NAG 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 (KIND=nag_wp) 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 (KIND=nag_wp) array *Output*

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

2: B(N,M) – REAL (KIND=nag_wp) 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 (KIND=nag_wp) 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 number of rows of the array A and of the array B.

Constraints:

if OPT $= 3$, N $=$ M;
otherwise N ≥ 1.

5: P – INTEGER *Input*

On entry: p, the number of columns of the array A and of the array C.

Constraints:

if OPT $= 2$, P $=$ M;
otherwise P ≥ 1.

6: M – INTEGER *Input*

On entry: m, the number of columns of the array B and rows of the array C.

Constraints:

if OPT $= 2$, M $=$ P;
if OPT $= 3$, M $=$ N;
if OPT $\neq 1$, M $\leq$ IZ;
otherwise M ≥ 1.

7: Z(IZ) – REAL (KIND=nag_wp) 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. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, if you are 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.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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 F01CKF is approximately proportional to mnp.

9 Example

This example 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

```
    Program f01ckfe

!     F01CKF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f01ckf, nag_wp, x04cbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: indent = 0, iz = 1, ncols = 80,      &
                                           nin = 5, nout = 6, opt = 1
      Character (1), Parameter          :: diag = 'N', matrix = 'G', nolabel =  &
                                           'N'
      Character (4), Parameter          :: form = 'F7.1'
!     .. Local Scalars ..
      Integer                           :: i, ifail, j, m, n, p
      Character (8)                     :: title
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable   :: a(:,:), b(:,:), c(:,:)
      Real (Kind=nag_wp)                :: z(iz)
      Character (1)                     :: dummy(1)
!     .. Intrinsic Procedures ..
      Intrinsic                         :: real
!     .. Executable Statements ..
      Write (nout,*) 'F01CKF Example Program Results'
```

```
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, p, m

      Allocate (a(n,p),b(n,m),c(m,p))

!     Set up example B and C matrices
      Do i = 1, m
        Do j = 1, n
          b(j,i) = real(i+j-2,kind=nag_wp)
        End Do
        Do j = 1, p
          c(i,j) = real(i+j-2,kind=nag_wp)
        End Do
      End Do

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f01ckf(a,b,c,n,p,m,z,iz,opt,ifail)

!     Print the result matrix A
      title = 'Matrix A'
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04cbf(matrix,diag,n,p,a,n,form,title,nolabel,dummy,nolabel,dummy, &
        ncols,indent,ifail)

    End Program f01ckfe
```

9.2 Program Data

```
F01CKF Example Program Data
  2  2  3           : n, p, m
```

9.3 Program Results

```
 F01CKF Example Program Results

 Matrix A
      5.0    8.0
      8.0   14.0
```