Advice on Replacement Calls for Withdrawn/Superseded Routines

The following list illustrates how a call to routine, which has been withdrawn or superseded since Mark 13, may be replaced by a call to a new routine. The list indicates the minimum change necessary, but many of the replacement routines have additional flexibility and users may wish to take advantage of new features. It is strongly recommended that users consult the routine documents.

C02 – Zeros of Polynomials

C02ADF
Withdrawn at Mark 15

```
Old: CALL C02ADF(AR,AC,N,REZ,IMZ,TOL,IFAIL)
New: CALL C02AFF(A,N-1,SCALE,Z,W,IFAIL)
```

The coefficients are stored in the ***real*** array A of dimension $(2, \mathrm{N}+1)$ rather than in the arrays AR and AC, the zeros are returned in the ***real*** array Z of dimension (2,N) rather than in the arrays REZ and IMZ, and W is a ***real*** work array of dimension $(4 * (\mathrm{N}+1))$.

C02AEF
Withdrawn at Mark 15

```
Old: CALL C02AEF(A,N,REZ,IMZ,TOL,IFAIL)
New: CALL C02AGF(A,N-1,SCALE,Z,W,IFAIL)
```

The zeros are returned in the ***real*** array Z of dimension (2,N) rather than in the arrays REZ and IMZ, and W is a ***real*** work array of dimension $(2 * (\mathrm{N}+1))$.

D02 – Ordinary Differential Equations

D02BAF
Withdrawn at Mark 18

```
Old: CALL D02BAF(X,XEND,N,Y,TOL,FCN,W,IFAIL)
New: DO 10 L = 1,N
       THRES(L) = TOL
  10 CONTINUE
     CALL D02PVF(N,X,Y,XEND,TOL,THRES,2,'usualtask',.FALSE.,
     +           0.0e0,W,14*N,IFAIL)
     CALL D02PCF(FCN,XEND,X,Y,YP,YMAX,W,IFAIL)
```

THRES, YP and YMAX are ***real*** arrays of length N and the length of array W needs extending to length 14*N.

D02BBF
Withdrawn at Mark 18

```
Old: CALL D02BBF(X,XEND,N,Y,TOL,IRELAB,FCN,OUTPUT,W,IFAIL)
New: CALL D02PVF(N,X,Y,XEND,TOL,THRES,2,'usualtask',.FALSE.,
  +           0.0e0,W,14*N,IFAIL)
     ... set XWANT ...
  10 CONTINUE
     CALL D02PCF(FCN,XWANT,X,Y,YP,YMAX,W,IFAIL)
     IF (XWANT.LT.XEND) THEN
       ... reset XWANT ...
       GO TO 10
     ENDIF
```

THRES, YP and YMAX are ***real*** arrays of length N and the length of array W needs extending to length 14*N.

D02BDF
Withdrawn at Mark 18

```
Old: CALL D02BDF(X,XEND,N,Y,TOL,IRELAB,FCN,STIFF,YNORM,W,
   +            IW,M,OUTPUT,IFAIL)
New: CALL D02PVF(N,X,Y,XEND,TOL,THRES,2,'usualtask',.TRUE.,
   +            0.0e0,W,32N,IFAIL)
     ... set XWANT ...
  10 CONTINUE
     CALL D02PCF(FCN,XWANT,X,Y,YP,YMAX,IFAIL)
     IF (XWANT.LT.XEND) THEN
       ... reset XWANT ...
       GO TO 10
     ENDIF
     CALL D02PZF(RMSERR,ERRMAX,TERRMX,W,IFAIL)
```

THRES, YP, YMAX and RMSERR are ***real*** arrays of length N and W is now a ***real*** one-dimensional array of length 32*N.

D02CAF
Withdrawn at Mark 18

```
Old: CALL D02CAF(X,XEND,N,Y,TOL,FCN,W,IFAIL)
New: CALL D02CJF(X,XEND,N,Y,FCN,TOL,'M',D02CJX,D02CJW,W,IFAIL)
```

D02CJX is a subroutine provided in the NAG Fortran Library and D02CJW is a ***real*** function also provided. Both must be declared as EXTERNAL. The array W needs to be 5 elements greater in length.

D02CBF
Withdrawn at Mark 18

```
Old: CALL D02CBF(X,XEND,N,Y,TOL,IRELAB,FCN,OUTPUT,W,IFAIL)
New: CALL D02CJF(X,XEND,N,Y,FCN,TOL,RELABS,OUTPUT,D02CJW,W,IFAIL)
```

D02CJW is a ***real*** function provided in the NAG Fortran Library and must be declared as EXTERNAL. The array W needs to be 5 elements greater on length. The integer parameter IRELAB (which can take values 0, 1, 2) is provided for by the new CHARACTER*1 argument RELABS (whose corresponding values are 'M', 'A' and 'R').

D02CGF
Withdrawn at Mark 18

```
Old: CALL D02CGF(X,XEND,N,Y,TOL,HMAX,M,VAL,FCN,W,IFAIL)
New: CALL D02CJF(X,XEND,N,Y,FCN,TOL,'M',D02CJX,G,W,IFAIL)
     .
     .
     .
     real FUNCTION G(X,Y)
     real X,Y(*)
     G = Y(M)-VAL
     END
```

D02CJX is a subroutine provided in the NAG Fortran Library and should be declared as EXTERNAL. Note the functionality of HMAX is no longer available directly. Checking the value of Y(M)−VAL at intervals of length HMAX can be effected by a user-supplied procedure OUTPUT in place of D02CJX in the call described above. See the document for D02CJF for more details.

D02CHF
Withdrawn at Mark 18

```
Old: CALL D02CHF(X,XEND,N,Y,TOL,IRELAB,HMAX,FCN,G,W,IFAIL)
New: CALL D02CJF(X,XEND,N,Y,FCN,TOL,RELABS,D02CJX,G,W,IFAIL)
```

D02CJX is a subroutine provided by the NAG Fortran Library and should be declared as EXTERNAL. The functionality of HMAX can be provided as described under the replacement call for D02CGF above. The relationship between the parameters IRELAB and RELABS is described under the replacement call for D02CBF.

D02EAF
Withdrawn at Mark 18

```
Old: CALL D02EAF(X,XEND,N,Y,TOL,FCN,W,IW,IFAIL)
New: CALL D02EJF(X,XEND,N,Y,FCN,TOL,'M',D02EJX,D02EJW,D02EJY,W,IW,
    +            IFAIL)
```

D02EJY and D02EJX are subroutines provided in the NAG Fortran Library and D02EJW is a ***real*** function also provided. All must be declared as EXTERNAL.

D02EBF
Withdrawn at Mark 18

```
Old: CALL D02EBF(X,XEND,N,Y,TOL,IRELAB,FCN,MPED,PEDERV,OUTPUT,W,IW,
    +            IFAIL)
New: CALL D02EJF(X,XEND,N,Y,FCN,PEDERV,TOL,RELABS,OUTPUT,D02EJW,W,IW,
    +            IFAIL)
```

D02EJW is a ***real*** function provided in the NAG Fortran Library and must be declared as EXTERNAL. The integer parameter IRELAB (which can take values 0, 1, 2) is provided for by the new CHARACTER*1 argument RELABS (whose corresponding values are 'M', 'A' and 'R'). If MPED = 0 in the call of D02EBF then PEDERV must be the routine D02EJY, which is supplied in the Library and should be declared as EXTERNAL.

D02EGF
Withdrawn at Mark 18

```
Old: CALL D02EGF(X,XEND,N,Y,TOL,HMAX,M,VAL,FCN,W,IW,IFAIL)
New: CALL D02EJF(X,XEND,N,Y,FCN,D02EJY,TOL,'M',D02EJX,G,W,IW,IFAIL)
     .
     .
     .
     real FUNCTION G(X,Y)
     real X,Y(*)
     G = Y(M)-VAL
     END
```

D02EJY and D02EJX are subroutines provided in the NAG Fortran Library and should be declared as EXTERNAL. Note the functionality of HMAX is no longer available directly. Checking the value of Y(M)−VAL at intervals of length HMAX can be effected by a user-supplied procedure OUTPUT in place of D02EJX in the call described above. See the document for D02EJF for more details.

D02EHF
Withdrawn at Mark 18

```
Old: CALL D02EHF(X,XEND,N,Y,TOL,IRELAB,HMAX,MPED,PEDERV,FCN,G,W,IFAIL)
New: CALL D02EJF(X,XEND,N,Y,FCN,PEDERV,TOL,RELABS,D02EJX,G,W,IFAIL)
```

D02EJX is a subroutine provided by the NAG Fortran Library and should be declared as EXTERNAL. The functionality of HMAX can be provided as described under the replacement call for D02EGF above. The relationship between the parameters IRELAB and RELABS is described under the replacement call for D02EBF. If MPED = 0 in the call of D02EHF then PEDERV must be the routine D02EJY, which is supplied in the Library and should be declared as EXTERNAL.

D02PAF
Withdrawn at Mark 18

Existing programs should be modified to call D02PVF and D02PDF. The interfaces are significantly different and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine documents.

D02QAF
Withdrawn at Mark 14

Existing programs should be modified to call D02QWF and D02QFF. The interfaces are significantly different and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine documents.

D02QBF
Withdrawn at Mark 13

Existing programs should be modified to call D02NSF, D02NVF and D02NBF. The interfaces are significantly different and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine documents.

D02QDF
Withdrawn at Mark 17

Existing programs should be modified to call D02NSF, D02NVF and D02NBF, or D02NTF, D02NVF and D02NCF. The interfaces are significantly different and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine documents.

D02QQF
Withdrawn at Mark 17

Not needed except with D02QDF.

D02XAF, D02XBF
Withdrawn at Mark 18

Not needed except with D02PAF. The equivalent routine is D02PXF.

D02XGF, D02XHF
Withdrawn at Mark 14

Not needed except with D02QAF. The equivalent routine is D02QZF.

D02YAF
Withdrawn at Mark 18

There is no precise equivalent to this routine. The closest alternative routine is D02PDF.

D03 – Partial Differential Equations

D03PAF, D03PBF, D03PGF
Withdrawn at Mark 17

Should be modified to call D03PCF. The replacement routine is designed to solve a broader class of problems. Therefore it is not possible to give a precise replacement call for these routines. Users are advised to consult the appropriate routine documents for more details.

E01 – Interpolation

E01ACF
Withdrawn at Mark 15

```
Old: CALL E01ACF(A,B,X,Y,F,VAL,VALL,IFAIL,XX,WORK,AM,D,IG1,M1,N1)
New: CALL E01DAF(N1,M1,X,Y,F,PX,PY,LAMDA,MU,C,WRK,IFAIL)
     A1(1) = A
     B1(1) = B
     M = 1
     CALL E02DEF(M,PX,PY,A1,B1,LAMDA,MU,C,FF,WRK,IWRK,IFAIL)
     VAL = FF(1)
     VALL = VAL
```

where PX, PY and M are INTEGER variables, LAMDA is a ***real*** array of dimension (N1 + 4), MU is a ***real*** array of dimension (M1 + 4), C is a ***real*** array of dimension (N1*M1), WRK is a ***real*** array of dimension ((N1 + 6) * (M1 + 6)), A1, B1 and FF are ***real*** arrays of dimension (1), IWRK is an INTEGER array of dimension (M1).

The above new calls duplicate almost exactly the effect of the old call, except that the new routines produce a single interpolated value for each point, rather than the two alternative values VAL and VALL produced by the old routine. By attempting this duplication, however, efficiency is probably being sacrificed. In general it is preferable to evaluate the interpolating function provided by E01DAF at a set of M points, supplied in arrays A1 and B1, rather than at a single point. In this case, A1, B1 and FF must be dimensioned of length M.

Note also that E01ACF uses natural splines, i.e., splines having zero second derivatives at the ends of the ranges. This is likely to be slightly unsatisfactory, and E01DAF does not have this problem. It does mean however that results produced by E01DAF may not be exactly the same as those produced by E01ACF.

E01SEF
Superseded at Mark 18
Scheduled for withdrawal at Mark 20

```
Old: CALL E01SEF(M,X,Y,F,RNW,RNQ,NW,NQ,FNODES,MINNQ,WRK,IFAIL)
New: CALL E01SGF(M,X,Y,F,NW,NQ,IQ,LIQ,RQ,LRQ,IFAIL)
```

E01SEF is superseded by E01SGF which gives improved accuracy, facilities for obtaining gradient values and a consistent interface with the Mark 18 routine E01TGF for interpolation of scattered data in three dimensions.

The interpolant generated by the two routines will not be identical, but similar results may be obtained by using the same values of NW and NQ. Details of the interpolant are passed to the evaluator through the arrays IQ and RQ rather than FNODES and RNW.

E01SFF
Superseded at Mark 18
Scheduled for withdrawal at Mark 20

```
Old: CALL E01SFF(M,X,Y,F,RNW,FNODES,PX,PY,PF,IFAIL)
New: CALL E01SHF(M,X,Y,F,IQ,LIQ,RQ,LRQ,1,PX,PY,PF,QX,QY,IFAIL)
```

The two routines will not produce identical results due to differences in the generation routines E01SEF and E01SGF. Details of the interpolant are passed from E01SGF through the arrays IQ and RQ rather than FNODES and RNW.

E01SHF also returns gradient values in QX and QY and allows evaluation at arrays of points rather than just single points.

E02 – Curve and Surface Fitting

E02DBF
Withdrawn at Mark 16

```
Old: CALL E02DBF(M,PX,PY,X,Y,FF,LAMDA,MV,POINT,NPOINT,C,NC,IFAIL)
New: CALL E02DEF(M,PX,PY,X,Y,LAMDA,MU,C,FF,WRK,IWRK,IFAIL)
```

where WRK is a ***real*** array of dimension (PY − 4), and IWRK is an INTEGER array of dimension (PY − 4).

E04 – Minimizing or Maximizing a Function

E04CGF
Withdrawn at Mark 13

```
Old: CALL E04CGF(N,X,F,IW,LIW,W,LW,IFAIL)
New: CALL E04JAF(N,1,W,W(N+1),X,F,IW,LIW,W(2*N+1),LW-2*N,IFAIL)
```

E04DBF
Withdrawn at Mark 13

```
Old: CALL E04DBF(N,X,F,G,XTOL,FEST,DUM,W,FUNCT,MONIT,MAXCAL,IFAIL)
New: CALL E04DGF(N,OBJFUN,ITER,F,G,X,IWORK,WORK,IUSER,USER,IFAIL)
```

The subroutine providing function and gradient values to E04DGF is OBJFUN: it has a different parameter list to FUNCT, but can be constructed simply as:

```
      SUBROUTINE OBJFUN(MODE,N,XC,FC,GC,NSTATE,IUSER,USER)
      INTEGER    MODE, N, NSTATE, IUSER(*)
      real       XC(N), FC, GC(N), USER(*)
C
      CALL FUNCT(N,XC,FC,GC)
      RETURN
      END
```

The parameters IWORK and WORK are workspace parameters for E04DGF and must have lengths at least (N + 1) and (12*N) respectively. IUSER and USER must be declared as arrays each of length at least (1).

There is no parameter MONIT to E04DGF, but monitoring output may be obtained by calling an option setting routine. Similarly, values for FEST and MAXCAL may be supplied by calling an option setting routine. See the routine document for further information.

E04DEF
Withdrawn at Mark 13

```
Old: CALL E04DEF(N,X,F,G,IW,LIW,W,LW,IFAIL)
New: CALL E04KAF(N,1,W,W(N+1),X,F,G,IW,LIW,W(2*N+1),LW-2*N,IFAIL)
```

E04DFF
Withdrawn at Mark 13

```
Old: CALL E04DFF(N,X,F,G,IW,LIW,W,LW,IFAIL)
New: CALL E04KCF(N,1,W,W(N+1),X,F,G,IW,LIW,W(2*N+1),LW-2*N,IFAIL)
```

E04EBF
Withdrawn at Mark 13

```
Old: CALL E04EBF(N,X,F,G,IW,LIW,W,LW,IFAIL)
New: CALL E04LAF(N,1,W,W(N+1),X,F,G,IW,LIW,W(2*N+1),LW-2*N,IFAIL)
```

E04FDF
Superseded at Mark 18
Schedule for withdrawal at Mark 19

```
Old: CALL E04FDF(M,N,X,FSUMSQ,IW,LIW,W,LW,IFAIL)
New: CALL E04FYF(M,N,LSFUN,X,FSUMSQ,W,LW,IUSER,USER,IFAIL)
```

LSFUN appears in the parameter list instead of the fixed-name subroutine LSFUN1 of E04FDF. LSFUN must be declared as external in the calling subprogram. In addition it has an extra two parameters, IUSER and USER, over and above those of LSFUN1. It may be derived from LSFUN1 as follows:

```
      SUBROUTINE LSFUN(M,N,XC,FVECC,IUSER,USER)
      INTEGER    M, N, IUSER(*)
      real       XC(N), FVECC(M), USER(*)

      CALL LSFUN1(M,N,XC,FVECC)

      RETURN
      END
```

In general the extra parameters, IUSER and USER, should be declared in the calling program as IUSER(1) and USER(1), but will not need initialising.

E04GCF
Superseded at Mark 18
Scheduled for withdrawal at Mark 19

```
Old: CALL E04GCF(M,N,X,FSUMSQ,IW,LIW,W,LW,IFAIL)
New: CALL E04GYF(M,N,LSFUN,X,FSUMSQ,W,LW,IUSER,USER,IFAIL)
```

LSFUN appears in the parameter list instead of the fixed-name subroutine LSFUN2 of E04GCF. LSFUN must be declared as external in the calling subprogram. In addition it has an extra two parameters, IUSER and USER, over and above those of LSFUN2. It may be derived from LSFUN2 as follows:

```
SUBROUTINE LSFUN(M,N,XC,FVECC,FJACC,LJC,IUSER,USER)
INTEGER    M, N, LJC, IUSER(*)
real       XC(N), FVECC(M), FJACC(LJC,N), USER(*)

CALL LSFUN2(M,N,XC,FVECC,FJACC,LJC)

RETURN
END
```

In general the extra parameters, IUSER and USER, should be declared in the calling program as IUSER(1) and USER(1), but will not need initialising. If however, the array IW was used to pass information through E04GCF into LSFUN2, or get information from LSFUN2, then the array IUSER should be declared appropriately and used for this purpose.

E04GEF
Superseded at Mark 18
Scheduled for withdrawal at Mark 19

```
Old: CALL E04GEF(M,N,X,FSUMSQ,IW,LIW,W,LW,IFAIL)
New: CALL E04GZF(M,N,LSFUN,X,FSUMSQ,W,LW,IUSER,USER,IFAIL)
```

LSFUN appears in the parameter list instead of the fixed-name subroutine LSFUN2 of E04GEF. LSFUN must be declared as external in the calling subprogram. In addition it has an extra two parameters, IUSER and USER, over and above those of LSFUN2. It may be derived from LSFUN2 as follows:

```
SUBROUTINE LSFUN(M,N,X,FVECC,FJACC,LJC,IUSER,USER)
INTEGER    M, N, LJC, IUSER(*)
real       XC(N), FVECC(M), FJACC(LJC,N), USER(*)

CALL LSFUN2(M,N,XC,FVECC,FJACC,LJC)

RETURN
END
```

In general the extra parameters, IUSER and USER, should be declared in the calling program as IUSER(1) and USER(1), but will not need initialising. If however, the array IW was used to pass information through E04GEF into LSFUN2, or get information from LSFUN2, then the array IUSER should be declared appropriately and used for this purpose.

E04HBF
Withdrawn at Mark 16
Only required in conjunction with E04JBF.

E04HFF Superseded at Mark 18
Scheduled for withdrawal at Mark 19

```
Old: CALL E04HFF(M,N,X,FSUMSQ,IW,LIW,W,LW,IFAIL)
New: CALL E04HYF(M,N,LSFUN,LSHES,X,FSUMSQ,W,LW,IUSER,USER,IFAIL)
```

LSFUN and LSHES appear in the parameter list instead of the fixed-name subroutines LSFUN2 and LSHES2 of E04HFF. LSFUN and LSHES must both be declared as external in the calling subprogram. In addition they have an extra two parameters, IUSER and USER, over and above those of LSFUN2 and LSHES2. They may be derived from LSFUN2 and LSHES2 as follows:

```
SUBROUTINE LSFUN(M,N,XC,FVECC,FJACC,LJC,IUSER,USER)
INTEGER    M, N, LJC, IUSER(*)
real       XC(N), FVECC(M), FJACC(LJC,N), USER(*)

CALL LSFUN2(M,N,XC,FVECC,FJACC,LJC)

RETURN
END

SUBROUTINE LSHES(M,N,FVECC,XC,B,LB,IUSER,USER)
INTEGER    M, N, LB, IUSER(*)
real       FVECC(M), XC(N), B(LB), USER(*)

CALL LSHES2(M,N,FVECC,XC,B,LB)

RETURN
END
```

In general, the extra parameters, IUSER and USER, should be declared in the calling program as IUSER(1) and USER(1), but will not need initialising. If however, the array IW was used to pass information through E04HFF into LSFUN2 or LSHES2, or to get information from LSFUN2, then the array IUSER should be declared appropriately and used for this purpose.

E04JAF
Superseded at Mark 18
Scheduled for withdrawal at Mark 19

```
Old: CALL E04JAF(N,IBOUND,BL,BU,X,F,IW,LIW,LW,IFAIL)
New: CALL E04JYF(N,IBOUND,FUNCT,BL,BU,X,F,IW,LIW,W,LW,IUSER,USER,IFAIL)
```

FUNCT appears in the parameter list instead of the fixed-name subroutine FUNCT1 of E04JAF. FUNCT must be declared as external in the calling subprogram. In addition it has an extra two parameters, IUSER and USER, over and above those of FUNCT1. It may be derived from FUNCT1 as follows:

```
SUBROUTINE FUNCT(N,XC,FC,IUSER,USER)
INTEGER    N, IUSER(*)
real       XC(N), FC, USER(*)

CALL FUNCT1(N,XC,FC)

RETURN
END
```

The extra parameters, IUSER and USER, should be declared in the calling program as IUSER(1) and USER(1), but will not need initialising.

E04JBF
Withdrawn at Mark 16

No comparative calls are given between E04JBF and E04UCF since both routines have considerable flexibility and can be called with many different options. E04UCF allows some values to be passed to it, not through the parameter list, but as 'optional parameters', supplied through calls to E04UDF or E04UEF. Names of optional parameters are given here in bold type.

E04UCF is a more powerful routine than E04JBF, in that it allows for general linear and nonlinear constraints, and for some or all of the first derivatives to be supplied; however when replacing E04JBF, only the simple bound constraints are relevant, and only function values are assumed to be available.

Therefore E04UCF must be called with NCLIN = NCNLN = 0, with dummy arrays of size (1) supplied as the arguments A, C and CJAC, and with the name of the auxiliary routine E04UDM (UDME04 in some implementations) as the argument CONFUN. The optional parameter **Derivative Level** must be set to 0.

The subroutine providing function values to E04UCF in OBJFUN. It has a different parameter list to FUNCT, but can be constructed as follows:

```
      SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
      INTEGER    MODE, N, NSTATE, IUSER(*)
      real       X(N), OBJF, OBJGRD(N), USER(*)
      INTEGER    IFLAG,IW(1)
      real       W(1)
C
      IFLAG = 0
      CALL FUNCT(IFLAG,N,X,OBJF,OBJGRD,IW,1,W,1)
      IF (IFLAG.LT.0) MODE = IFLAG
      RETURN
      END
```

(This assumes that the arrays IW and W are not used to communicate between FUNCT and the calling program; E04UCF supplied the arrays IUSER and USER for this purpose.)

The functions of the parameters BL and BU are similar, but E04UCF has no parameter corresponding to IBOUND; all elements of BL and BU must be set (as when IBOUND = 0 in the call to E04JBF). The optional parameter **Infinite bound size** must be set to $1.0e+6$ if there are any infinite bounds. The function of the parameter ISTATE is similar but the specification is slightly different. The parameters F and G are equivalent to OBJF and OBJGRD of E04UCF. E04UCF does not allow a user-supplied routine MONIT, but intermediate output is provided by the routine, under the control of the optional parameters **Major print level** and **Minor print level**.

Most of the 'tuning' parameters in E04JBF have their counterparts as 'optional parameters' to E04UCF, as indicated in the following list, but the correspondence is not exact and the specifications must be read carefully.

IPRINT	**Minor print level**
INTYPE	**Cold start/Warm start**
MAXCAL	**Minor iteration limit** (note that this counts iterations rather than function calls)
ETA	**Line search tolerance**
XTOL	**Optimality tolerance** (note that this specifies the accuracy in F rather than the accuracy in X)
STEPMX	**Step limit**
DELTA	**Difference interval**

E04KAF
Superseded at Mark 18
Scheduled for withdrawal at Mark 19

```
Old: CALL E04KAF(N,IBOUND,BL,BU,X,F,G,IW,LIW,W,LW,IFAIL)
New: CALL E04KYF(N,IBOUND,FUNCT2,BL,BU,X,F,G,IW,LIW,W,LW,IUSER,USER,IFAIL)
```

FUNCT appears in the parameter list instead of the fixed-name subroutine FUNCT2 of E04KAF. FUNCT must be declared as external in the calling subprogram. In addition it has an extra two parameters, IUSER and USER, over and above those of FUNCT2. It may be derived from FUNCT2 as follows:

```
      SUBROUTINE FUNCT(N,XC,FC,GC,IUSER,USER)
      INTEGER    N, IUSER(*)
      real       XC(N), FC, GC(N), USER(*)

      CALL FUNCT2(N,XC,FC,GC)

      RETURN
      END
```

The extra parameters, IUSER and USER, should be declared in the calling program as IUSER(1) and USER(1), but will not need initialising.

E04KBF
Withdrawn at Mark 16

No comparative calls are given between E04KBF and E04UCF since both routines have considerable flexibility and can be called with many different options. Most of the advice given for replacing E04JBF (see above) applies also to E04KBF, and only the differences are given here.

The optional parameter **Derivative Level** must be set to 1.

The subroutine providing both function and gradient values to E04UCF is OBJFUN. It has a different parameter list to FUNCT, but can be constructed as follows:

```
      SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
      INTEGER    MODE, N, NSTATE, IUSER(*)
      real       X(N), OBJF, OBJGRD(N), USER(*)
      INTEGER    IW(1)
      real       W(1)
C
      CALL FUNCT(MODE,N,X,OBJF,OBJGRD,IW,1,W,1)
      RETURN
      END
```

E04MBF
Withdrawn at Mark 18

```
Old: CALL E04MBF(ITMAX,MSGLVL,N,NCLIN,NCTOTL,NROWA,A,BL,BU,CVEC,
     +            LINOBJ,X,ISTATE,OBJLP,CLAMDA,IWORK,LIWORK,WORK,
     +            LWORK,IFAIL)
New: CALL E04MFF(N,NCLIN,A,NROWA,BL,BU,CVEC,ISTATE,X,ITER,OBJLP,
     +            AX,CLAMDA,IWORK,LIWORK,WORK,LWORK,IFAIL)
```

The parameter NCTOTL is no longer required. Values for ITMAX, MSGLVL and LINOBJ may be supplied by calling an option setting routine.

E04MFF contains two additional parameters as follows:

ITER – INTEGER.

AX(*) – ***real*** array of dimension at least max(1,NCLIN).

The minimum value of the parameter LIWORK must be increased from 2×N to 2×N + 3. The minimum value of the parameter LWORK may also need to be changed. See the routine documents for further information.

E04NAF
Withdrawn at Mark 18

```
Old: CALL E04NAF(ITMAX,MSGLVL,N,NCLIN,NCTOTL,NROWA,NROWH,NCOLH,
     +            BIGBND,A,BL,BU,CVEC,FEATOL,HESS,QPHESS,COLD,LP,
     +            ORTHOG,X,ISTATE,ITER,OBJ,CLAMDA,IWORK,LIWORK,
     +            WORK,LWORK,IFAIL)
New: CALL E04NFF(N,NCLIN,A,NROWA,BL,BU,CVEC,HESS,NROWH,QPHESS,
     +            ISTATE,X,ITER,OBJ,AX,CLAMDA,IWORK,LIWORK,WORK,
     +            LWORK,IFAIL)
```

The specification of the subroutine QPHESS must also be changed as follows.

```
Old: SUBROUTINE QPHESS(N,NROWH,NCOLH,JTHCOL,HESS,X,HX)
     INTEGER    N, NROWH, NCOLH, JTHCOL
     real       HESS(NROWH,NCOLH), X(N), HX(N)
New: SUBROUTINE QPHESS(N,JTHCOL,HESS,NROWH,X,HX)
     INTEGER    N, JTHCOL, NROWH
     real       HESS(NROWH,*), X(N), HX(N)
```

The parameters NCTOTL, NCOLH and ORTHOG are no longer required. Values for ITMAX, MSGLVL, BIGBND, FEATOL, COLD and LP may be supplied by calling an option setting routine.

E04NFF contains one additional parameter as follows:

AX(*) – ***real*** array of dimension at least max(1,NCLIN).

The minimum value of the parameter LIWORK must be increased from 2×N to 2×N + 3. The minimum value of the parameter LWORK may also need to be changed. See the routine documents for further information.

E04UAF
Withdrawn at Mark 13

No comparative calls are given between E04UAF and E04UCF since both routines have considerable flexibility and can be called with many different options. However users of E04UAF should have no difficulty in making the transition. Most of the 'tuning' parameters in E04UAF have their counterparts as optional parameters to E04UCF, and these may be provided by calling an option setting routine prior to the call to E04UCF. The subroutines providing function and constraint values to E04UCF are OBJFUN and CONFUN respectively: they have different parameter lists to FUNCT1 and CON1, but can be constructed simply as:

```
      SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
      INTEGER    MODE, N, NSTATE, IUSER(*)
      real       X(N), OBJF, OBJGRD(N), USER(*)
C
      CALL FUNCT1(MODE,N,X,OBJF)
      RETURN
      END
      SUBROUTINE CONFUN(MODE,NCNLN,N,NROWJ,NEEDC,X,C,CJAC.NSTATE,
     +                  IUSER,USER)
      INTEGER    MODE, NCNLN, N, NROWJ, NEEDC(*), NSTATE, IUSER(*)
      real       X(X), C(*), CJAC(NROWJ,*), USER(*)
C
      CALL CON1(MODE,N,NCNLN,X,C)
      RETURN
      END
```

The parameters OBJGRD, NEEDC, CJAC, IUSER and USER are the same as those for E04UCF itself. It is important to note that, unlike FUNCT1 and CON1, a call to CONFUN is not preceded by a call to OBJFUN with the same values in X, so that FUNCT1 and CON1 will need to be modified if this property was being utilized. It should also be noted that E04UCF allows general linear constraints to be supplied separately from nonlinear constraints, and indeed this is to be encouraged, but the above call to CON1 assumes that linear constraints are being regarded as nonlinear.

E04UPF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

```
Old: CALL E04UPF(M,N,NCLIN,LDA,LDCJ,LDFJ,LDR,A,BL,BU,
    +            CONFUN,OBJFUN,ITER,ISTATE,C,CJAC,F,FJAC,
    +            CLAMDA,OBJF,R,X,IWORK,LIWORK,WORK,LWORK,
    +            IUSER,USER,IFAIL)
New: CALL E04UNF(M,N,NCLIN,LDA,LDCJ,LDFJ,LDR,A,BL,BU,Y,
    +            CONFUN,OBJFUN,ITER,ISTATE,C,CJAC,F,FJAC,
    +            CLAMDA,OBJF,R,X,IWORK,LIWORK,WORK,LWORK,
    +            IUSER,USER,IFAIL)
```

E04UNF contains one additional parameter as follows:

Y(M) – ***real*** array.

Note that a call to E04UPF is the same as a call to E04UNF with $Y(i) = 0.0$, for $i = 1, 2, \ldots, M$.

E04VCF
Withdrawn at Mark 17

```
Old: CALL E04VCF(ITMAX,MSGLVL,N,NCLIN,NCNLN,NCTOTL,NROWA,NROWJ,
    +            NROWR,BIGBND,EPSAF,ETA,FTOL,A,BL,BU,FEATOL,
    +            CONFUN,OBJFUN,COLD,FEALIN,ORTHOG,X,ISTATE,R,ITER,
    +            C,CJAC,OBJF,OBJGRD,CLAMDA,IWORK,LIWORK,WORK,LWORK,
    +            IFAIL)
New: CALL E04UCF(N,NCLIN,NCNLN,NROWA,NROWJ,NROWR,A,BL,BU,CONFUN,
    +            OBJFUN,ITER,ISTATE,C,CJAC,CLAMDA,OBJF,OBJGRD,R,X,
    +            IWORK,LIWORK,WORK,LWORK,IUSER,USER,IFAIL)
```

The specification of the subroutine OBJFUN must also be changed as follows.

```
Old: SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE)
     INTEGER    MODE, N, NSTATE
     real       X(N), OBJF, OBJGRD(N)
New: SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
     INTEGER    MODE, N, NSTATE, IUSER(*)
     real       X(N), OBJF, OBJGRD(N), USER(*)
```

If NCNLN > 0, the specification of the subroutine CONFUN must also be changed as follows.

```
Old: SUBROUTINE CONFUN(MODE,NCNLN,N,NROWJ,X,C,CJAC,NSTATE)
     INTEGER    MODE, NCNLN, N, NROWJ, NSTATE
     real       X(N), C(NROWJ), CJAC(NROWJ,N)
New: SUBROUTINE CONFUN(MODE,NCNLN,N,NROWJ,NEEDC,X,C,CJAC,NSTATE,
    +                   IUSER,USER)
     INTEGER    MODE, NCNLN, N, NROWJ, NEEDC(NCNLN), NSTATE, IUSER(*)
     real       X(N), C(NCNLN), CJAC(NROWJ,N), USER(*)
```

If NCNLN $= 0$, then the name of the dummy routine E04VDM (VDME04 in some implementations) may need to be changed to E04UDM (UDME04 in some implementations) in the calling program.

The parameters NCTOTL, EPSAF, FEALIN and ORTHOG are no longer required. Values for ITMAX, MSGLVL, BIGBND, ETA, FTOL, COLD and FEATOL may be supplied by calling an option setting routine.

E04UCF contains two additional parameters as follows:

IUSER(*) − INTEGER array of dimension at least 1.

USER(*) − ***real*** array of dimension at least 1.

The minimum value of the parameter LIWORK must be increased from $3\times$N + NCLIN + NCNLN to $3\times$N + NCLIN + $2\times$NCNLN. The minimum value of the parameter LWORK may also need to be changed. See the routine documents for further information.

E04VDF
Withdrawn at Mark 17

```
Old: IFAIL = 110
     CALL E04VDF(ITMAX,MSGLVL,N,NCLIN,NCNLN,NCTOTL,NROWA,NROWJ,
    +            CTOL,FTOL,A,BL,BU,CONFUN,OBJFUN,X,ISTATE,C,CJAC,
    +            CJAC,OBJF,OBJGRD,CLAMDA,IWORK,LIWORK,WORK,LWORK,
    +            IFAIL)
New: IFAIL = -1
     CALL E04UCF(N,NCLIN,NCNLN,NROWA,NROWJ,N,A,BL,BU,CONFUN,OBJFUN,
    +            ITER,ISTATE,C,CJAC,CLAMDA,OBJF,OBJGRD,R,X,IWORK,
    +            LIWORK,WORK,LWORK,IUSER,USER,IFAIL)
```

The specification of the subroutine OBJFUN must also be changed as follows.

```
Old: SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE)
     INTEGER    MODE, N, NSTATE
     real       X(N), OBJF, OBJGRD(N)
New: SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
     INTEGER    MODE, N, NSTATE, IUSER(*)
     real       X(N), OBJF, OBJGRD(N), USER(*)
```

If NCNLN > 0, the specification of the subroutine CONFUN must also be changed as follows.

```
Old: SUBROUTINE CONFUN(MODE,NCNLN,N,NROWJ,X,C,CJAC,NSTATE)
     INTEGER    MODE, NCNLN, N, NROWJ, NSTATE
     real       X(N), C(NROWJ), CJAC(NROWJ,N)
New: SUBROUTINE CONFUN(MODE,NCNLN,N,NROWJ,NEEDC,X,C,CJAC,NSTATE,
    +                   IUSER,USER)
     INTEGER    MODE, NCNLN, N, NROWJ, NEEDC(NCNLN), NSTATE, IUSER(*)
     real       X(N), C(NCNLN), CJAC(NROWJ,N), USER(*)
```

If NCNLN $= 0$, then the name of the dummy routine E04VDM (VDME04 in some implementations) may need to be changed to E04UDM (UDME04 in some implementations) in the calling program.

The parameter NCTOTL is no longer required. Values for ITMAX, MSGLVL, CTOL and FTOL may be supplied by calling an option setting routine.

E04UCF contains four additional parameters as follows:

ITER – INTEGER.

R(N,N) – ***real*** array.

IUSER(*) – INTEGER array of dimension at least 1.

USER(*) – ***real*** array of dimension at least 1.

The minimum value of the parameter LIWORK must be increased from 3×N + NCLIN + NCNLN to 3×N + NCLIN + 2×NCNLN. The minimum value of the parameter LWORK may also need to be changed. See the routine documents for further information.

F01 – Matrix Operations, Including Inversion

F01AAF
Withdrawn at Mark 17

```
Old: CALL F01AAF(A,IA,N,X,IX,WKSPCE,IFAIL)
New: CALL sgetrf(N,N,A,IA,IPIV,IFAIL)
     CALL F06QFF('General',N,N,A,IA,X,IX)
     CALL sgetri(N,X,IX,IPIV,WKSPCE,LWORK,IFAIL)
```

where IPIV is an INTEGER vector of length N, and the INTEGER LWORK is the length of array WKSPCE, which must be at least max(1,N). In the replacement calls, F07ADF (SGETRF/DGETRF) computes the LU factorization of the matrix A, F06QFF copies the factorization from A to X, and F07AJF (SGETRI/DGETRI) overwrites X by the inverse of A. If the original matrix A is no longer required, the call to F06QFF is not necessary, and references to X and IX in the call of F07AJF (SGETRI/DGETRI) may be replaced by references to A and IA, in which case A will be overwritten by the inverse.

F01ACF
Withdrawn at Mark 16

```
Old: CALL F01ACF(N,EPS,A,IA,B,IB,Z,L,IFAIL)
New: CALL F01ABF(A,IA,N,B,IB,Z,IFAIL)
```

The number of iterative refinement corrections returned by F01ACF in L is no longer available. The parameter EPS is no longer required.

F01AEF
Withdrawn at Mark 18

```
Old: CALL F01AEF(N,A,IA,B,IB,DL,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = J, N
           A(I,J) = A(J,I)
           B(I,J) = B(J,I)
 10     CONTINUE
        DL(J) = B(J,J)
 20  CONTINUE
     CALL spotrf('L',N,B,IB,INFO)
     IF (INFO.EQ.0) THEN
        CALL ssygst(1,'L',N,A,IA,B,IB,INFO)
     ELSE
        IFAIL = 1
     END IF
     CALL sswap(N,DL,1,B,IB+1)
```

IFAIL is set to 1 if the matrix B is not positive-definite. It is essential to test IFAIL.

F01AFF
Withdrawn at Mark 18

```
Old: CALL F01AFF(N,M1,M2,B,IB,DL,Z,IZ)
New: CALL sswap(N,DL,1,B,IB+1)
     CALL strsm('L','L','T','N',N,M2-M1+1,1.0e0,B,IB,Z(1,M1),IZ)
     CALL sswap(N,DL,1,B,IB+1)
```

F01AGF
Withdrawn at Mark 18

```
Old: CALL F01AGF(N,TOL,A,IA,D,E,E2)
New: CALL ssytrd('L',N,A,IA,D,E(2),TAU,WORK,LWORK,INFO)
     E(1) = 0.0e0
     DO 10 I = 1, N
        E2(I) = E(I)*E(I)
 10  CONTINUE
```

where TAU is a ***real*** array of length at least (N−1), WORK is a ***real*** array of length at least (1) and LWORK is its actual length.

Note that the tridiagonal matrix computed by F08FEF (SSYTRD/DSYTRD) is different from that computed by F01AGF, but it has the same eigenvalues.

F01AHF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01AGF has been replaced by a call to F08FEF (SSYTRD/DSYTRD) as shown above.

```
Old: CALL F01AHF(N,M1,M2,A,IA,E,Z,IZ)
New: CALL sormtr('L','L','N',N,M2-M1+1,A,IA,TAU,Z(1,M1),IZ,WORK,
    +            LWORK,INFO)
```

where WORK is a ***real*** array of length at least (M2−M1+1), and LWORK is its actual length.

F01AJF
Withdrawn at Mark 18

```
Old: CALL F01AJF(N,TOL,A,IA,D,E,Z,IZ)
New: CALL ssytrd('L',N,A,IA,D,E(2),TAU,WORK,LWORK,INFO)
     E(1) = 0.0e0
     CALL F06QFF('L',N,N,A,IA,Z,IZ)
     CALL dorgtr('L',N,Z,IZ,TAU,WORK,LWORK,INFO)
```

where TAU is a ***real*** array of length at least (N−1), WORK is a ***real*** array of length at least (N−1) and LWORK is its actual length.

Note that the tridiagonal matrix T and the orthogonal matrix Q computed by F08FEF (SSYTRD/DSYTRD) and F08FFF (SORGTR/DORGTR) are different from those computed by F01AJF, but they satisfy the same relation $Q^T AQ = T$.

F01AKF
Withdrawn at Mark 18

```
Old: CALL F01AKF(N,K,L,A,IA,INTGER)
New: CALL sgehrd(N,K,L,A,IA,TAU,WORK,LWORK,INFO)
```

where TAU is a ***real*** array of length at least (N−1), WORK is a ***real*** array of length at least (N) and LWORK is its actual length.

Note that the Hessenberg matrix computed by F08NEF (SGEHRD/DGEHRD) is different from that computed by F01AKF, because F08NEF (SGEHRD/DGEHRD) uses orthogonal transformations, whereas F01AKF uses stabilized elementary transformations.

F01ALF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01AKF has been replaced by a call to F08NEF (SGEHRD/DGEHRD) as indicated above.

```
Old: CALL F01ALF(K,L,IR,A,IA,INTGER,Z,IZ,N)
New: CALL sormhr('L','N',N,IR,K,L,A,IA,TAU,Z,IZ,WORK,LWORK,INFO)
```

where WORK is a ***real*** array of length at least (IR) and LWORK is its actual length.

F01AMF
Withdrawn at Mark 18

```
Old: CALL F01AMF(N,K,L,AR,IAR,AI,IAI,INTGER)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
 10     CONTINUE
 20  CONTINUE
     CALL cgehrd(N,K,L,A,IA,TAU,WORK,LWORK,INFO)
```

where A is a ***complex*** array of dimension (IA,N), TAU is a ***complex*** array of length at least (N−1), WORK is a ***complex*** array of length at least (N) and LWORK is its actual length.

Note that the Hessenberg matrix computed by F08NSF (CGEHRD/ZGEHRD) is different from that computed by F01AMF, because F08NSF (CGEHRD/ZGEHRD) uses orthogonal transformations, whereas F01AMF uses stabilized elementary transformations.

F01ANF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01AMF has been replaced by a call to F08NSF (CGEHRD/ZGEHRD) as indicated above.

```
Old: CALL F01ANF(K,L,IR,AR,IAR,AI,IAI,INTGER,ZR,IZR,ZI,IZI,N)
New: CALL cunhmr('L','N',N,IR,K,L,A,IA,TAU,Z,IZ,WORK,LWORK,INFO)
     DO 20 J = 1, IR
        DO 10 I = 1, N
           ZR(I,J) = real(Z(I,J))
           ZI(I,J) = imag(Z(I,J))
 10     CONTINUE
 20  CONTINUE
```

where A is a ***complex*** array of dimension (IA,N), TAU is a ***complex*** array of length at least (N−1), Z is a ***complex*** array of dimension (IZ,IR), WORK is a ***complex*** array of length at least (IR) and LWORK is its actual length.

F01APF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01AKF has been replaced by a call to F08NEF (SGEHRD/DGEHRD) as indicated above.

```
Old: CALL F01APF(N,K,L,INTGER,H,IH,V,IV)
New: CALL F06QFF('L',N,N,H,IH,V,IV)
     CALL sorghr(N,K,L,V,IV,TAU,WORK,LWORK,INFO)
```

where WORK is a ***real*** array of length at least (N), and LWORK is its actual length.

Note that the orthogonal matrix formed by F08NFF (SORGHR/DORGHR) is not the same as the non-orthogonal matrix formed by F01APF. See F01AKF above.

F01ATF
Withdrawn at Mark 18

```
Old: CALL F01ATF(N,IB,A,IA,K,L,D)
New: CALL sgebal('B',N,A,IA,K,L,D,INFO)
```

Note that the balanced matrix returned by F08NHF (SGEBAL/DGEBAL) may be different from that returned by F01ATF.

F01AUF
Withdrawn at Mark 18

```
Old: CALL F01AUF(N,K,L,M,D,Z,IZ)
New: CALL sgebak('B','R',N,K,L,D,M,Z,IZ,INFO)
```

F01AVF
Withdrawn at Mark 18

```
Old: CALL F01AVF(N,IB,AR,IAR,AI,IAI,K,L,D)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
 10     CONTINUE
 20  CONTINUE
     CALL cgebal('B',N,A,IA,K,L,D,INFO)
     DO 20 J = 1, N
        DO 10 I = 1, N
           AR(I,J) = real(A(I,J))
           AI(I,J) = imag(A(I,J))
 10     CONTINUE
 20  CONTINUE
```

where A is a ***complex*** array of dimension (IA,N).

Note that the balanced matrix returned by F08NVF (CGEBAL/ZGEBAL) may be different from that returned by F01AVF.

F01AWF
Withdrawn at Mark 18

```
Old: CALL F01AWF(N,K,L,M,D,ZR,IZR,ZI,IZI)
New: DO 20 J = 1, M
        DO 10 I = 1, N
           Z(I,J) = cmplx(ZR(I,J),ZI(I,J))
 10     CONTINUE
 20  CONTINUE
     CALL cgebak('B','R',N,K,L,D,M,Z,IZ,INFO)
     DO 40 J = 1, M
        DO 30 I = 1, N
```

```
            ZR(I,J) = real(Z(I,J))
            ZI(I,J) = imag(Z(I,J))
 30      CONTINUE
 40   CONTINUE
```

where Z is a ***complex*** array of dimension (IZ,M).

F01AXF
Withdrawn at Mark 18

```
Old: CALL F01AXF(M,N,QR,IQR,ALPHA,IPIV,Y,E,IFAIL)
New: CALL sgeqpf(M,N,QR,IQR,IPIV,Y,WORK,INFO)
     CALL scopy(N,QR,IQR+1,ALPHA,1)
```

where WORK is a ***real*** array of length at least (3*N).

Note that the details of the Householder matrices returned by F08BEF (SGEQPF/DGEQPF) are different from those returned by F01AXF, but they determine the same orthogonal matrix Q.

F01AYF
Withdrawn at Mark 18

```
Old: CALL F01AYF(N,TOL,A,IA,D,E,E2)
New: CALL ssptrd('U',N,A,D,E(2),TAU,INFO)
     E(1) = 0.0e0
     DO 10 I = 1, N
        E2(I) = E(I)*E(I)
  10 CONTINUE
```

where TAU is a ***real*** array of length at least (N−1).

F01AZF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01AYF has been replaced by a call to F08GEF (SSPTRD/DSPTRD) as shown above.

```
Old: CALL F01AZF(N,M1,M2,A,IA,Z,IZ)
New: CALL sopmtr('L','U','N',N,M2-M1+1,A,TAU,Z(1,M1),IZ,WORK,INFO)
```

where WORK is a ***real*** array of length at least (M2−M1+1).

F01BCF
Withdrawn at Mark 18

```
Old: CALL F01BCF(N,TOL,AR,IAR,AI,IAI,D,E,WK1,WK2)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
 10     CONTINUE
 20  CONTINUE
     CALL chetrd('L',N,A,IA,D,E(2),TAU,WORK,LWORK,INFO)
     E(1) = 0.0e0
     CALL cungtr('L',N,A,IA,TAU,WORK,LWORK,INFO)
     DO 40 J = 1, N
        DO 30 I = 1, N
           AR(I,J) = real(A(I,J))
           AI(I,J) = imag(A(I,J))
 30     CONTINUE
 40  CONTINUE
```

where A is a ***complex*** array of dimension (IA,N), TAU is a ***complex*** array of length at least (N−1), WORK is a ***complex*** array of length at least (N−1), and LWORK is its actual length.

Note that the tridiagonal matrix T and the unitary matrix Q computed by F08FSF (CHETRD/ZHETRD) and F08FTF (CUNGTR/ZUNGTR) are different from those computed by F01BCF, but they satisfy the same relation $Q^H AQ = T$.

F01BDF
Withdrawn at Mark 18

```
Old: CALL F01BDF(N,A,IA,B,IB,DL,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = J, N
           A(I,J) = A(J,I)
           B(I,J) = B(J,I)
 10     CONTINUE
        DL(J) = B(J,J)
 20  CONTINUE
     CALL spotrf('L',N,B,IB,INFO)
     IF (INFO.EQ.0) THEN
        CALL ssygst(2,'L',N,A,IA,B,IB,INFO)
     ELSE
        IFAIL = 1
     END IF
     CALL sswap(N,DL,1,B,IB+1)
```

IFAIL is set to 1 if the matrix B is not positive-definite. It is essential to test IFAIL.

F01BEF
Withdrawn at Mark 18

```
Old: CALL F01BEF(N,M1,M2,B,IB,DL,V,IV)
New: CALL sswap(N,DL,1,B,IB+1)
     CALL strmm('L','L','N','N',N,M2-M1+1,1.0e0,B,IB,V(1,M1),IV)
     CALL sswap(N,DL,1,B,IB+1)
```

F01BNF
Withdrawn at Mark 17

```
Old: CALL F01BNF(N,A,IA,P,IFAIL)
New: CALL cpotrf('Upper',N,A,IA,IFAIL)
```

where, before the call, array A contains the upper triangle of the matrix to be factorized rather than the lower triangle (note that the elements of the upper triangle are the complex conjugates of the elements of the lower triangle). The ***real*** array P is no longer required; the upper triangle of A is overwritten by the upper triangular factor U, including the diagonal elements (which are not reciprocated).

F01BPF
Withdrawn at Mark 17

```
Old: CALL F01BPF(N,A,IA,V,IFAIL)
New: CALL cpotrf('Upper',N,A,IA,IFAIL)
     CALL cpotri('Upper',N,A,IA,IFAIL)
```

where, before the calls, the upper triangle of the matrix to be inverted must be contained in rows 1 to N of A, rather than the lower triangle being in rows 2 to $\mathrm{N}+1$ (note that the elements of the upper triangle are the complex conjugates of the elements of the lower triangle). The workspace vector V is no longer required.

F01BQF
Withdrawn at Mark 16

The replacement routines do not have exactly the same functionality as F01BQF; if this functionality is genuinely required, please contact NAG.

(a) where the symmetric matrix is known to be positive-definite (if the matrix is in fact not positive-definite, the replacement routine will return a positive value in IFAIL)

```
Old: CALL F01BQF(N,EPS,RL,IRL,D,IFAIL)
New: CALL spptrf('Lower',N,RL,IFAIL)
```

(b) where the matrix is not positive-definite (the replacement routine forms an LDL^T factorization where D is block diagonal, rather than a Cholesky factorization)

```
Old: CALL F01BQF(N,EPS,RL,IRL,D,IFAIL)
New: CALL ssptrf('Lower',N,RL,IPIV,IFAIL)
```

For the replacement calls in both (a) and (b), the array RL must now hold the complete lower triangle of the symmetric matrix, including the diagonal elements, which are no longer required to be stored in the redundant array D. The declared dimension of RL must be increased from at least $N(N-1)/2$ to at least $N(N+1)/2$. It is important to note that for the calls of F07GDF (SPPTRF/DPPTRF) and F07PDF (SSPTRF/DSPTRF), the lower triangle of the matrix must be stored packed by column instead of by row. The dimension parameter IRL is no longer required. For the call of F07PDF (SSPTRF/DSPTRF), the INTEGER array IPIV of length N must be supplied.

F01BTF
Withdrawn at Mark 18

```
Old: CALL F01BTF(N,A,IA,P,DP,IFAIL)
New: CALL sgetrf(N,N,A,IA,IPIV,IFAIL)
```

where IPIV is an INTEGER array of length N which holds the indices of the pivot elements, and the array P is no longer required. It may be important to note that after a call of F07ADF (SGETRF/DGETRF), A is overwritten by the upper triangular factor U and the off-diagonal elements of the unit lower triangular factor L, whereas the factorization returned by F01BTF gives U the unit diagonal. The permutation determinant DP returned by F01BTF is not computed by F07ADF (SGETRF/DGETRF). If this value is required, it may be calculated after a call of F07ADF (SGETRF/DGETRF) by code similar to the following:

```
      DP = 1.0e0
      DO 10 I = 1, N
         IF (I.NE.IPIV(I)) DP = -DP
10 CONTINUE
```

F01BWF
Withdrawn at Mark 18

```
Old: CALL F01BWF(N,M1,A,IA,D,E)
New: CALL ssbtrd('N','U',N,M1-1,A,IA,D,E(2),Q,1,WORK,INFO)
     E(1) = 0.0e0
```

where Q is a dummy ***real*** array of length (1) (not used in this call), and WORK is a ***real*** array of length at least (N).

Note that the tridiagonal matrix computed by F08HEF (SSBTRD/DSBTRD) is different from that computed by F01BWF, but it has the same eigenvalues.

F01BXF
Withdrawn at Mark 17

```
Old: CALL F01BXF(N,A,IA,P,IFAIL)
New: CALL spotrf('Upper',N,A,IA,IFAIL)
```

where, before the call, array A contains the upper triangle of the matrix to be factorized rather than the lower triangle. The array P is no longer required; the upper triangle of A is overwritten by the upper triangular factor U, including the diagonal elements (which are not reciprocated).

F01CAF
Withdrawn at Mark 14

```
Old: CALL F01CAF(A,M,N,IFAIL)
New: CALL F06QHF('General',M,N,0.0e0,0.0e0,A,M)
```

F01CBF
Withdrawn at Mark 14

```
Old: CALL F01CBF(A,M,N,IFAIL)
New: CALL F06QHF('General',M,N,0.0e0,1.0e0,A,M)
```

F01CDF
Withdrawn at Mark 15

```
Old: CALL F01CDF(A,B,C,M,N,IFAIL)
New: CALL F01CTF('N','N',M,N,1.0e0,B,M,1.0e0,C,M,A,M,IFAIL)
```

F01CEF
Withdrawn at Mark 15

```
Old: CALL F01CEF(A,B,C,M,N,IFAIL)
New: CALL F01CTF('N','N',M,N,1.0e0,B,M,-1.0e0,C,M,A,M,IFAIL)
```

F01CFF
Withdrawn at Mark 14

```
Old: CALL F01CFF(A,MA,NA,P,Q,B,MB,NB,M1,M2,N1,N2,IFAIL)
New: CALL F06QFF('General',M2-M1+1,N2-N1+1,B(M1,N1),MB,A(P,Q),MA)
```

F01CGF
Withdrawn at Mark 15

```
Old: CALL F01CGF(A,MA,NA,P,Q,B,MB,NB,M1,M2,N1,N2,IFAIL)
New: CALL F01CTF('N','N',M2-M1+1,N2-N1+1,1.0e0,A(P,Q),MA,1.0e0,
    +            B(M1,N1),MB,A(P,Q),MA,IFAIL)
```

F01CHF
Withdrawn at Mark 15

```
Old: CALL F01CHF(A,MA,NA,P,Q,B,MB,NB,M1,M2,N1,N2,IFAIL)
New: CALL F01CTF('N','N',M2-M1+1,N2-N1+1,1.0e0,A(P,Q),MA,-1.0e0,
    +            B(M1,N1),MB,A(P,Q),MA,IFAIL)
```

F01CLF
Withdrawn at Mark 16

```
Old: CALL F01CLF(A,B,C,N,P,M,IFAIL)
New: CALL sgemm('N','T',N,P,M,1.0e0,B,N,C,P,0.0e0,A,N)
```

F01CMF
Withdrawn at Mark 14

```
Old: CALL F01CMF(A,LA,B,LB,M,N)
New: CALL F06QFF('General',M,N,A,LA,B,LB)
```

F01CNF
Withdrawn at Mark 13

```
Old: CALL F01CNF(V,M,A,LA,I)
New: CALL scopy(M,V,1,A(I,1),LA)
```

F01CPF
Withdrawn at Mark 13

```
Old: CALL F01CPF(A,B,N)
New: CALL scopy(N,A,1,B,1)
```

F01CQF
Withdrawn at Mark 13

```
Old: CALL F01CQF(A,N)
New: CALL F06FBF(N,0.0e0,A,1)
```

F01CSF
Withdrawn at Mark 13

```
Old: CALL F01CSF(A,LA,B,N,C)
New: CALL sspmv('U',N,1.0e0,A,B,1,0.0e0,C,1)
```

F01DAF
Withdrawn at Mark 13

```
Old: F01DAF(L,M,C1,IRA,ICB,A,IA,B,IB,N)
New: C1 + sdot(M-L+1,A(IRA,L)IA,B(L,ICB),1)
```

F01DBF
Withdrawn at Mark 13

```
Old: D = F01DBF(L,M,C1,IRA,ICB,A,IA,B,IB,N)
New: CALL X03AAF(A(IRA,L),(M-L)*IA+1,B(L,ICB),M-L+1,IA,1,C1,0.0e0,D,
    +            D2,.TRUE.,IFAIL)
```

(here D2 is a new ***real*** variable whose value is not used).

F01DCF
Withdrawn at Mark 13

```
Old: CALL F01DCF(L,M,CX,IRA,ICB,A,IA,B,IB,N,CR,CI)
New: DX = CX - cdotu(M-L+1,A(IRA,L),IA,B(L,ICB),1)
     CR = real(DX)
     CI = imag(DX)
```

(here DX is a new ***complex*** variable).

F01DDF
Withdrawn at Mark 13

```
Old: CALL F01DDF(L,M,CX,IRA,ICB,A,IA,B,IB,N,CR,CI)
New: CALL X03ABF(A(IRA,L),(M-L)*IA+1,B(L,ICB),M-L+1,IA,1,-CX,DX,
    +            .TRUE.,IFAIL)
     CR = -real(DX)
     CI = -imag(DX)
```

(here DX is a new ***complex*** variable).

F01DEF
Withdrawn at Mark 14

```
Old: F01DEF(A,B,N)
New: sdot(N,A,1,B,1)
```

F01LBF
Withdrawn at Mark 18

```
Old: CALL F01LBF(N,M1,M2,A,IA,AL,IL,IN,IV,IFAIL)
New: CALL sgbtrf(N,N,M1,M2,A,IA,IN,IFAIL)
```

where the size of array A must now have a leading dimension IA of at least $2 \times \text{M1}+\text{M2}+1$. The array AL, its associated dimension parameter IL, and the parameter IV are not required for F07BDF (SGBTRF/DGBTRF) because this routine overwrites A by both the L and U factors. The scheme by which the matrix is packed into the array is completely different from that used by F01LBF; the relevant routine document should be consulted for details.

F01LZF
Withdrawn at Mark 15

```
Old: CALL F01LZF(N,A,NRA,C,NRC,WANTB,B,WANTQ,WANTY,Y,NRY,LY,WANTZ,Z,
    +            NRZ,NCZ,D,E,WORK1,WORK2,IFAIL)
New: CALL sgebrd(N,N,A,NRA,D,E(2),TAUQ,TAUP,WORK1,LWORK,INFO)
     IF (WANTB) THEN
        CALL sormbr('Q','L','T',N,1,NA,NRA,TAUQ,B,N,WORK1,LWORK,INFO)
     ELSE IF (WANTQ) THEN
        CALL sorgbr('Q',N,N,N,A,NRA,TAUQ,WORK,LWORK,INFO)
     ELSE IF (WANTY) THEN
        CALL sormbr('Q','R','N',LY,N,N,A,NRA,TAUQ,Y,NRY,WORK1,LWORK,
    +               INFO)
     ELSE IF (WANTZ) THEN
        CALL sormbr('P','L','T',N,NCZ,N,A,NRA,TAUP,Z,NRZ,WORK1,LWORK,
    +               INFO)
     END IF
```

where TAUQ and TAUP are real arrays of length at least (N) and LWORK is the actual length of WORK1. The parameter WORK2 is no longer required.

F01MAF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

Existing programs should be modified to call F11JAF. The interfaces are significantly different and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine document.

F01NAF
Withdrawn at Mark 17

```
Old: CALL F01NAF(N,ML,MU,A,NRA,TOL,IN,SCALE,IFAIL)
New: CALL cgbtrf(N,N,ML,MU,A,NRA,IN,IFAIL)
```

where the parameter TOL and array SCALE are no longer required. The input matrix must be stored using the same scheme as for F01NAF, except in rows ML + 1 to $2 \times$ ML + MU + 1 of A instead of rows 1 to ML + MU + 1. In F07BRF(CGBTRF/ZGBTRF), the value returned in IN(N) has no significance as an indicator of near-singularity of the matrix.

F01QAF
Withdrawn at Mark 15

```
Old: CALL F01QAF(M,N,A,NRA,C,NRC,Z,IFAIL)
New: CALL sgeqrf(M,N,A,NRA,Z,WORK,LWORK,INFO)
```

where WORK is a real array of length at least (LWORK). The parameters C and NRC are no longer required.

Note that the representation of the matrix Q is not identical, but subsequent calls to routines F08AFF (SORGQR/DORGQR) and F08AGF (SORMQR/DORMQR) may be used to obtain Q explicitly and to transform by Q or Q^T respectively.

F01QBF
Withdrawn at Mark 15

```
Old: CALL F01QBF(M,N,A,NRA,C,NRC,WORK,IFAIL)
New: CALL F06QFF('General',M,N,A,NRA,C,NRC)
     CALL F01QJF(M,N,C,NRC,WORK,IFAIL)
```

The call to F06QFF simply copies the M by N part of A to C. This may be omitted if it is desired to use the same arrays for A and C. Note that the representation of the orthogonal matrix Q is not identical, but following F01QJF routine F01QKF may be used to form Q.

F01QCF
Withdrawn at Mark 18

```
Old: CALL F01QCF(M,N,A,LDA,ZETA,IFAIL)
New: CALL sgeqrf(M,N,A,LDA,ZETA,WORK,LWORK,INFO)
```

where WORK is a ***real*** array of length at least (N), and LWORK is its actual length.

The subdiagonal elements of A and the elements of ZETA returned by F08AEF (SGEQRF/DGEQRF) are not the same as those returned by F01QCF. Subsequent calls to F01QDF or F01QEF must also be replaced by calls to F08AGF (SORMQR/DORMQR) or F08AFF (SORGQR/DORGQR) as shown below.

F01QDF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01QCF has been replaced by a call to F08AEF (SGEQRF/DGEQRF) as shown above. It also assumes that the 2nd argument of F01QDF (WHERET) is 'S', which is appropriate if the contents of A and ZETA have not been changed after the call of F01QCF.

```
Old: CALL F01QDF(TRANS,'S',M,N,A,LDA,ZETA,NCOLB,B,LDB,WORK,IFAIL)
New: CALL sormqr('L',TRANS,M,NCOLB,N,A,LDA,ZETA,B,LDB,WORK,LWORK,INFO)
```

where LWORK is the actual length of WORK.

F01QEF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01QCF has been replaced by a call to F08AEF (SGEQRF/DGEQRF) as shown above. It also assumes that the 1st argument of F01QEF (WHERET) is 'S', which is appropriate if the contents of A and ZETA have not been changed after the call of F01QCF.

```
Old: CALL F01QEF('S',M,N,NCOLQ,A,LDA,ZETA,WORK,IFAIL)
New: CALL sorgqr(M,NCOLQ,N,A,LDA,ZETA,WORK,LWORK,INFO)
```

where LWORK is the actual length of WORK.

F01QFF
Withdrawn at Mark 18

The following replacement assumes that the 1st argument of F01QFF (PIVOT) is 'C'. There is no direct replacement if PIVOT = 'S'.

```
Old: CALL F01QFF('C',M,N,A,LDA,ZETA,PERM,WORK,IFAIL)
New: DO 10 I = 1, N
        PERM(I) = 0
  10 CONTINUE
     CALL sgeqpf(M,N,A,LDA,PERM,ZETA,WORK,INFO)
```

where WORK is a ***real*** array of length at least (3*N) (F01QFF only requires WORK to be of length (2*N)).

The subdiagonal elements of A and the elements of ZETA returned by F08BEF (SGEQPF/DGEQPF) are not the same as those returned by F01QFF. Subsequent calls to F01QDF or F01QEF must also be replaced by calls to F08AGF (SORMQR/DORMQR) or F08AFF (SORGQR/DORGQR) as shown above. Note also that the array PERM returned by F08BEF (SGEQPF/DGEQPF) holds details of the interchanges in a different form than that returned by F01QFF.

F01RCF
Withdrawn at Mark 18

```
Old: CALL F01RCF(M,N,A,LDA,THETA,IFAIL)
New: CALL cgeqrf(M,N,A,LDA,THETA,WORK,LWORK,INFO)
```

where WORK is a ***complex*** array of length at least (N), and LWORK is its actual length.

The subdiagonal elements of A and the elements of THETA returned by F08ASF (CGEQRF/ZGEQRF) are not the same as those returned by F01RCF. Subsequent calls to F01RDF or F01REF must also be replaced by calls to F08AUF (CUNMQR/ZUNMQR) or F08ATF (CUNGQR/ZUNGQR) as shown below.

F01RDF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01RCF has been replaced by a call to F08ASF (CGEQRF/ZGEQRF) as shown above. It also assumes that the 2nd argument of F01RDF (WHERET) is 'S', which is appropriate if the contents of A and THETA have not been changed after the call of F01RCF.

```
Old: CALL F01RDF(TRANS,'S',M,N,A,LDA,THETA,NCOLB,B,LDB,WORK,IFAIL)
New: CALL cunmqr('L',TRANS,M,NCOLB,N,A,LDA,THETA,B,LDB,WORK,LWORK,
    +            INFO)
```

where LWORK is the actual length of WORK.

F01REF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F01RCF has been replaced by a call to F08ASF (CGEQRF/ZGEQRF) as shown above. It also assumes that the 1st argument of F01REF (WHERET) is 'S', which is appropriate if the contents of A and THETA have not been changed after the call of F01RCF.

```
Old: CALL F01REF('S',M,N,NCOLQ,A,LDA,THETA,WORK,IFAIL)
New: CALL cungqr(M,NCOLQ,N,A,LDA,THETA,WORK,LWORK,INFO)
```

where LWORK is the actual length of WORK.

F01RFF
Withdrawn at Mark 18

The following replacement assumes that the 1st argument of F01RFF (PIVOT) is 'C'. There is no direct replacement if PIVOT = 'S'.

```
Old: CALL F01RFF('C',M,N,A,LDA,THETA,PERM,WORK,IFAIL)
New: DO 10 I = 1, N
        PERM(I) = 0
  10 CONTINUE
     CALL cgeqpf(M,N,A,LDA,PERM,THETA,CWORK,WORK,INFO)
```

where CWORK is a ***complex*** array of length at least (N).

The subdiagonal elements of A and the elements of THETA returned by F08BSF (CGEPQF/ZGEPQF) are not the same as those returned by F01RFF. Subsequent calls to F01RDF or F01REF must also be replaced by calls to F08AUF (CUNMQR/ZUNMQR) or F08ATF (CUNGQR/ZUNGQR) as shown above. Note also that the array PERM returned by F08BSF (CGEPQF/ZGEPQF) holds details of the interchanges in a different form than that returned by F01RFF.

F02 – Eigenvalues and Eigenvectors

Notes:

1. Replacement routines require complex matrices to be stored in ***complex*** arrays, whereas most of the corresponding old routines require the real and imaginary parts to be stored separately in two ***real*** arrays.
2. Replacement routines for computing eigenvectors may scale the eigenvectors in a different manner from the old routines, and hence at first glance the eigenvectors may appear to disagree completely; they may indeed be different, but they are equally acceptable as eigenvectors; some replacement routines may also return the eigenvalues (and the corresponding eigenvectors) in a different order.

3. Replacement routines in Chapter F07 and Chapter F08 have a parameter INFO, which has a different specification to the usual NAG error-handling parameter IFAIL. See the F07 or F08 Chapter Introduction for details.

F02AAF
Withdrawn at Mark 18

```
Old: CALL F02AAF(A,IA,N,R,E,IFAIL)
New: CALL F02FAF('N','L',N,A,IA,R,WORK,LWORK,IFAIL)
```

where WORK is a ***real*** array of length at least (3*N) and LWORK is its actual length.

F02ABF
Withdrawn at Mark 18

```
Old: CALL F02ABF(A,IA,B,IB,N,R,V,IV,E,IFAIL)
New: CALL F06QFF('L',N,N,A,IA,V,IV)
     CALL F02FAF('V','L',N,V,IV,R,WORK,LWORK,IFAIL)
```

where WORK is a ***real*** array of length at least (3*N) and LWORK is its actual length. If F02ABF was called with the same array supplied for V and A, then the call to F06QFF (which copies A to V) may be omitted.

F02ADF
Withdrawn at Mark 18

```
Old: CALL F02ADF(A,IA,B,IB,N,R,DE,IFAIL)
New: CALL F02FDF(1,'N','U',N,A,IA,B,IB,R,WORK,LWORK,IFAIL)
```

where WORK is a ***real*** array of length at least (3*N) and LWORK is its actual length.

Note that the call to F02FDF will overwrite the upper triangles of the arrays A and B and leave the subdiagonal elements unchanged, whereas the call to F02ADF overwrites the lower triangle and leaves the elements above the diagonal unchanged.

F02AEF
Withdrawn at Mark 18

```
Old: CALL F02AEF(A,IA,N,R,V,IV,DL,E,IFAIL)
New: CALL F06QFF('U',N,N,A,IA,V,IV)
     CALL F02FDF(1,'V','U',N,V,IV,B,IB,R,WORK,LWORK,IFAIL)
```

where WORK is a ***real*** array of length at least (3*N) and LWORK is its actual length.

Note that the call to F02FDF will overwrite the upper triangle of the array B and leave the subdiagonal elements unchanged, whereas the call to F02ADF overwrites the lower triangle and leaves the elements above the diagonal unchanged. The call to F06QFF copies A to V, so A is left unchanged. If F02AEF was called with the same array supplied for V and A, then the call to F06QFF may be omitted.

F02AFF
Withdrawn at Mark 18

```
Old: CALL F02AFF(A,IA,N,RR,RI,INTGER,IFAIL)
New: CALL F02EBF('N',N,A,IA,RR,RI,VR,1,VI,1,WORK,LWORK,IFAIL)
```

where VR and VI are dummy arrays of length (1) (not used in this call), WORK is a ***real*** array of length at least (4*N) and LWORK is its actual length; the iteration counts (returned by F02AFF in the array INTGER) are not available from F02EBF.

F02AGF
Withdrawn at Mark 18

```
Old: CALL F02AGF(A,IA,N,RR,RI,VR,IVR,VI,IVI,INTGER,IFAIL)
New: CALL F02EBF('V',N,A,IA,RR,RI,VR,IVR,VI,IVI,WORK,LWORK,IFAIL)
```

where WORK is a ***real*** array of length at least (4*N) and LWORK is its actual length; the iteration counts (returned by F02AGF in the array INTGER) are not available from F02EBF.

F02AJF
Withdrawn at Mark 18

```
Old: CALL F02AJF(AR,IAR,AI,IAI,N,RR,RI,INTGER,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
 10     CONTINUE
 20  CONTINUE
     CALL F02GBF('N',N,A,IA,R,V,1,RWORK,WORK,LWORK,IFAIL)
     DO 30 I = 1, N
        RR(I) = real(R(I))
        RI(I) = imag(R(I))
 30  CONTINUE
```

where A is a ***complex*** array of dimension (IA,N), R is a ***complex*** array of dimension (N), V is a dummy ***complex*** array of length (1) (not used in this call), RWORK is a ***real*** array of length at least (2*N), WORK is a ***complex*** array of length at least (2*N) and LWORK is its actual length.

F02AKF
Withdrawn at Mark 18

```
Old: CALL F02AKF(AR,IAR,AI,IAI,N,RR,RI,VR,IVR,VI,IVI,INTGER,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
  10    CONTINUE
  20 CONTINUE
     CALL F02GBF('V',N,A,IA,R,V,IV,RWORK,WORK,LWORK,IFAIL)
     DO 40 J = 1, N
        RR(J) = real(R(J))
        RI(J) = imag(R(J))
        DO 30 I = 1, N
           VR(I,J) = real(V(I,J))
           VI(I,J) = imag(V(I,J))
  30    CONTINUE
  40 CONTINUE
```

where A is a ***complex*** array of dimension (IA,N), R is a ***complex*** array of length (N), V is a ***complex*** array of dimension (IV,N), RWORK is a ***real*** array of length at least (2*N), WORK is a ***complex*** array of length at least (2*N) and LWORK is its actual length.

F02AMF
Withdrawn at Mark 18

```
Old: CALL F02AMF(N,EPS,D,E,V,IV,IFAIL)
New: CALL ssteqr('V',N,D,E(2),V,IV,WORK,INFO)
```

where WORK is a ***real*** array of length at least (2*(N−1)).

F02ANF
Withdrawn at Mark 18

```
Old: CALL F02ANF(N,EPS,HR,IHR,HI,IHI,RR,RI,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           H(I,J) = cmplx(HR(I,J),HI(I,J))
  10    CONTINUE
  20 CONTINUE
     CALL chseqr('E','N',N,1,N,H,IH,R,Z,1,WORK,1,INFO)
     DO 30 I = 1, N
        RR(I) = real(R(I))
        RI(I) = imag(R(I))
  30 CONTINUE
```

where H is a ***complex*** array of dimension (IH,N), R is a ***complex*** array of length (N), Z is a dummy ***complex*** array of length (1) (not used in this call), and WORK is a ***complex*** array of length at least (N).

F02APF
Withdrawn at Mark 18

```
Old: CALL F02APF(N,EPS,H,IH,RR,RI,ICNT,IFAIL)
New: CALL shseqr('E','N',N,1,N,H,IH,RR,RI,Z,1,WORK,1,INFO)
```

where Z is a dummy ***real*** array of length (1) (not used in this call), and WORK is a ***real*** array of length at least (N); the iteration counts (returned by F02APF in the array ICNT) are not available from F08PEF (SHSEQR/DHSEQR).

F02AQF
Withdrawn at Mark 18

```
Old: CALL F02AQF(N,K,L,EPS,H,IH,V,IV,RR,RI,INTGER,IFAIL)
New: CALL shseqr('S','V',N,K,L,H,IH,RR,RI,V,IV,WORK,1,INFO)
     CALL strevc('R','O',SELECT,N,H,IH,V,IV,V,IV,N,M,WORK,INFO)
```

where SELECT is a dummy logical array of length (1) (not used in this call), and WORK is a ***real*** array of length at least (N); the iteration counts (returned by F02AQF in the array INTGER) are not available from F08PEF (SHSEQR/DHSEQR); M is an integer which is set to N by F08QKF (STREVC/DTREVC).

F02ARF
Withdrawn at Mark 18

```
Old: CALL F02ARF(N,K,L,EPS,INTGER,HR,IHR,HI,IHI,RR,RI,VR,IVR,VI,
    +            IVI, IFAIL)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           H(I,J) = cmplx(HR(I,J),HI(I,J))
 10     CONTINUE
 20  CONTINUE
     CALL chseqr('S','V',N,K,L,H,IH,R,V,IV,WORK,1,INFO)
     CALL ctrevc('R','O',SELECT,N,H,IH,V,IV,V,IV,N,M,WORK,INFO)
     DO 40 J = 1, N
        RR(J) = real(R(J))
        RI(J) = imag(R(J))
        DO 30 I = 1, N
           VR(I,J) = real(V(I,J))
           VI(I,J) = imag(V(I,J))
 30     CONTINUE
 40  CONTINUE
```

where H is a ***complex*** array of dimension (IH,N), R is a ***complex*** array of length (N), V is a ***complex*** array of dimension (IV,N), WORK is a ***complex*** array of length at least (2*N) and RWORK is a ***real*** array of length at least (N); M is an integer which is set to N by F08QXF (CTREVC/ZTREVC).

If F02ARF was preceded by a call to F01AMF to reduce a full complex matrix to Hessenberg form, then the call to F01AMF must also be replaced by calls to F08NSF (CGEHRD/ZGEHRD) and F08NTF (CUNGHR/ZUNGHR).

F02AVF
Withdrawn at Mark 18

```
Old: CALL F02AVF(N,EPS,D,E,IFAIL)
New: CALL ssterf(N,D,E(2),INFO)
```

F02AWF
Withdrawn at Mark 18

```
Old: CALL F02AWF(AR,IAR,AI,IAI,N,R,WK1,WK2,WK3,IFAIL)
```

```
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
  10    CONTINUE
  20 CONTINUE
     CALL F02HAF('N','L',N,A,IA,R,RWORK,WORK,LWORK,IFAIL)
```

where A is a ***complex*** array of dimension (IA,N), RWORK is a ***real*** array of length at least (3*N), WORK is a ***complex*** array of length at least (2*N) and LWORK is its actual length.

F02AXF
Withdrawn at Mark 18

```
Old: CALL F02AXF(AR,IAR,AI,IAI,N,R,VR,IVR,VI,IVI,WK1,WK2,WK3,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
  10    CONTINUE
  20 CONTINUE
     CALL F06TFF('L',N,N,A,IA,V,IV)
     CALL F02HAF('V','L',N,V,IV,R,RWORK,WORK,LWORK,IFAIL)
     DO 40 J = 1, N
        DO 30 I = 1, N
           VR(I,J) = real(V(I,J))
           VI(I,J) = imag(V(I,J))
  30    CONTINUE
  40 CONTINUE
```

where A is a ***complex*** array of dimension (IA,N), V is a ***complex*** array of dimension (IV,N), RWORK is a ***real*** array of length at least (3*N), WORK is a ***complex*** array of length at least (2*N) and LWORK is its actual length. If F02AXF was called with the same arrays supplied for VR and AR and for VI and AI, then the call to F06TFF (which copies A to V) may be omitted.

F02AYF
Withdrawn at Mark 18

```
Old: CALL F02AYF(N,EPS,D,E,VR,IVR,VI,IVI,IFAIL)
New: CALL csteqr('V',N,D,E(2),V,IV,WORK,INFO)
     DO 40 J = 1, N
        DO 30 I = 1, N
           VR(I,J) = real(V(I,J))
           VI(I,J) = imag(V(I,J))
 30     CONTINUE
 40  CONTINUE
```

where V is a ***complex*** array of dimension (IV,N), and WORK is a ***real*** array of length at least (2*(N−1)).

F02BBF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

```
Old: CALL F02BBF(A,IA,N,ALB,UB,M,MM,R,V,IV,D,E,E2,X,G,C,
    +            ICOUNT,IFAIL)
New: CALL F02FCF('Vectors','Value','Lower',N,A,IA,ALB,UB,0,0,
    +            M,MM,R,V,IV,WORK,LWORK,IWORK,IFAIL)
```

where R must have dimension (N), WORK is a ***real*** array of length at least (8*N), LWORK is its actual length, and IWORK is an integer array of length at least (5*N). Note that in the call to F02BBF R needs only to be of dimension (M).

F02BCF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

```
Old: CALL F02BCF(A,IA,N,ALB,UB,M,MM,RR,RI,VR,IVR,VI,IVI,
    +            INTGER,ICNT,C,B,IB,U,V,IFAIL)
New: CALL F02ECF('Moduli',N,A,IA,ALB,UB,M,MM,RR,RI,VR,IVR,
    +            VI,IVI,WORK,LWORK,ICNT,C,IFAIL)
```

where WORK is a ***real*** array of length at least (N*(N+4)) and LWORK is its actual length.

F02BDF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

```
Old: CALL F02BDF(AR,IAR,AI,IAI,N,ALB,UB,M,MM,RR,RI,VR,IVR,
    +            VI,IVI,INTGER,C,BR,IBR,BI,IBI,U,V,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = 1, N
           A(I,J) = cmplx(AR(I,J),AI(I,J))
  10    CONTINUE
  20 CONTINUE
     CALL F02GCF('Moduli', N, A, IA, ALB, UB, M, MM, R, V, IV, WORK,
    +             LWORK, RWORK, INTGER, C, IFAIL)
     DO 30 I = 1, N
        RR(I) = real(R(I))
        RI(I) = imag(R(I))
  30 CONTINUE
     DO 50 J = 1, MM
        DO 40 I = 1, N
           VR(I,J) = real(V(I,J))
           VI(I,J) = imag(V(I,J))
  40    CONTINUE
  50 CONTINUE
```

where A is a ***complex*** array of dimension (IA,N), R is a ***complex*** array of dimension (N), V is a ***complex*** array of dimension (IV,M), WORK is a ***complex*** array of length at least (N*(N+2)), LWORK is its actual length, and RWORK is a ***real*** array of length at least (2*N).

F02BEF
Withdrawn at Mark 18

```
Old: CALL F02BEF(N,D,ALB,UB,EPS,EPS1,E,E2,M,MM,R,V,IV,ICOUNT,X,C,
    +            IFAIL)
New: CALL sstebz('V','B',N,ALB,UB,0,0,EPS1,D,E(2),MM,NSPLIT,R,IBLOCK,
    +            ISPLIT,X,IWORK,INFO)
     CALL sstein(N,D,E(2),MM,R,IBLOCK,ISPLIT,V,IV,X,IWORK,IFAILV,INFO)
```

where NSPLIT is an integer variable, IBLOCK, ISPLIT and IFAILV are integer arrays of length at least (N), and IWORK is an integer array of length at least (3*N).

F02BFF
Withdrawn at Mark 18

```
Old: CALL F02BFF(D,E,E2,N,M1,M2,MM12,EPS1,EPS,EPS2,IZ,R,WU)
New: CALL sstebz('I','E',N,0.0e0,0.0e0,M1,M2,EPS1,D,E(2),M,
    +            NSPLIT,R,IBLOCK,ISPLIT,WORK,IWORK,INFO)
```

where M and NSPLIT are integer variables, IBLOCK and ISPLIT are integer arrays of length at least (N), WORK is a ***real*** array of length at least (4*N), and IWORK is an integer array of length at least (3*N).

F02BKF
Withdrawn at Mark 18

```
Old: CALL F02BKF(N,M,H,IH,RI,C,RR,V,IV,B,IB,U,W,IFAIL)
New: CALL shsein('R','Q','N',C,N,H,IH,RR,RI,V,IV,V,IV,M,M2,B,IFAILR,
    +            IFAILR,INFO)
```

where M2 is an integer variable, and IFAILR is an integer array of length at least (N).

Note that the array C may be modified by F08PKF (SHSEIN/DHSEIN) if there are complex conjugate pairs of eigenvalues.

F02BLF
Withdrawn at Mark 18

```
Old: CALL F02BLF(N,M,HR,IHR,HI,IHI,RI,C,RR,VR,IVR,VI,IVI,BR,IBR,BI,
    +            IBI,U,W,IFAIL)
New: DO 20 J = 1, N
        R(J) = cmplx(RR(J),RI(J))
        DO 10 I = 1, N
           H(I,J) = cmplx(HR(I,J),HI(I,J))
  10    CONTINUE
  20 CONTINUE
     CALL chsein('R','Q','N',C,N,H,IH,R,V,IV,V,IV,M,M2,WORK,RWORK,
    +            IFAILR,IFAILR,INFO)
     DO 30 I = 1, N
        RR(I) = real(R(I))
  30 CONTINUE
     DO 50 J = 1, M
        DO 40 I = 1, N
           VR(I,J) = real(V(I,J))
           VI(I,J) = imag(V(I,J))
  40    CONTINUE
  50 CONTINUE
```

where H is a ***complex*** array of dimension (IH,N), R is a ***complex*** array of length (N), V is a ***complex*** array of dimension (IV,M), M2 is an integer variable, WORK is a ***complex*** array of length at least (N*N), RWORK is a ***real*** array of length at least (N), and IFAILR is an integer array of length at least (N).

F02SWF
Withdrawn at Mark 18

The following replacement ignores the triangular structure of A, and therefore references the subdiagonal elements of A; however on many machines the replacement code will be more efficient.

```
Old: CALL F02SWF(N,A,LDA,D,E,NCOLY,Y,LDY,WANTQ,Q,LDQ,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = J+1, N
           A(I,J) = 0.0e0
  10    CONTINUE
  20 CONTINUE
     CALL sgebrd(N,N,A,LDA,D,E,TAUQ,TAUP,WORK,LWORK,INFO)
     IF (WANTQ) THEN
        CALL F06QFF('L',N,N,A,LDA,Q,LDQ)
        CALL sorgbr('Q',N,N,N,Q,LDQ,TAUQ,WORK,LWORK,INFO)
     END IF
     IF (NCOLY.GT.0) THEN
        CALL sormbr('Q','L','T',N,NCOLY,N,A,LDA,TAUQ,Y,LDY,
    +               WORK,LWORK,INFO)
     END IF
```

where TAUQ, TAUP and WORK are ***real*** arrays of length at least (N), and LWORK is the actual length of WORK.

F02SXF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F02SWF has been replaced by a call to F08KEF as shown above.

```
Old: CALL F02SXF(N,A,LDA,NCOLY,Y,LDY,WORK,IFAIL)
New: IF (NCOLY.EQ.0) THEN
        CALL sorgbr('P',N,N,N,A,LDA,TAUP,WORK,LWORK,INFO)
     ELSE
        CALL sormbr('P','L','T',N,NCOLY,N,A,LDA,TAUP,Y,LDY,WORK,
    +                LWORK,INFO)
     END IF
```

F02SYF
Withdrawn at Mark 18

```
Old: CALL F02SYF(N,D,E,NCOLB,B,LDB,NROWY,Y,LDY,NCOLZ,Z,LDZ,WORK,
    +            IFAIL)
New: CALL sbdsqr('U',N,NCOLZ,NROWY,NCOLB,D,E,Z,LDZ,Y,LDY,B,LDB,WORK,
    +            INFO)
```

where WORK is a ***real*** array of length at least (4*(N−1)) unless NCOLB = NROWY = NCOLZ = 0.

F02SZF
Withdrawn at Mark 15

```
Old: CALL F02SZF(N,D,E,SV,WANTB,B,WANTY,Y,NRY,LY,WANTZ,Z,NRZ,NCZ,
    +            WORK1,WORK2,WORK3,IFAIL)
New: IF (WANTB) THEN
        NCC = 1
     ELSE
        NCC = 0
     END IF
     IF (WANTY) THEN
        NRU = LY
     ELSE
        NRU = 0
     END IF
     IF (WANTZ) THEN
        NCVT = NCZ
     ELSE
        NCVT = 0
     END IF
     CALL sbdsqr('U',N,NCVT,NRU,NCC,D,E(2),Z,NRZ,Y,NRY,B,N,WORK,INFO)
```

WORK must be a one-dimensional ***real*** array of length at least (lwork) given by:

$lwork = 1$ when WANTB, WANTY and WANTZ are all false;

$lwork = max(1, 4x(N-1))$ otherwise.

The parameters WORK1, WORK2 and WORK3 are no longer required.

F02UWF
Withdrawn at Mark 18

The following replacement ignores the triangular structure of A, and therefore references the subdiagonal elements of A; however on many machines the replacement code will be more efficient.

```
Old: CALL F02UWF(N,A,LDA,D,E,NCOLY,Y,LDY,WANTQ,Q,LDQ,WORK,IFAIL)
New: DO 20 J = 1, N
        DO 10 I = J+1, N
           A(I,J) = 0.0e0
```

```
10    CONTINUE
20 CONTINUE
   CALL cgebrd(N,N,A,LDA,D,E,TAUQ,TAUP,WORK,LWORK,INFO)
   IF (WANTQ) THEN
      CALL F06TFF('L',N,N,A,LDA,Q,LDQ)
      CALL cungbr('Q',N,N,N,Q,LDQ,TAUQ,WORK,LWORK,INFO)
   END IF
   IF (NCOLY.GT.0) THEN
      CALL cunmbr('Q','L','C',N,NCOLY,N,A,LDA,TAUQ,Y,LDY,
  +                WORK,LWORK,INFO)
   END IF
```

where TAUQ and TAUP are ***complex*** arrays of length at least (N), and LWORK is the actual length of WORK.

F02UXF
Withdrawn at Mark 18

The following replacement is valid only if the previous call to F02UWF has been replaced by a call to F08KSF (CGEBRD/ZGEBRD) as shown above.

```
Old: CALL F02UXF(N,A,LDA,NCOLY,Y,LDY,RWORK,CWORK,IFAIL)
New: IF (NCOLY.EQ.0) THEN
        CALL cungbr('P',N,N,N,A,LDA,TAUP,CWORK,LWORK,INFO)
     ELSE
        CALL cunmbr('P','L','C',N,NCOLY,N,A,LDA,TAUP,Y,LDY,CWORK,
  +                LWORK,INFO)
     END IF
```

where LWORK is the actual length of CWORK.

F02UYF
Withdrawn at Mark 18

```
Old: CALL F02UYF(N,D,E,NCOLB,B,LDB,NROWY,Y,LDY,NCOLZ,Z,LDZ,WORK,
    +            IFAIL)
New: CALL cbdsqr('U',N,NCOLZ,NROWY,NCOLB,D,E,Z,LDZ,Y,LDY,B,LDB,WORK,
    +            INFO)
```

where WORK is a ***real*** array of length at least (4*(N−1)) unless NCOLB = NROWY = NCOLZ = 0.

F02WAF
Withdrawn at Mark 16

```
Old: CALL F02WAF(M,N,A,NRA,WANTB,B,SV,WORK,LWORK,IFAIL)
New: IF (WANTB) THEN
        NCOLB = 1
     ELSE
        NCOLB = 0
     END IF
     CALL F02WEF(M,N,A,NRA,NCOLB,B,M,.FALSE.,WORK,1,SV,.TRUE.,
    +            WORK,1,RWORK,IFAIL)
```

RWORK must be a one-dimensional ***real*** array of length at least *lwork* given by:

$lwork = \max(3 \times (\mathrm{N} - 1), 1)$ when WANTB is false;

$lwork = \max(5 \times (\mathrm{N} - 1), 2)$ when WANTB is true.

If, in the call to F02WAF, LWORK satisfies these conditions then F02WEF may be called with RWORK as WORK.

F02WBF
Withdrawn at Mark 14

```
Old: CALL F02WBF(M,N,A,NRA,WANTB,B,SV,WORK,LWORK,IFAIL)
New: IF (WANTB) THEN
        NCOLB = 1
     ELSE
        NCOLB = 0
     END IF
     CALL F02WEF(M,N,A,NRA,NCOLB,B,M,.FALSE.,WORK,1,SV,.TRUE.,
    +            WORK,1,RWORK,IFAIL)
```

RWORK must be a one-dimensional ***real*** array of length at least *lwork* given by:

$lwork = \max(3 \times (\mathrm{M}-1), 1)$ when M = N and WANTB is false;

$lwork = \max(5 \times (\mathrm{M}-1), 1)$ when M = N and WANTB is true;

$lwork = \mathrm{M}^2 + 3 \times (\mathrm{M}-1)$ when M < N and WANTB is false;

$lwork = \mathrm{M}^2 + 5 \times (\mathrm{M}-1)$ when M < N and WANTB is true.

In the cases where WANTB is false F02WEF may be called with RWORK as WORK, but when WANTB is true the user should check that, in the call to F02WBF, LWORK satisfies the above conditions before replacing RWORK with WORK.

F02WCF
Withdrawn at Mark 14

```
Old: CALL F02WCF(M,N,MINMN,A,NRA,Q,NRQ,SV,PT,NRPT,WORK,LWORK,
    +            IFAIL)
New: IF (M.GE.N) THEN
        CALL F06QFF('General',M,N,A,NRA,Q,NRQ)
        CALL F02WEF(M,N,Q,NRQ,0,WORK,1,.TRUE.,WORK,1,SV,.TRUE.,
    +               PT,NRPT,RWORK,IFAIL)
     ELSE
        CALL F06QFF('General',M,N,A,NRA,PT,NRPT)
        CALL F02WEF(M,N,PT,NRPT,0,WORK,1,.TRUE.,Q,NRQ,SV,.TRUE.,
    +               WORK,1,RWORK,IFAIL)
     END IF
```

RWORK must be a one-dimensional ***real*** array of length at least *lwork* given by:

$lwork = \mathrm{N}^2 + 5 \times (\mathrm{N}-1)$ when M $\geq$ N

$lwork = \mathrm{M}^2 + 5 \times (\mathrm{M}-1)$ when M < N.

If, in the call to F02WCF, LWORK satisfies these conditions then F02WEF may be called with RWORK as WORK.

F03 – Determinants

F03AGF
Withdrawn at Mark 17

```
Old: CALL F03AGF(N,M,A,IA,RL,IL,M1,D1,ID,IFAIL)
New: CALL spbtrf('Lower',N,M,A,IA,IFAIL)
```

where the array RL and its associated dimension parameter IL, and the parameters M1, D1 and ID are no longer required. In F07HDF (SPBTRF/DPBTRF), the array A holds the matrix packed using a different scheme to that used by F03AGF; see the routine document for details. F07HDF (SPBTRF/DPBTRF) overwrites A with the Cholesky factor L (without reciprocating diagonal elements) rather than returning L in the array RL. F07HDF (SPBTRF/DPBTRF) does not compute the determinant of the input matrix, returned as $\mathrm{D1} \times 2.0^{\mathrm{ID}}$ by F03AGF. If this is required, it may be calculated after the call of F07HDF (SPBTRF/DPBTRF) by code similar to the following. The code computes the determinant by multiplying the diagonal elements of the factor L, taking care to avoid possible overflow or underflow.

```
      D1 = 1.0e0
      ID = 0
      DO 30 I = 1, N
         D1 = D1*A(1,I)**2
10    IF (D1.GE.1.0e0) THEN
         D1 = D1*0.0625e0
         ID = ID + 4
         GO TO 10
      END IF
20    IF (D1.LT.0.0625e0) THEN
         D1 = D1*16.0e0
         ID = ID - 4
         GO TO 20
      END IF
30 CONTINUE
```

F03AHF
Withdrawn at Mark 17

```
Old: CALL F03AHF(N,A,IA,DETR,DETI,ID,RINT,IFAIL)
New: CALL cgetrf(N,N,A,IA,IPIV,IFAIL)
```

where IPIV is an INTEGER array of length N which holds the indices of the pivot elements, and the array RINT is no longer required. It may be important to note that after a call of F07ARF (CGETRF/ZGETRF), A is overwritten by the upper triangular factor U and the off-diagonal elements of the unit lower triangular factor L, whereas the factorization returned by F03AHF gives U the unit diagonal. F07ARF (CGETRF/ZGETRF) does not compute the determinant of the input matrix, returned as ***cmplx***(DETR,DETI)$\times 2.0^{\mathrm{ID}}$ by F03AHF. If this is required, it may be calculated after a call of F07HDF (SPBTRF/DPBTRF) by code similar to the following, where DET is a ***complex*** variable. The code computes the determinant by multiplying the diagonal elements of the factor U, taking care to avoid possible overflow or underflow.

```
      DET = cmplx(1.0e0,0.0e0)
      ID = 0
      DO 30 I = 1, N
         IF (IPIV(I).NE.I) DET = -DET
         DET = DET*A(I,I)
10    IF (MAX(ABS(real(DET)),ABS(imag(DET))).GE.1.0e0) THEN
         DET = DET*0.0625e0
         ID = ID + 4
         GO TO 10
      END IF
20    IF (MAX(ABS(real(DET)),ABS(imag(DET))).LT.0.0625e0) THEN
         DET = DET*16.0e0
         ID = ID - 4
         GO TO 20
      END IF
30 CONTINUE
      DETR = real(DET)
      DETI = imag(DET)
```

F03AMF
Withdrawn at Mark 17

```
Old: CALL F01BNF(N,A,IA,P,IFAIL)
     CALL F03AMF(N,TEN,P,D1,D2)
New: CALL cpotrf('Upper',N,A,IA,IFAIL)
     D1 = 1.0e0
     D2 = 0.0e0
     DO 30 I = 1, N
        D1 = D1*real(A(I,I))**2
  10    IF (D1.GE.1.0e0) THEN
           D1 = D1*0.0625e0
           D2 = D2 + 4
           GO TO 10
        END IF
  20    IF (D1.LT.0.0625e0) THEN
           D1 = D1*16.0e0
           D2 = D2 - 4
           GO TO 20
       END IF
  30 CONTINUE
     IF (TEN) THEN
        I = D2
        D2 = D2*LOG10(2.0e0)
        D1 = D1*2.0e0**(I-D2/LOG10(2.0e0))
     END IF
```

F03AMF computes the determinant of a Hermitian positive-definite matrix after factorization by F01BNF, and has no replacement routine. F01BNF has been superseded by F07FRF (CPOTRF/ZPOTRF). To compute the determinant of such a matrix, in the same form as that returned by F03AMF, code similar to the above may be used. The code computes the determinant by multiplying the (real) diagonal elements of the factor U, taking care to avoid possible overflow or underflow.

Note that before the call of F07FRF (CPOTRF/ZPOTRF), array A contains the upper triangle of the matrix rather than the lower triangle.

F04 – Simultaneous Linear Equations

F04AKF
Withdrawn at Mark 17

```
Old: CALL F04AKF(N,IR,A,IA,P,B,IB)
New: CALL cgetrs('No Transpose',N,IR,A,IA,IPIV,B,IB,INFO)
```

It is assumed that the matrix has been factorized by a call of F07ARF (CGETRF/ZGETRF) rather than F03AHF; see the F03 Chapter Introduction for details. IPIV is an INTEGER array of length N, as returned by F07ARF (CGETRF/ZGETRF), and the array P is no longer required. INFO is an INTEGER diagnostic parameter; see the F07ASF (CGETRS/ZGETRS) routine document for details.

F04ALF
Withdrawn at Mark 17

```
Old: CALL F04ALF(N,M,IR,RL,IRL,M1,B,IB,X,IX)
New: CALL F06QFF('General',N,IR,B,IB,X,IX)
     CALL spbtrs('Lower',N,M,IR,A,IA,X,IX,INFO)
```

It is assumed that the matrix has been factorized by a call of F07HDF (SPBTRF/DPBTRF) rather than F03AGF; see the F03 Chapter Introduction for details. A is the factorized matrix as returned by F07HDF (SPBTRF/DPBTRF). The array RL, its associated dimension parameter IRL, and the parameter M1 are no longer required. INFO is an INTEGER diagnostic parameter; see the F07HEF (SPBTRS/DPBTRS) routine document for details. If the original right-hand side matrix B is no longer required, the call to

F06QFF is not necessary, and references to X and IX in the call of F07HEF (SPBTRS/DPBTRS) may be replaced by references to B and IB, in which case B will be overwritten by the solution.

F04ANF
Withdrawn at Mark 18

```
Old: CALL F04ANF(M,N,QR,IQR,ALPHA,IPIV,B,X,Z)
New: CALL scopy(N,ALPHA,1,QR,IQR+1)
     CALL sormqr('L','T',M,1,N,QR,IQR,Y,B,M,Z,N,INFO)
     CALL strsv('U','N','N',N,QR,IQR,B,1)
     DO 10 I = 1, N
        X(IPIV(I)) = B(I)
  10 CONTINUE
```

where Y must be the same ***real*** array as was used as the 7th argument in the previous call of F01AXF.

This replacement is valid only if the previous call to F01AXF has been replaced by a call to F08BEF (SGEQPF/DGEQPF) as shown above.

F04AQF
Withdrawn at Mark 16

may be replaced by calls to F06EFF (SCOPY/DCOPY), and F07GEF (SPPTRS/DPPTRS) or F07PEF (SSPTRS/DSPTRS), depending on whether the symmetric matrix has previously been factorized by F07GDF (SPPTRF/DPPTRF) or F07PDF (SSPTRF/DSPTRF) (see the description above of how to replace calls to F01BQF).

(a) where the symmetric matrix has been factorized by F07GDF (SPPTRF/DPPTRF)

```
Old: CALL F04AQF(N,M,RL,D,B,X)
New: CALL scopy(N,B,1,X,1)
     CALL spptrs('Lower',N,1,RL,X,N,INFO)
```

(b) where the symmetric matrix has been factorized by F07PDF (SSPTRF/DSPTRF)

```
Old: CALL F04AQF(N,M,RL,D,B,X)
New: CALL scopy(N,B,1,X,1)
     CALL ssptrs('Lower',N,1,RL,IPIV,X,N,INFO)
```

In both (a) and (b), the array RL must be as returned by the relevant factorization routine. The INTEGER parameter INFO is a diagnostic parameter. The INTEGER array IPIV in (b) must be as returned by F07PDF (SSPTRF/DSPTRF). The dimension parameter M, and the array D, are no longer required. If the right-hand-side array B is not needed after solution of the equations, the call to F06EFF (SCOPY/DCOPY), which simply copies array B to X, is not necessary. References to X in the calls of F07GEF (SPPTRS/DPPTRS) and F07PEF (SSPTRS/DSPTRS) may then be replaced by references to B, in which case B will be overwritten by the solution vector.

F04AWF
Withdrawn at Mark 17

```
Old: CALL F04AWF(N,IR,A,IA,P,B,IB,X,IX)
New: CALL F06TFF('General',N,IR,B,IB,X,IX)
     CALL cpotrs('Upper',N,IR,A,IA,X,IX,INFO)
```

It is assumed that the matrix has been factorized by a call of F07FRF (CPOTRF/ZPOTRF) rather than F01BNF; see the F01 Chapter Introduction for details. A is the factorized matrix as returned by F07FRF (CPOTRF/ZPOTRF). The array P is no longer required. INFO is an INTEGER diagnostic parameter; see the F07FSF (CPOTRS/ZPOTRS) routine document for details. If the original right-hand side array B is no longer required, the call to F06TFF is not necessary, and references to X and IX in the call of F07FSF (CPOTRS/ZPOTRS) may be replaced by references to B and IB, in which case B will be overwritten by the solution.

F04AYF
Withdrawn at Mark 18

```
Old: CALL F04AYF(N,IR,A,IA,P,B,IB,IFAIL)
New: CALL sgetrs('No Transpose',N,IR,A,IA,IPIV,B,IB,IFAIL)
```

It is assumed that the matrix has been factorized by a call of F07ADF (SGETRF/DGETRF) rather than F01BTF. IPIV is an INTEGER array of length N, and the array P is no longer required.

F04AZF
Withdrawn at Mark 17

```
Old: CALL F04AZF(N,IR,A,IA,P,B,IB,IFAIL)
New: CALL spotrs('Upper',N,IR,A,IA,B,IB,IFAIL)
```

It is assumed that the matrix has been factorized by a call of F07FDF (SPOTRF/DPOTRF) rather than F01BXF. The array P is no longer required.

F04LDF
Withdrawn at Mark 18

```
Old: CALL F04LDF(N,M1,M2,IR,A,IA,AL,IL,IN,B,IB,IFAIL)
New: CALL sgbtrs('No Transpose',N,M1,M2,IR,A,IA,IN,B,IB,IFAIL)
```

It is assumed that the matrix has been factorized by a call of F07BDF (SGBTRF/DGBTRF) rather than F01LBF. The array AL and its associated dimension parameter IL are no longer required.

F04MAF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

Existing programs should be modified to call F11JCF. The interfaces are significantly different and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine document.

F04MBF
Superseded at Mark 17
Scheduled for withdrawal at Mark 19

If a user-defined preconditioner is required existing programs should be modified to call F11GAF, F11GBF and F11GCF. Otherwise F11JCF or F11JEF may be used. The interfaces for these routines are significantly different from that for F04MBF and therefore precise details of a replacement call cannot be given. Please consult the appropriate routine document.

F04NAF
Withdrawn at Mark 17

```
Old: CALL F04NAF(JOB,N,ML,MU,A,NRA,IN,B,TOL,IFAIL)
New: JOB = ABS(JOB)
     IF (JOB.EQ.1) THEN
        CALL cgbtrs('No Transpose',N,ML,MU,1,A,NRA,IN,B,N,IFAIL)
     ELSE IF (JOB.EQ.2) THEN
        CALL cgbtrs('Conjugate Transpose',N,ML,MU,1,A,NRA,IN,B,N,IFAIL)
     ELSE IF (JOB.EQ.3) THEN
        CALL ctbsv('Upper','No Transpose','Non-unit',N,ML+MU,A,NRA,B,1)
     END IF
```

It is assumed that the matrix has been factorized by a call of F07BRF (CGBTRF/ZGBTRF) rather than F01NAF. The replacement routines do not have the functionality to perturb diagonal elements of the triangular factor U, as specified by a negative value of JOB in F04NAF. The parameter TOL is therefore no longer useful. If this functionality is genuinely required, please contact NAG.

F05 – Orthogonalisation

F05ABF
Withdrawn at Mark 14

```
Old: U = F05ABF(X,N)
New: U = snrm2(N,X,1)
```

F06 – Linear Algebra Support Routines

F06QGF
Withdrawn at Mark 16

```
Old: ANORM = F06QGF(NORM,MATRIX,M,N,A,LDA)
New: C = MATRIX(1:1)
     IF ( (C.EQ.'G') .OR. (C.EQ.'g') ) THEN
        ANORM = F06RAF(NORM,M,N,A,LDA,WORK1)
     ELSE IF ( (C.EQ.'H') .OR. (C.EQ.'h') .OR. (C.EQ.'S') .OR.
    +          (C.EQ.'s')) THEN
        ANORM = F06RCF(NORM,'U',N,A,LDA,WORK2)
     ELSE IF ( (C.EQ.'E') .OR. (C.EQ.'e') .OR. (C.EQ.'Y') .OR.
    +     (C.EQ.'y')) THEN
        ANORM = F06RCF(NORM,'L',N,N,A,LDA,WORK1)
     ELSE IF ( (C.EQ.'U') .OR. (C.EQ.'u') ) THEN
        ANORM = F06RJF(NORM,'U','N',M,N,A,LDA,WORK1)
     ELSE IF ( (C.EQ.'L') .OR. (C.EQ.'l') ) THEN
        ANORM = F06RJF(NORM,'L','N',M,N,A,LDA,WORK1)
     END IF
```

C must be declared as `CHARACTER*1`, WORK1 as a ***real*** array of dimension (1) and WORK2 as a ***real*** array of dimension (N).

F06VGF
Withdrawn at Mark 16

```
Old: ANORM = F06VGF(NORM,MATRIX,M,N,A,LDA)
New: C = MATRIX(1:1)
     IF ( (C.EQ.'G') .OR. (C.EQ.'g') ) THEN
        ANORM = F06UAF(NORM,M,N,A,LDA,WORK1)
     ELSE IF ( (C.EQ.'H') .OR. (C.EQ.'h') .OR. (C.EQ.'S') .OR.
    +          (C.EQ.'s')) THEN
        ANORM = F06UCF(NORM,'U',N,A,LDA,WORK2)
     ELSE IF ( (C.EQ.'E') .OR. (C.EQ.'e') .OR. (C.EQ.'Y') .OR.
    +     (C.EQ.'y')) THEN
        ANORM = F06UCF(NORM,'L',N,A,LDA,WORK1)
     ELSE IF ( (C.EQ.'U') .OR. (C.EQ.'u') ) THEN
        ANORM = F06UJF(NORM,'U','N',M,N,A,LDA,WORK1)
     ELSE IF ( (C.EQ.'L') .OR. (C.EQ.'l') ) THEN
        ANORM = F06UJF(NORM,'L','N',M,N,A,LDA,WORK1)
     END IF
```

C must be declared as `CHARACTER*1`, WORK1 as a ***real*** array of dimension (1) and WORK2 as a ***real*** array of dimension (N).

G01 – Simple Calculations on Statistical Data

G01BAF
Withdrawn at Mark 16

```
Old: P = G01BAF(IDF,T,IFAIL)
New: P = G01EBF('Lower-tail',T,real(IDF),IFAIL)
```

G01BBF
Withdrawn at Mark 16

```
Old: P = G01BBF(I1,I2,A,IFAIL)
New: P = G01EDF('Upper-tail',A,real(I1),real(I2),IFAIL)
```

G01BCF
Withdrawn at Mark 16

```
Old: P = G01BCF(X,N,IFAIL)
New: P = G01ECF('Upper-tail',X,real(N),IFAIL)
```

G01BDF
Withdrawn at Mark 16

```
Old: P = G01BDF(X,A,B,IFAIL)
New: CALL G01EEF(X,A,B,TOL,P,Q,PDF,IFAIL)
```

where TOL is set to the accuracy required by the user and Q and PDF are additional output quantities.

Note. The values of A and B must be $\leq 10^6$.

G01CAF
Withdrawn at Mark 16

```
Old: T = G01CAF(P,N,IFAIL)
New: T = G01FBF('Lower-tail',P,real(N),IFAIL)
```

G01CBF
Withdrawn at Mark 16

```
Old: F = G01CBF(P,M,N,IFAIL)
New: F = G01FDF(P,real(M),real(N),IFAIL)
```

G01CCF
Withdrawn at Mark 16

```
Old: X = G01CCF(P,N,IFAIL)
New: X = G01FCF(P,real(N),IFAIL)
```

G01CDF
Withdrawn at Mark 16

```
Old: X = G01CDF(P,A,B,IFAIL)
New: X = G01FEF(P,A,B,TOL,IFAIL)
```

where TOL is set to the accuracy required by the user.

Note. The values of A and B must be $\leq 10^6$.

G01CEF
Withdrawn at Mark 18

```
Old: X = G01CEF(P,IFAIL)
New: X = G01FAF('Lower-tail',P,IFAIL)
```

G02 – Correlation and Regression Analysis

G02CJF
Withdrawn at Mark 16

```
Old:       CALL G02CJF(X,IX,Y,IY,N,M,IR,THETA,IT,SIGSQ,C,IC,IPIV,
          +            WK1,WK2,IFAIL)
New: C     set the first M elements of ISX to 1
           CALL F06DBF(M,1,ISX,1)
     C     THEN
           TOL = X02AJF()
           CALL G02DAF('Zero','Unweighted',N,X,IX,M,ISX,M,Y,WT,
          +            RSS,IDF,THETA,SE,COV,RES,H,C,IC,SVD,IRANK,
          +            P,TOL,WK,IFAIL)
```

```
      SIGSQ(1) = RSS/IDF
C     there are two or more dependent variables,
C     i.e., IR is greater than or equal to 2 then:
      DO 20 I = 2, IR
         CALL G02DGF('Unweighted',N,WT,RSS,IP,IRANK,COV,C,IC,SVD,
     +               P,Y(1,I),THETA(1,I),SE,RES,WK,IFAIL)
         SIGSQ(I) = RSS/IDF
   20 CONTINUE
```

For unweighted regression, as is used here, WT may be any ***real*** array and will not be referenced, e.g. SIGSQ could be used.

The array C no longer contains $(X^TX)^{-1}$; however, $(X^TX)^{-1}$ scaled by $\hat{\sigma}^2$ is returned in packed form in array COV. The upper triangular part of C will now contain a factorization of X^TX.

The ***real*** arrays SE(M), COV(M*(M + 1)/2), RES(N), H(N), P(M*(M + 2)), the logical variable SVD and the INTEGER variable IRANK are additional outputs. There is also a single ***real*** workspace WK(5*(M − 1) + M ∗ M).

G04 – Analysis of Variance

G04ADF
Withdrawn at Mark 17

```
Old: CALL G04ADF(DATA,VAR,AMR,AMC,AMT,LCODE,IA,N,NN)
New: IFAIL = 0
     CALL G04BCF(1,N,N,DATA,N,IT,GMEAN,AMT,TABLE,6,C,NMAX,
    +            IREP,RPMEAN,AMR,AMC,R,EF,0.0,0,WK,IFAIL)
```

The arrays AMR, AMC and AMT contain the means of the rows, columns and treatments rather than the totals. The values equivalent to those returned in the array VAR of G04ADF are returned in the second column of the two-dimensional array TABLE starting at the second row, e.g., VAR(1) = TABLE(2,2). The two dimensional integer array LCODE (containing the treatment codes) has been replaced by the one-dimensional array IT. These arrays will be the equivalent if IA = N. The following additional declarations are required.

```
 real       GMEAN
 INTEGER    IFAIL
 real       C(NMAX,NMAX), EF(NMAX), TABLE(6,5), R(NMAX*NMAX),
+           RPMEAN(1), WK(NMAX*NMAX+NMAX)
 INTEGER    IREP(NMAX), IT(NMAX*NMAX)
```

where NMAX is an integer such that NMAX ≥ N.

G04AEF
Withdrawn at Mark 17

```
Old: CALL G04AEF(Y,N,K,NOBS,GBAR,GM,SS,IDF,F,FP,IFAIL)
New: CALL G04BBF(N,Y,0,K,IT,GM,BMEAN,GBAR,TABLE,4,C,KMAX,NOBS,
    +            R,EF,0.0e0,0,WK,IFAIL)
```

The values equivalent to those returned by G04AEF in the arrays IDF and SS are returned in the first and second columns of TABLE starting at row 2 and the values equivalent to those returned in the scalars F and FP are returned in TABLE(2,4) and TABLE(2,5) respectively. NOBS is output from G04BBF rather than input. The groups are indicated by the array IT. The following code illustrates how IT can be computed from NOBS.

```
      IJ = 0
      DO 40 I = 1, K
         DO 20 J = 1, NOBS(I)
            IJ = IJ + 1
            IT(IJ) = I
20       CONTINUE
40    CONTINUE
```

The following additional declarations are required.

```
 real        BMEAN(1),C(KMAX,KMAX),EF(KMAX),R(NMAX),TABLE(4,5),
+            WK(KMAX*KMAX+KMAX)
 INTEGER     IT(NMAX)
```

NMAX and KMAX are integers such that NMAX $\geq$ N and KMAX $\geq$ K.

G04AFF
Withdrawn at Mark 17

```
Old: CALL G04AFF(Y,IY1,IY2,M,NR,NC,ROW,COL,CELL,ICELL,GM,SS,IDF,F,FP,
    +            IFAIL)
New: CALL G04CAF(M*NR*NC,Y1,2,LFAC,1,2,0,6,TABLE,ITOTAL,TMEAN,MAXT,E,
    +            IMEAN,SEMEAN,BMEAN,R,IWK,IFAIL)
```

Where Y1 is a one-dimensional array containing the observations in the same order as Y, if IY1 = M and IY2 = NR then these are equivalent. LFAC is an integer array such that LFAC(1) = NC and LFAC(2) = NR. The following indicates how the results equivalent to those produced by G04AFF can be extracted from the results produced by G04CAF.

G04AFF	G04CAF
ROW(i)	TMEAN(IMEAN(1)+i), i = 1,2,...,NR
COL(j)	TMEAN(j), j = 1,2,...,NC
CELL(i,j)	TMEAN(IMEAN(2)+(j-1)*NR+i), i = 1,2,...,NR; j = 1,2,...,NC
GM	BMEAN(1)
SS(1)	TABLE(3,2)
SS(2)	TABLE(2,2)
SS(i)	TABLE(4,2)
IDF(1)	TABLE(3,1)
IDF(2)	TABLE(2,1)
IDF(i)	TABLE(4,1)
F(1)	TABLE(3,4)
F(2)	TABLE(2,4)
F(3)	TABLE(4,4)
FP(1)	TABLE(3,5)
FP(2)	TABLE(2,5)
FP(3)	TABLE(4,5)

Note how rows and columns have swapped.

The following additional declarations are required.

```
 real        TABLE(6,5), R(NMAX), TMEAN(MAXT), E(MAXT), BMEAN(1),
+            SEMEAN(5)
 INTEGER     IMEAN(5), IWK(NMAX+6), LFAC(2)
```

NMAX and MAXT are integers such that NMAX $\geq$ M $\times$ NR $\times$ NC and MAXT $\geq$ NR + NC + NR $\times$ NC.

G05 – Random Number Generators

G05DGF
Withdrawn at Mark 16

```
Old: X = G05DGF(G,H,IFAIL)
New: CALL G05FFF(G,H,1,X(1),IFAIL)
```

where X must now be declared as an array of length at least 1.

G05DLF
Withdrawn at Mark 16

```
Old: X = G05DLF(G,H,IFAIL)
New: CALL G05FEF(G,H,1,X(1),IFAIL)
```

where X must now be declared as an array of length at least 1.

G05DMF
Withdrawn at Mark 16

```
Old: X = G05DMF(G,H,IFAIL)
New: CALL G05FEF(G,H,1,X(1),IFAIL)
     IF (X(1).LT.1.0e0) X(1) = X(1)/(1.0e0-X(1))
```

where X must now be declared as an array of length at least 1. If the value of X(1) returned by G05FEF is 1.0, appropriate action should be taken. Alternatively the ratio of gamma variates can be used i.e.,

```
CALL G05FFF(G,1.0e0,1,X(1),IFAIL1)
CALL G05FFF(H,1.0e0,1,Y(1),IFAIL2)
IF (Y(1).NE.0.0e0) X(1) = X(1)/Y(1)
```

where Y must be declared as an array of length at least 1.

G08 – Nonparametric Statistics

G08ABF
Withdrawn at Mark 16

```
Old: CALL G08ABF(X,Y,N,W1,W2,W,N1,P,IFAIL)
New: DO 20 I = 1, N
        Z(I) = X(I) - Y(I)
  20 CONTINUE
     XME = 0.0e0
     CALL G08AGF(N,Z,XME,'Lower-tail','No-zeros',W,WNOR,P,
    +            N1,W1,IFAIL)
```

W1 is a ***real*** work array of dimension (3*N). The ***real*** array W2 is no longer required. WNOR returns the normalized Wilcoxon test statistic. The ***real*** array Z, of dimension (N), contains the difference between the paired sample observations, and by setting the ***real*** variable XME to zero the routine may be used to test whether the medians of the two matched or paired samples are equal.

G08ADF
Withdrawn at Mark 16

```
Old: CALL G08ADF(X,N,N1,W,U,P,IFAIL)
New: N2 = N - N1
     CALL G08AHF(N1,X,N2,X(N1+1),'Lower-tail',U,UNOR,P,
    +            TIES,RANKS,W,IFAIL)
```

The observations from the two independent samples must be stored in two separate ***real*** arrays, of dimensions N1 and N2, where N2 = N − N1, rather than consecutively in one array as in the superseded routine G08ADF.

UNOR returns the normalized Mann–Whitney U statistic. The LOGICAL parameter TIES indicates whether ties were present in the pooled sample or not and RANKS, a ***real*** array of dimension (N1 + N2), returns the ranks of the pooled sample.

Both G08ADF and its replacement routine G08AHF return approximate tail probabilities for the test statistic. To compute exact tail probabilities G08AJF may be used if there are no ties in the pooled sample and G08AKF may be used if there are ties in the pooled sample.

G08CAF
Withdrawn at Mark 16

```
Old: CALL G08CAF(N,X,NULL,NP,P,NEST,NTYPE,D,PROB,S,IND,IFAIL)
New: CALL G08CBF(N,X,DIST,PAR,NEST,NTYPE,D,Z,PROB,S,IFAIL)
```

The following table indicates how existing choices for the null distribution, indicated through the INTEGER variable NULL in G08CAF, may be made in G08CBF using the character variable DIST.

null distribution	G08CAF – NULL	G08CBF – DIST
uniform	1	'U'
Normal	2	'N'
Poisson	3	'P'
exponential	4	'E'

PAR is a ***real*** array of dimension (1) for both the one and two parameter distributions, but only the first element of PAR is actually referenced (used) if the chosen null distribution has only one parameter. The input parameter NP is no longer required.

On exit S contains the sample observations sorted into ascending order. It no longer contains the sample cumulative distribution function but this may be computed from S.

G13 – Time Series Analysis

G13DAF
Withdrawn at Mark 17

```
Old:        CALL G13DAF(X,NXM,NX,NSM,NS,NL,ICR,C0,C,IFAIL)
New: C      First transpose the data matrix X
     C      note NSM is used as the first dimension of the array W
            DO 20 I = 1, NS
               CALL F06EFF(NX,X,(1,I),1,W(I,1),NSM)
         20 CONTINUE
     C      then if ICR = 0 in the call to G13DAF
            CALL G13DMF('V-Covariances',NS,NX,W,NSM,NL,WMEAN,C0,C,IFAIL)
     C      else if ICR = 1 in the call to G13DAF
            CALL G13DMF('R-Correlations',NS,NX,W,NSM,NL,WMEAN,C0,C,IFAIL)
```

Note that in G13DAF the NS series are stored in the columns of X whereas in G13DMF these series are stored in rows; hence it is necessary to transpose the data array.

The ***real*** array WMEAN must be of length NS, and on output stores the means of each of the NS series.

The diagonal elements of C0 store the variances of the series if covariances are requested, but the standard deviations if correlations are requested.

H – Operations Research

H02BAF
Withdrawn at Mark 15

```
Old:    CALL H02BAF(A,MM,N1,M,N,200,L,X,NUMIT,OPT,IFAIL)
New: C  M, N and MM must be set before these declaration statements
        INTEGER    MAXDPT, LIWORK, LRWORK, ITMAX, MSGLVL, MAXNOD, INTFST
        PARAMETER  (LIWORK = (25+N+M)*MAXDPT + 5*N + M + 4)
        PARAMETER  (LRWORK = MAXDPT*(N+2) + 2*N*N + 13*N + 12*M)
        INTEGER    INTVAR(N), IWORK(LIWORK)
        real       BIGBND, TOLFES, TOLIV, ROPT
        real       RA(MM,N), RX(N), CVEC(N), BL(N+M), BU(N+M),
       +           RWORK(LRWORK)
        IFAIL = 0
        DO 10 J = 1, N
           INTVAR(J) = 1
```

```
            CVEC(J) = A(1,J)
            RX(J) = 1.0e0
            DO 20 I = 1, M
               RA(I,J) = A(I+1,J)
   20       CONTINUE
   10    CONTINUE

         BIGBND = 1.0e20
         DO 30 I = 1, N
            BL(I) = 0.0e0
            BU(I) = BIGBND
   30    CONTINUE
         DO 40 I = N+1, N+M
            BU(I) = A(I-N+1,N+1)
            BL(I) = -BIGBND
   40    CONTINUE
         ITMAX = 0
         MSGLVL = 0
         MAXNOD = 0
         INTFST = 0
         TOLIV = 0.0e0
         TOLFES = 0.0e0
         MAXDPT = 3*N/2

         CALL H02BBF(ITMAX,MSGLVL,N,M,RA,MM,BL,BU,INTVAR,CVEC,MAXNOD,
        +            INTFST,MAXDPT,TOLIV,TOLFES,BIGBND,RX,ROPT,IWORK,
        +            LIWORK,RWORK,LRWORK,IFAIL)
         L = 1
         IF (IFAIL.EQ.0) L = 0
         IF (IFAIL.EQ.4) L = 2

         IF (L.EQ.0) THEN
            DO 50 I = 1, N
               X(I) = RX(I)
   50       CONTINUE
            OPT = ROPT
         ENDIF
```

The code indicates the minimum changes necessary, but H02BBF has additional flexibility and users may wish to take advantage of new features. It is strongly recommended that users consult the routine document.

M01 – Sorting

M01AAF
Withdrawn at Mark 13

```
Old: CALL M01AAF(A,M,N,IP,IST,IFAIL)
New: CALL M01DAF(A(M),1,N-M+1,'A',IP(M),IFAIL)
```

The array IST is no longer needed.

M01ABF
Withdrawn at Mark 13

```
Old: CALL M01ABF(A,M,N,IP,IST,IFAIL)
New: CALL M01DAF(A(M),1,N-M+1,'D',IP(M),IFAIL)
```

The array IST is no longer needed.

M01ACF
Withdrawn at Mark 13

```
Old: CALL M01ACF(IA,M,N,IP,IST,IFAIL)
New: CALL M01DBF(IA(M),1,N-M+1,'A',IP(M),IFAIL)
```

The array IST is no longer needed.

M01ADF
Withdrawn at Mark 13

```
Old: CALL M01ADF(IA,M,N,IP,IST,IFAIL)
New: CALL M01DBF(IA(M),1,N-M+1,'D',IP(M),IFAIL)
```

The array IST is no longer needed.

M01AEF
Withdrawn at Mark 13

```
Old: CALL M01AEF(A,NR,NC,IC,T,TT,IFAIL)
New: CALL M01DEF(A,NR,1,NR,IC,IC,'A',IRANK,IFAIL)
     DO 10 I = 1, NC
        CALL M01EAF(A(1,I),1,NR,IRANK,IFAIL)
  10 CONTINUE
```

The ***real*** arrays T and TT are no longer needed, but a new integer array IRANK of length NR is required.

M01AFF
Withdrawn at Mark 13

```
Old: CALL M01AFF(A,NR,NC,IC,T,TT,IFAIL)
New: CALL M01DEF(A,NR,1,NR,IC,IC,'D',IRANK,IFAIL)
     DO 10 I = 1, NC
        CALL M01EAF(A(1,I),1,NR,IRANK,IFAIL)
  10 CONTINUE
```

The ***real*** arrays T and TT are no longer needed, but a new integer array IRANK of length NR is required.

M01AGF
Withdrawn at Mark 13

```
Old: CALL M01AGF(IA,NR,NC,IC,K,L,IFAIL)
New: CALL M01DFF(IA,NR,1,NR,IC,IC,'A',IRANK,IFAIL)
     DO 10 I = 1, NC
        CALL M01EBF(IA(1,I),1,NR,IRANK,IFAIL)
  10 CONTINUE
```

The integer arrays K and L are no longer needed, but a new integer array IRANK of length NR is required.

M01AHF
Withdrawn at Mark 13

```
Old: CALL M01AHF(IA,NR,NC,IC,K,L,IFAIL)
New: CALL M01DFF(IA,NR,1,NR,IC,IC,'D',IRANK,IFAIL)
     DO 10 I = 1, NC
        CALL M01EBF(IA(1,I),1,NR,IRANK,IFAIL)
  10 CONTINUE
```

The integer arrays K and L are no longer needed, but a new integer array IRANK of length NR is required.

M01AJF
Withdrawn at Mark 16

```
Old: CALL M01AJF(A,W,IND,INDW,N,NW,IFAIL)
New: CALL M01DAF(A,1,N,'A',IND,IFAIL)
     CALL M01ZAF(IND,1,N,IFAIL)
     CALL M01CAF(A,1,N,'A',IFAIL)
```

The arrays W and INDW are no longer needed.

M01AKF
Withdrawn at Mark 16

```
Old: CALL M01AKF(A,W,IND,INDW,N,NW,IFAIL)
New: CALL M01DAF(A,1,N,'D',IND,IFAIL)
     CALL M01ZAF(IND,1,N,IFAIL)
     CALL M01CAF(A,1,N,'D',IFAIL)
```

The arrays W and INDW are no longer needed.

M01ALF
Withdrawn at Mark 13

```
Old: CALL M01ALF(IA,IW,IND,INDW,N,NW,IFAIL)
New: CALL M01DBF(IA,1,N,'A',IND,IFAIL)
     CALL M01ZAF(IND,1,N,IFAIL)
     CALL M01CBF(IA,1,N,'A',IFAIL)
```

The arrays IW and INDW are no longer needed.

M01AMF
Withdrawn at Mark 13

```
Old: CALL M01AMF(IA,IW,IND,INDW,N,NW,IFAIL)
New: CALL M01DBF(IA,1,N,'D',IND,IFAIL)
     CALL M01ZAF(IND,1,N,IFAIL)
     CALL M01CBF(IA,1,N,'D',IFAIL)
```

The arrays IW and INDW are no longer needed.

M01ANF
Withdrawn at Mark 13

```
Old: CALL M01ANF(A,I,J,IFAIL)
New: CALL M01CAF(A,I,J,'A',IFAIL)
```

M01APF
Withdrawn at Mark 16

```
Old: CALL M01APF(A,I,J,IFAIL)
New: CALL M01CAF(A,I,J,'D',IFAIL)
```

M01AQF
Withdrawn at Mark 13

```
Old: CALL M01AQF(IA,I,J,IFAIL)
New: CALL M01CBF(IA,I,J,'A',IFAIL)
```

M01ARF
Withdrawn at Mark 13

```
Old: CALL M01ARF(IA,I,J,IFAIL)
New: CALL M01CBF(IA,I,J,'D',IFAIL)
```

The character-sorting routines M01BAF, M01BBF, M01BCF and M01BDF have no exact replacements, because they require the data to be stored in an integer array, whereas the new character-sorting routines require the data to be stored in a character array. The following advice assumes that calling programs are modified so that the data is stored in a character array CH instead of in an integer array IA; ***nchar*** denotes the machine-dependent number of characters stored in an integer variable. The new routines sort according to the ASCII collating sequence, which may differ from the machine-dependent collating sequence used by the old routines.

M01BAF
Withdrawn at Mark 13

```
Old: CALL M01BAF(IA,I,J,IFAIL)
New: CALL M01CCF(CH,I,J,1,nchar,'D',IFAIL)
```

assuming that each element of the character array CH corresponds to one element of the integer array IA.

M01BBF
Withdrawn at Mark 13

```
Old: CALL M01BBF(IA,I,J,IFAIL)
New: CALL M01CCF(CH,I,J,1,nchar,'A',IFAIL)
```

assuming that each element of the character array CH corresponds to one element of the integer array IA.

M01BCF
Withdrawn at Mark 13

```
Old: CALL M01BCF(IA,NR,NC,L1,L2,LC,IUC,IT,ITT,IFAIL)
New: CALL M01CCF(CH,LC,IUC,(L1-1)*nchar-1,L2*nchar,'D',IFAIL)
```

provided that each element of the character array CH corresponds to a whole column of the integer array IA. The arrays IT and ITT are no longer needed. The call of M01CCF will fail if NR****nchar*** exceeds 255.

M01BDF
Withdrawn at Mark 13

```
Old: CALL M01BDF(IA,NR,NC,L1,L2,LC,IUC,IT,ITT,IFAIL)
New: CALL M01CCF(CH,LC,IUC,(L1-1)*nchar-1,L2*nchar,'A',IFAIL)
```

provided that each element of the character array CH corresponds to a whole column of the integer array IA. The arrays IT and ITT are no longer needed. The call of M01CCF will fail if NR****nchar*** exceeds 255.

X02 – Machine Constants

X02AAF
Withdrawn at Mark 16

```
Old: X02AAF(X)
New: X02AJF()
```

X02ABF
Withdrawn at Mark 16

```
Old: X02ABF(X)
New: X02AKF()
```

X02ACF
Withdrawn at Mark 16

```
Old: X02ACF(X)
New: X02ALF()
```

X02ADF
Withdrawn at Mark 14

```
Old: X02ADF(X)
New: X02AKF()/X02AJF()
```

X02AEF∗
Withdrawn at Mark 14

```
Old: X02AEF(X)
New: LOG(X02AMF())
```

X02AFF∗
Withdrawn at Mark 14

```
Old: X02AFF(X)
New: -LOG(X02AMF())
```

X02AGF∗
Withdrawn at Mark 16

```
Old: X02AGF(X)
New: X02AMF()
```

X02BAF
Withdrawn at Mark 14

```
Old: X02BAF(X)
New: X02BHF()
```

X02BCF∗
Withdrawn at Mark 14

```
Old: X02BCF(X)
New: -LOG(X02AMF())/LOG(2.0)
```

X02BDF∗
Withdrawn at Mark 14

```
Old: X02BDF(X)
New: LOG(X02AMF())/LOG(2.0)
```

X02CAF
Withdrawn at Mark 17

This routine is no longer required.

Note. In the case of the routines marked with an asterisk (*), the replacement expressions may not return the same value, but the value will be sufficiently close, and safe, for the purposes for which it is used in the Library.
