: the dimension of the array Z as declared in the (sub)program from which F01CKF is called.

Constraints:

if OPT $= 1$, IZ ≥ 1,
if OPT $\neq 1$, IZ $\geq$ M.

9: OPT – INTEGER *Input*

On entry: the value of OPT determines which array is to contain the final result.

OPT $= 1$

A must be distinct from B and C and, on exit, contains the result. B and C need not be distinct in this case.

OPT $= 2$

B must be distinct from C and on exit, contains the result. A is not used in this case and need not be distinct from B or C.

OPT $= 3$

C must be distinct from B and on exit, contains the result. A is not used in this case and need not be distinct from B or C.

Constraint: $1 \leq$ OPT ≤ 3.

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, M or P or N ≤ 0.

IFAIL $= 2$

OPT $= 2$ and M $\neq$ P.

IFAIL $= 3$

OPT $= 3$ and N $\neq$ M.

IFAIL $= 4$

OPT $\neq 1$ and IZ $<$ M.

7 Accuracy

Each element of the result is effectively computed as an inner product using ***basic precision***.

8 Further Comments

The time taken by the routine is approximately proportional to mnp.

9 Example

The example program multiplies the 2 by 3 matrix B and the 3 by 2 matrix C together and places the result in the 2 by 2 matrix A.

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.

```
*     F01CKF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, M, P, IZ
      PARAMETER        (N=2,M=3,P=2,IZ=1)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J
*     .. Local Arrays ..
      real             A(N,P), B(N,M), C(M,P), Z(IZ)
*     .. External Subroutines ..
      EXTERNAL         F01CKF
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F01CKF Example Program Results'
      DO 20 I = 1, M
         B(1,I) = real(I) - 1.0e0
         C(I,1) = B(1,I)
         B(2,I) = real(I)
         C(I,2) = B(2,I)
   20 CONTINUE
      IFAIL = 0
*
      CALL F01CKF(A,B,C,N,P,M,Z,IZ,1,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Matrix A'
      WRITE (NOUT,*)
      WRITE (NOUT,99999) ((A(I,J),J=1,P),I=1,N)
      STOP
*
99999 FORMAT (1X,2F7.1)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 F01CKF Example Program Results

 Matrix A

     5.0    8.0
     8.0   14.0
```