NAG Library

Advice on Replacement Calls for Withdrawn/Superseded Routines

The following list gives the names of replacement routines for those routines that have been withdrawn or superseded. For routines that have been withdrawn or superseded since Mark 13 replacement calls are also given.

The list indicates the minimum change necessary, but many of the replacement routines have additional flexibility and you may wish to take advantage of new features. It is strongly recommended that you consult the routine documents.

C02 – Zeros of Polynomials

C02AEF

Withdrawn at Mark 16.
Replaced by C02AGF.

```
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 ***double precision*** array Z of dimension $(2,\mathrm{N})$ rather than in the arrays REZ and IMZ, and W is a ***double precision*** work array of dimension $(2 \times (\mathrm{N}+1))$.

D02 – Ordinary Differential Equations

D02BAF

Withdrawn at Mark 18.
Replaced by D02PCF and associated D02P routines.

```
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.,
    +            0.0D0,W,20*N,IFAIL)
     CALL D02PCF(FCN,XEND,X,Y,YP,YMAX,W,IFAIL)
```

THRES, YP, YMAX are ***double precision*** arrays of length N and the length of array W needs extending to length 20 × N.

D02BBF

Withdrawn at Mark 18.
Replaced by D02PCF and associated D02P routines.

```
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.,
    +            0.0D0,W,20*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, YMAX are ***double precision*** arrays of length N and the length of array W needs extending to length 20 × N.

D02BDF

Withdrawn at Mark 18.
Replaced by D02PCF and associated D02P routines.

```
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.,
    +            0.0D0,W,32*N,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 ***double precision*** arrays of length N. W is now a ***double precision*** one-dimensional array of length 32 × N.

D02CAF

Withdrawn at Mark 18.
Replaced by D02CJF.

```
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 ***double precision*** 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.
Replaced by D02CJF.

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

D02CGF

Withdrawn at Mark 18.
Replaced by D02CJF.

```
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)
     .
     .
     .
     double precision FUNCTION G(X,Y)
     double precision 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 routine document for D02CJF for more details.

D02CHF
Withdrawn at Mark 18.
Replaced by D02CJF.

```
     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. The relationship between the parameters IRELAB and RELABS is described under the replacement call for D02CBF.

D02EAF
Withdrawn at Mark 18.
Replaced by D02EJF.

```
     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 ***double precision*** function also provided. All must be declared as EXTERNAL.

D02EBF
Withdrawn at Mark 18.
Replaced by D02EJF.

```
     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 ***double precision*** function provided in the NAG Fortran Library and must be declared as EXTERNAL. The integer parameter IRELAB (which can take values 0, 1 or 2) is catered 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.
Replaced by D02EJF.

```
     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)
          .
          .
          .
          double precision FUNCTION G(X,Y)
          double precision 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 that 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 routine document for D02EJF for more details.

D02EHF
Withdrawn at Mark 18.
Replaced by D02EJF.

```
     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,IW,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. 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.
Replaced by D02PDF and associated D02P routines.

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.

D02QDF

Withdrawn at Mark 17.
Replaced by D02NBF or D02NCF.

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.
Replaced by D02NBF or D02NCF.

D02XAF

Withdrawn at Mark 18.
Replaced by D02PXF and associated D02P routines.

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

D02XBF

Withdrawn at Mark 18.
Replaced by D02PXF and associated D02P routines.

Not needed except with D02PAF.

D02YAF

Withdrawn at Mark 18.
Replaced by D02PDF and associated D02P routines.

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

D03 – Partial Differential Equations

D03PAF

Withdrawn at Mark 17.
Replaced by D03PCF/D03PCA.

Existing programs should be modified to call D03PCF/D03PCA. The replacement routine is designed to solve a broader class of problems. Therefore it is not possible to give precise details of a replacement call. Please consult the appropriate routine documents.

D03PBF

Withdrawn at Mark 17.
Replaced by D03PCF/D03PCA.

Existing programs should be modified to call D03PCF/D03PCA. The replacement routine is designed to solve a broader class of problems. Therefore it is not possible to give precise details of a replacement call. Please consult the appropriate routine documents.

D03PGF
Withdrawn at Mark 17.
Replaced by D03PCF/D03PCA.

Existing programs should be modified to call D03PCF/D03PCA. The replacement routine is designed to solve a broader class of problems. Therefore it is not possible to give precise details of a replacement call. Please consult the appropriate routine documents.

E01 – Interpolation

E01ACF
Withdrawn at Mark 15.
Replaced by E01DAF and E02DEF.

```
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 ***double precision*** array of dimension $(\mathrm{N1}+4)$, MU is a ***double precision*** array of dimension $(\mathrm{M1}+4)$, C is a ***double precision*** array of dimension $(\mathrm{N1}\times\mathrm{M1})$, WRK is a ***double precision*** array of dimension $((\mathrm{N1}+6)\times(\mathrm{M1}+6))$, A1, B1 and FF are ***double precision*** arrays of dimension (M), and 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.

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
Withdrawn at Mark 20.
Replaced by E01SGF.

```
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 has been superseded by E01SGF which gives improved accuracy, facilities for obtaining gradient values and a consistent interface with 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
Withdrawn at Mark 20.
Replaced by E01SHF.

```
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 calls 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.
Replaced by E02DEF.

```
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 ***double precision*** array of dimension $(PY - 4)$, and IWRK is an INTEGER array of dimension $(PY - 4)$.

E04 – Minimizing or Maximizing a Function

E04CCF/E04CCA

Scheduled for withdrawal at Mark 24.
Replaced by E04CBF.

```
Old: CALL E04CCF(N,X,F,TOL,IW,W1,W2,W3,W4,W5,W6,FUNCT,MONIT,MAXCAL,
     *           IFAIL)
                      or
     CALL E04CCA(N,X,F,TOL,IW,W1,W2,W3,W4,W5,W6,FUNCT2,MONIT2,MAXCAL,
     *           IUSER,RUSER,IFAIL)
New: CALL E04CBF(N,X,F,TOLF,TOLX,FUNCT2,MONIT3,MAXCAL,IUSER,RUSER,
     *           IFAIL)
```

The new routine can be derived from the old as follows:

```
      SUBROUTINE MONIT3(FMIN,FMAX,SIM,N,NCALL,SERROR,VRATIO,IUSER,RUSER)
      INTEGER    N, NCALL, IUSER(*)
      double precision FMIN, FMAX, SIM(N+1,N), SERROR, VRATIO, RUSER(*)
C
      CALL MONIT2(FMIN,FMAX,SIM,N,N+1,NCALL,IUSER,RUSER)
C      Add code here to monitor the values of SERROR and VRATIO, if neces-
sary
      RETURN
      END
```

E04FDF

Withdrawn at Mark 19.
Replaced by E04FYF.

```
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 (sub)program. 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(*)
      double precision XC(N), FVECC(M), USER(*)
C
      CALL LSFUN1(M,N,XC,FVECC)
C
      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 initializing.

E04GCF
Withdrawn at Mark 19.
Replaced by E04GYF.

```
     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 (sub)program. 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(*)
           double precision XC(N), FVECC(M), FJACC(LJC,N), USER(*)
     C
           CALL LSFUN2(M,N,XC,FVECC,FJACC,LJC)
     C
           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 initializing. 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
Withdrawn at Mark 19.
Replaced by E04GZF.

```
     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 (sub)program. 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(*)
           double precision XC(N), FVECC(M), FJACC(LJC,N), USER(*)
     C
           CALL LSFUN2(M,N,XC,FVECC,FJACC,LJC)
     C
           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 initializing. 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.
There is no replacement for this routine.

E04HFF
Withdrawn at Mark 19.
Replaced by E04HYF.

```
     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 (sub)program. 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(*)
      double precision XC(N), FVECC(M), FJACC(LJC,N), USER(*)
C
      CALL LSFUN2(M,N,XC,FVECC,FJACC,LJC)
C
      RETURN
      END
C
      SUBROUTINE  LSHES(M,N,FVECC,XC,B,LB,IUSER,USER)
      INTEGER     M, N, LB, IUSER(*)
      double precision FVECC(M), XC(N), B(LB), USER(*)
C
      CALL LSHES2(M,N,FVECC,XC,B,LB)
C
      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 initializing. If, however, the array IW was used to pass information through E04HFF into LSFUN2 or LSHES2, or to get information from LSFUN2 or LSHES2, then the array IUSER should be declared appropriately and used for this purpose.

E04JAF
Withdrawn at Mark 19.
Replaced by E04JYF.

```
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 (sub)program. 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(*)
      double precision XC(N), FC, USER(*)
C
      CALL FUNCT1(N,XC,FC)
C
      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 initializing.

E04JBF
Withdrawn at Mark 16.
Replaced by E04UCF/E04UCA.

No comparative calls are given between E04JBF and E04UCF/E04UCA since both routines have considerable flexibility and can be called with many different options. E04UCF/E04UCA allows some values to be passed to it, not through the parameter list, but as ‘optional parameters’, supplied through calls to E04UDF/E04UDA or E04UEF/E04UEA.

E04UCF/E04UCA 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/E04UCA 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/E04UCA 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(*)
              double precision X(N), OBJF, OBJGRD(N), USER(*)
              INTEGER     IFLAG,IW(1)
              double precision 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/E04UCA supplies the arrays IUSER and USER specifically for this purpose.)

The functions of the parameters BL and BU are similar, but E04UCF/E04UCA 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 $1.0\mathrm{D}+6$ 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/E04UCA. It should also be noted that E04UCF/E04UCA does not allow a user-supplied parameter 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/E04UCA, 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 OBJFUN rather than the accuracy in X)
STEPMX	Step Limit
DELTA	Difference Interval

E04KAF
Withdrawn at Mark 19.
Replaced by E04KYF.

```
     Old: CALL E04KAF(N,IBOUND,BL,BU,X,F,G,IW,LIW,W,LW,IFAIL)
     New: CALL E04KYF(N,IBOUND,FUNCT,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 (sub)program. 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(*)
            double precision XC(N), FC, GC(N), USER(*)
      C
            CALL FUNCT2(N,XC,FC,GC)
      C
            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 initializing.

E04KBF

Withdrawn at Mark 16.
Replaced by E04UCF/E04UCA.

No comparative calls are given between E04KBF and E04UCF/E04UCA 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/E04UCA 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(*)
      double precision X(N), OBJF, OBJGRD(N), USER(*)
      INTEGER     IW(1)
      double precision W(1)
C
      CALL FUNCT(MODE,N,X,OBJF,OBJGRD,IW,1,W,1)
      RETURN
      END
```

E04KCF

Withdrawn at Mark 19.
Replaced by E04KZF.

```
Old: CALL E04KCF(N,IBOUND,BL,BU,X,F,G,IW,LIW,W,LW,IFAIL)
New: CALL E04KZF(N,IBOUND,FUNCT,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 E04KCF. FUNCT must be declared as EXTERNAL in the calling (sub)program. 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(*)
      double precision XC(N), FC, GC(N), USER(*)
C
      CALL FUNCT2(N,XC,FC,GC)
C
      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 initializing.

E04LAF

Withdrawn at Mark 19.
Replaced by E04LYF.

```
Old: CALL E04LAF(N,IBOUND,BL,BU,X,F,G,IW,LIW,W,LW,IFAIL)
New: CALL E04LYF(N,IBOUND,FUNCT,HESS,BL,BU,X,F,G,IW,LIW,W,LW,IUSER,USER,
    +            IFAIL)
```

FUNCT and HESS appear in the parameter list instead of the fixed-name subroutines FUNCT2 and HESS2 of E04LAF. FUNCT and HESS must both be declared as EXTERNAL in the calling (sub)program. In addition they have an extra two parameters, IUSER and USER, over and above those of FUNCT2 and HESS2. They may be derived from FUNCT2 and HESS2 as follows:

```
      SUBROUTINE  FUNCT(N,XC,FC,GC,IUSER,USER)
      INTEGER     N, IUSER(*)
      double precision XC(N), FC, GC(N), USER(*)
C
      CALL FUNCT2(N,XC,FC,GC)
C
      RETURN
      END
      SUBROUTINE  HESS(N,XC,HESLC,LH,HESDC,IUSER,USER)
      INTEGER     N, LH, IUSER(*)
      double precision XC(N), HESLC(LH), HESDC(N), USER(*)
C
      CALL HESS2(N,XC,HESLC,LH,HESDC)
C
      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 initializing.

E04MBF
Withdrawn at Mark 18.
Replaced by E04MFF/E04MFA.

```
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/E04MFA contains two additional parameters as follows:

ITER– INTEGER.

AX(∗) – ***double precision*** array of dimension at least $\max(1,\mathrm{NCLIN})$.

The minimum value of the parameter LIWORK must be increased from $2\times\mathrm{N}$ to $2\times\mathrm{N}+3$. 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.
Replaced by E04NFF/E04NFA.

```
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
     double precision HESS(NROWH,NCOLH), X(N), HX(N)
New: SUBROUTINE  QPHESS(N,JTHCOL,HESS,NROWH,X,HX)
     INTEGER     N, JTHCOL, NROWH
     double precision 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/E04NFA contains one additional parameter as follows:

AX(∗) – ***double precision*** array of dimension at least $\max(1,\mathrm{NCLIN})$.

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

E04UNF

Withdrawn at Mark 22.
Replaced by E04USF/E04USA.

```
Old: CALL E04UNF(M,N,NCLIN,NCNLN,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,
    +            RUSER,IFAIL)
New: CALL E04USF(M,N,NCLIN,NCNLN,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,
    +            RUSER,IFAIL)
```

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

```
Old: SUBROUTINE  OBJFUN(MODE,M,N,LDFJ,X,F,FJAC,NSTATE,IUSER,RUSER)
     INTEGER     MODE,M,N,LDFJ,NSTATE,IUSER(*)
     double precision X(N),F(*),FJAC(LDFJ,*),RUSER(*)
New: SUBROUTINE  OBJFUN(MODE,M,N,LDFJ,NEEDFI,X,F,FJAC,NSTATE,
    +                   IUSER,RUSER)
     INTEGER     MODE,M,N,LDFJ,NEEDFI,NSTATE,IUSER(*)
     double precision X(N),F(*),FJAC(LDFJ,*),RUSER(*)
```

See the routine documents for further information.

E04UPF

Withdrawn at Mark 19.
Replaced by E04USF/E04USA.

```
Old: CALL E04UPF(M,N,NCLIN,NCNLN,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 E04USF(M,N,NCLIN,NCNLN,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)
```

E04USF/E04USA contains one additional parameter as follows:

Y(M) – ***double precision*** array.

Note that a call to E04UPF is the same as a call to E04USF/E04USA with Y$(i) = 0.0$, for $i = 1, 2, \ldots,$ M.

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

```
Old: SUBROUTINE  OBJFUN(MODE,M,N,LDFJ,X,F,FJAC,NSTATE,IUSER,USER)
     INTEGER     MODE,M,N,LDFJ,NSTATE,IUSER(*)
     double precision X(N),F(*),FJAC(LDFJ,*),USER(*)
 New: SUBROUTINE  OBJFUN(MODE,M,N,LDFJ,NEEDFI,X,F,FJAC,NSTATE,
    +                   IUSER,USER)
     INTEGER     MODE,M,N,NEEFI,NSTATE,IUSER(*)
     double precision X(N),F(*),FJAC(LDFJ,*),USER(*)
```

See the routine documents for further information.

E04VCF

Withdrawn at Mark 17.
Replaced by E04UCF/E04UCA.

```
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
     double precision X(N), OBJF, OBJGRD(N)
New: SUBROUTINE  OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
     INTEGER     MODE, N, NSTATE, IUSER(*)
     double precision 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
     double precision 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(*)
     double precision X(N), C(NCNLN), CJAC(NROWJ,N), USER(*)
```

If NCNLN $= 0$, then the name of the dummy routine E04VDM/E54VDM may need to be changed to E04UDM 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/E04UCA contains two additional parameters as follows:

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

USER($*$) – ***double precision*** array of dimension at least 1.

The minimum value of the parameter LIWORK must be increased from $3 \times \mathrm{N} + \mathrm{NCLIN} + \mathrm{NCNLN}$ to $3 \times \mathrm{N} + \mathrm{NCLIN} + 2 \times \mathrm{NCNLN}$. 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.
Replaced by E04UCF/E04UCA.

```
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
     double precision X(N), OBJF, OBJGRD(N)
New: SUBROUTINE  OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
     INTEGER     MODE, N, NSTATE, IUSER(*)
     double precision 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
     double precision 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(*)
     double precision X(N), C(NCNLN), CJAC(NROWJ,N), USER(*)
```

If NCNLN $= 0$, then the name of the dummy routine E04VDM/E54VDM (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/E04UCA contains four additional parameters as follows:

> ITER – INTEGER.
>
> R(LDR, N) – ***double precision*** array.
>
> IUSER($*$) – INTEGER array of dimension at least 1.
>
> USER($*$) – ***double precision*** array of dimension at least 1.

The minimum value of the parameter LIWORK must be increased from $3 \times \mathrm{N} + \mathrm{NCLIN} + \mathrm{NCNLN}$ to $3 \times \mathrm{N} + \mathrm{NCLIN} + 2 \times \mathrm{NCNLN}$. 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.
Replaced by F07ADF (DGETRF) and F07AJF (DGETRI).

```
Old: CALL F01AAF(A,IA,N,X,IX,WKSPCE,IFAIL)
New: CALL DGETRF(N,N,A,IA,IPIV,IFAIL)
     CALL F06QFF('General',N,N,A,IA,X,IX)
     CALL DGETRI(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 (DGETRF) computes the *LU* factorization of the matrix A, F06QFF copies the factorization from A to X, and F07AJF (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 (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.
Replaced by F01ABF.

```
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.
Replaced by F06EGF (DSWAP), F07FDF (DPOTRF) and F08SEF (DSYGST).

```
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 DPOTRF('L',N,B,IB,INFO)
     IF (INFO.EQ.0) THEN
        CALL DSYGST(1,'L',N,A,IA,B,IB,INFO)
     ELSE
        IFAIL = 1
     END IF
     CALL DSWAP(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.
Replaced by F06EGF (DSWAP) and F06YJF (DTRSM).

```
Old: CALL F01AFF(N,M1,M2,B,IB,DL,Z,IZ)
New: CALL DSWAP(N,DL,1,B,IB+1)
     CALL DTRSM('L','L','T','N',N,M2-M1+1,1.0D0,B,IB,Z(1,M1),IZ)
     CALL DSWAP(N,DL,1,B,IB+1)
```

F01AGF
Withdrawn at Mark 18.
Replaced by F08FEF (DSYTRD).

```
Old: CALL F01AGF(N,TOL,A,IA,D,E,E2)
New: CALL DSYTRD('L',N,A,IA,D,E(2),TAU,WORK,LWORK,INFO)
     E(1) = 0.0D0
     DO 10 I = 1, N
        E2(I) = E(I)*E(I)
  10 CONTINUE
```

where TAU is a ***double precision*** array of length at least $(\mathrm{N}-1)$, WORK is a ***real*** array of length at least (1) and LWORK is its actual length.

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

F01AHF
Withdrawn at Mark 18.
Replaced by F08FGF (DORMTR).

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

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

where WORK is a ***double precision*** array of length at least $(\mathrm{M2}-\mathrm{M1}+1)$, and LWORK is its actual length.

F01AJF
Withdrawn at Mark 18.
Replaced by F08FEF (DSYTRD) and F08FFF (DORGTR).

```
Old: CALL F01AJF(N,TOL,A,IA,D,E,Z,IZ)
New: CALL DSYTRD('L',N,A,IA,D,E(2),TAU,WORK,LWORK,INFO)
     E(1) = 0.0D0
     CALL F06QFF('L',N,N,A,IA,Z,IZ)
     CALL DORGTR('L',N,Z,IZ,TAU,WORK,LWORK,INFO)
```

where TAU is a ***double precision*** array of length at least $(\mathrm{N}-1)$, WORK is a ***real*** array of length at least $(\mathrm{N}-1)$ and LWORK is its actual length.

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

F01AKF
Withdrawn at Mark 18.
Replaced by F08NEF (DGEHRD).

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

where TAU is a ***double precision*** array of length at least $(\mathrm{N}-1)$, WORK is a ***real*** array of length at least (N) and LWORK is its actual length.

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

F01ALF

Withdrawn at Mark 18.
Replaced by F08NGF (DORMHR).

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

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

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

F01AMF

Withdrawn at Mark 18.
Replaced by F08NSF (ZGEHRD).

```
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 ZGEHRD(N,K,L,A,IA,TAU,WORK,LWORK,INFO)
```

where A is a ***complex*16*** array of dimension (IA, N), TAU is a ***complex*16*** array of length at least (N − 1), WORK is a ***complex*16*** array of length at least (N) and LWORK is its actual length.

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

F01ANF

Withdrawn at Mark 18.
Replaced by F08NUF (ZUNMHR).

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

```
Old: CALL F01ANF(K,L,IR,AR,IAR,AI,IAI,INTGER,ZR,IZR,ZI,IZI,N)
New: CALL ZUNMHR('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*16*** array of dimension (IA, N), TAU is a ***complex*16*** array of length at least (N − 1), Z is a ***complex*16*** array of dimension (IZ, IR), WORK is a ***complex*16*** array of length at least (IR) and LWORK is its actual length.

F01APF

Withdrawn at Mark 18.
Replaced by F06QFF and F08NFF (DORGHR).

The following replacement is valid only if the previous call to F01AKF has been replaced by a call to F08NEF (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 DORGHR(N,K,L,V,IV,TAU,WORK,LWORK,INFO)
```

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

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

F01ATF
Withdrawn at Mark 18.
Replaced by F08NHF (DGEBAL).

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

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

F01AUF
Withdrawn at Mark 18.
Replaced by F08NJF (DGEBAK).

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

F01AVF
Withdrawn at Mark 18.
Replaced by F08NVF (ZGEBAL).

```
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 ZGEBAL('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*16*** array of dimension (IA, N).

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

F01AWF
Withdrawn at Mark 18.
Replaced by F08NWF (ZGEBAK).

```
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 ZGEBAK('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*16*** array of dimension (IZ, M).

F01AXF

Withdrawn at Mark 18.
Replaced by F06EFF (DCOPY) and F08BEF (DGEQPF).

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

where WORK is a ***double precision*** array of length at least $(3 \times \mathrm{N})$.

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

F01AYF

Withdrawn at Mark 18.
Replaced by F08GEF (DSPTRD).

```
Old: CALL F01AYF(N,TOL,A,IA,D,E,E2)
New: CALL DSPTRD('U',N,A,D,E(2),TAU,INFO)
     E(1) = 0.0D0
     DO 10 I = 1, N
        E2(I) = E(I)*E(I)
  10 CONTINUE
```

where TAU is a ***double precision*** array of length at least $(\mathrm{N}-1)$.

F01AZF

Withdrawn at Mark 18.
Replaced by F08GGF (DOPMTR).

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

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

where WORK is a ***double precision*** array of length at least $(\mathrm{M2}-\mathrm{M1}+1)$.

F01BCF

Withdrawn at Mark 18.
Replaced by F08FSF (ZHETRD) and F08FTF (ZUNGTR).

```
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 ZHETRD('L',N,A,IA,D,E(2),TAU,WORK,LWORK,INFO)
     E(1) = 0.0D0
     CALL ZUNGTR('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*16*** array of dimension (IA,N), TAU is a ***complex*16*** array of length at least $(\mathrm{N}-1)$, WORK is a ***complex*16*** array of length at least $(\mathrm{N}-1)$, and LWORK is its actual length.

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

F01BDF
Withdrawn at Mark 18.
Replaced by F06EGF (DSWAP), F07FDF (DPOTRF) and F08SEF (DSYGST).

```
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 DPOTRF('L',N,B,IB,INFO)
     IF (INFO.EQ.0) THEN
        CALL DSYGST(2,'L',N,A,IA,B,IB,INFO)
     ELSE
        IFAIL = 1
     END IF
     CALL DSWAP(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.
Replaced by F06YFF (DTRMM).

```
Old: CALL F01BEF(N,M1,M2,B,IB,DL,V,IV)
New: CALL DSWAP(N,DL,1,B,IB+1)
     CALL DTRMM('L','L','N','N',N,M2-M1+1,1.0D0,B,IB,V(1,M1),IV)
     CALL DSWAP(N,DL,1,B,IB+1)
```

F01BNF
Withdrawn at Mark 17.
Replaced by F07FRF (ZPOTRF).

```
Old: CALL F01BNF(N,A,IA,P,IFAIL)
New: CALL ZPOTRF('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 ***double precision*** 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.
Replaced by F07FRF (ZPOTRF) and F07FWF (ZPOTRI).

```
Old: CALL F01BPF(N,A,IA,V,IFAIL)
New: CALL ZPOTRF('Upper',N,A,IA,IFAIL)
     CALL ZPOTRI('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.
Replaced by F07GDF (DPPTRF) and F07PDF (DSPTRF).

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 DPPTRF('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 DSPTRF('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 $\mathrm{N}(\mathrm{N}-1)/2$ to at least $\mathrm{N}(\mathrm{N}+1)/2$. It is important to note that for the calls of F07GDF (DPPTRF) and F07PDF (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 (DSPTRF), the INTEGER array IPIV of length N must be supplied.

F01BTF

Withdrawn at Mark 18.
Replaced by F07ADF (DGETRF).

```
Old: CALL F01BTF(N,A,IA,P,DP,IFAIL)
New: CALL DGETRF(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 (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 (DGETRF). If this value is required, it may be calculated after a call of F07ADF (DGETRF) by code similar to the following:

```
DP = 1.0D0
   DO 10 I = 1, N
      IF (I.NE.IPIV(I)) DP = -DP
10 CONTINUE
```

F01BWF

Withdrawn at Mark 18.
Replaced by F08HEF (DSBTRD).

```
Old: CALL F01BWF(N,M1,A,IA,D,E)
New: CALL DSBTRD('N','U',N,M1-1,A,IA,D,E(2),Q,1,WORK,INFO)
     E(1) = 0.0D0
```

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

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

F01BXF

Withdrawn at Mark 17.
Replaced by F07FDF (DPOTRF).

```
Old: CALL F01BXF(N,A,IA,P,IFAIL)
New: CALL DPOTRF('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).

F01CDF
Withdrawn at Mark 15.
Replaced by F01CTF.

```
Old: CALL F01CDF(A,B,C,M,N,IFAIL)
New: CALL F01CTF('N','N',M,N,1.0D0,B,M,1.0D0,C,M,A,M,IFAIL)
```

F01CEF
Withdrawn at Mark 15.
Replaced by F01CTF.

```
Old: CALL F01CEF(A,B,C,M,N,IFAIL)
New: CALL F01CTF('N','N',M,N,1.0D0,B,M,-1.0D0,C,M,A,M,IFAIL)
```

F01CGF
Withdrawn at Mark 15.
Replaced by F01CTF.

```
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+1,1.0D0,A(P,Q),MA,1.0D0,
    +             B(M1,N1),MB,A(P,Q),MA,IFAIL)
```

F01CHF
Withdrawn at Mark 15.
Replaced by F01CTF.

```
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+1,1.0D0,A(P,Q),MA,-1.0D0,
    +             B(M1,N1),MB,A(P,Q),MA,IFAIL)
```

F01CLF
Withdrawn at Mark 16.
Replaced by F06YAF (DGEMM).

```
Old: CALL F01CLF(A,B,C,N,P,M,IFAIL)
New: CALL DGEMM('N','T',N,P,M,1.0D0,B,N,C,P,0.0D0,A,N)
```

F01LBF
Withdrawn at Mark 18.
Replaced by F07BDF (DGBTRF).

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

where the size of array A must now have a leading dimension IA of at least $2 \times \mathrm{M1} + \mathrm{M2} + 1$. The array AL, its associated dimension parameter IL, and the parameter IV are not required for F07BDF (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.
Replaced by F08KEF (DGEBRD), F08KFF (DORGBR) or F08KGF (DORMBR).

```
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 DGEBRD(N,N,A,NRA,D,E(2),TAUQ,TAUP,WORK1,LWORK,INFO)
     IF (WANTB) THEN
        CALL DORMBR('Q','L','T',N,1,NA,NRA,TAUQ,B,N,WORK1,LWORK,INFO)
     ELSE IF (WANTQ) THEN
        CALL DORGBR('Q',N,N,N,A,NRA,TAUQ,WORK,LWORK,INFO)
     ELSE IF (WANTY) THEN
        CALL DORMBR('Q','R','N',LY,N,N,A,NRA,TAUQ,Y,NRY,WORK1,LWORK,
    +             INFO)
     ELSE IF (WANTZ) THEN
```

```
            CALL DORMBR('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
Withdrawn at Mark 19.
Replaced by F11JAF.

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.
Replaced by F07BRF (ZGBTRF).

```
Old: CALL F01NAF(N,ML,MU,A,NRA,TOL,IN,SCALE,IFAIL)
New: CALL ZGBTRF(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 $\mathrm{ML} + 1$ to $2 \times \mathrm{ML} + \mathrm{MU} + 1$ of A instead of rows 1 to $\mathrm{ML} + \mathrm{MU} + 1$. In F07BRF (ZGBTRF), the value returned in IN(N) has no significance as an indicator of near-singularity of the matrix.

F01QAF
Withdrawn at Mark 15.
Replaced by F08AEF (DGEQRF).

```
Old: CALL F01QAF(M,N,A,NRA,C,NRC,Z,IFAIL)
New: CALL DGEQRF(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 (DORGQR) and F08AGF (DORMQR) may be used to obtain Q explicitly and to transform by Q or Q^{T} respectively.

F01QBF
Withdrawn at Mark 15.
Replaced by F01QJF.

```
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 leading 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.
Replaced by F08AEF (DGEQRF).

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

where WORK is a ***double precision*** 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 (DGEQRF) are not the same as those returned by F01QCF. Subsequent calls to F01QDF or F01QEF must also be replaced by calls to F08AFF (DORGQR) or F08AGF (DORMQR) as shown below.

F01QDF
Withdrawn at Mark 18.
Replaced by F08AGF (DORMQR).

The following replacement is valid only if the previous call to F01QCF has been replaced by a call to F08AEF (DGEQRF) as shown above. It also assumes that the second 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 DORMQR('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.
Replaced by F08AFF (DORGQR).

The following replacement is valid only if the previous call to F01QCF has been replaced by a call to F08AEF (DGEQRF) as shown above. It also assumes that the first 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 DORGQR(M,NCOLQ,N,A,LDA,ZETA,WORK,LWORK,INFO)
```

where LWORK is the actual length of WORK.

F01QFF
Withdrawn at Mark 18.
Replaced by F08BEF (DGEQPF).

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 DGEQPF(M,N,A,LDA,PERM,ZETA,WORK,INFO)
```

where WORK is a ***double precision*** 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 (DGEQPF) are not the same as those returned by F01QFF. Subsequent calls to F01QDF or F01QEF must also be replaced by calls to F08AGF (DORMQR) or F08AFF (DORGQR) as shown above. Note also that the array PERM returned by F08BEF (DGEQPF) holds details of the interchanges in a different form than that returned by F01QFF.

F01RCF
Withdrawn at Mark 18.
Replaced by F08ASF (ZGEQRF).

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

where WORK is a ***complex*16*** 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 (ZGEQRF) are not the same as those returned by F01RCF. Subsequent calls to F01RDF or F01REF must also be replaced by calls to F08AUF (ZUNMQR) or F08ATF (ZUNGQR) as shown below.

F01RDF
Withdrawn at Mark 18.
Replaced by F08AUF (ZUNMQR).

The following replacement is valid only if the previous call to F01RCF has been replaced by a call to F08ASF (ZGEQRF) as shown above. It also assumes that the second 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 ZUNMQR('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.
Replaced by F08ATF (ZUNGQR).

The following replacement is valid only if the previous call to F01RCF has been replaced by a call to F08ASF (ZGEQRF) as shown above. It also assumes that the first argument of F01REF WHERET = '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 ZUNGQR(M,NCOLQ,N,A,LDA,THETA,WORK,LWORK,INFO)
```

where LWORK is the actual length of WORK.

F01RFF

Withdrawn at Mark 18.
Replaced by F08BSF (ZGEQPF).

The following replacement assumes that the first 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 ZGEQPF(M,N,A,LDA,PERM,THETA,CWORK,WORK,INFO)
```

where CWORK is a ***complex*16*** array of length at least (N).

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

F02 – Eigenvalues and Eigenvectors

F02AAF

Withdrawn at Mark 18.
Replaced by F08FAF (DSYEV).

```
Old: CALL F02AAF(A,IA,N,R,E,IFAIL)
New: CALL DSYEV('N','L',N,A,IA,R,WORK,LWORK,INFO)       IF (INFO.NE.0) THEN
...
```

where WORK is a ***double precision*** array of length at least $(3 \times N)$ and LWORK is its actual length. Larger values of LWORK, up to some optimal value, may improve performance.

F02ABF

Withdrawn at Mark 18.
Replaced by F08FAF (DSYEV).

```
Old: CALL F02ABF(A,IA,N,R,V,IV,E,IFAIL)
New: CALL F06QFF('L',N,N,A,IA,V,IV)
      CALL DSYEV('V','L',N,V,IV,R,WORK,LWORK,INFO)       IF (INFO.NE.0) THEN
...
```

where WORK is a ***double precision*** array of length at least (3 × N) and LWORK is its actual length. Larger values of LWORK, up to some optimal value, may improve performance. If F02ABF was called with the same array supplied for V and A, then the call to F06QFF may be omitted.

F02ADF
Withdrawn at Mark 18.
Replaced by F08SAF (DSYGV).

```
Old: CALL F02ADF(A,IA,B,IB,N,R,DE,IFAIL)
New:  CALL  DSYGV(1,'N','U',N,A,IA,B,IB,R,WORK,LWORK,INFO)          IF  (IN-
FO.NE.0) THEN         ...
```

where WORK is a ***double precision*** array of length at least (3 × N) and LWORK is its actual length. Larger values of LWORK, up to some optimal value, may improve performance.

Note that the call to F08SAF (DSYGV) 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.
Replaced by F08SAF (DSYGV).

```
Old: CALL F02AEF(A,IA,B,IB,N,R,V,IV,DL,E,IFAIL)
New: CALL F06QFF('U',N,N,A,IA,V,IV)
       CALL  DSYGV(1,'V','U',N,V,IV,B,IB,R,WORK,LWORK,INFO)         IF  (IN-
FO.NE.0) THEN         ...
```

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

Note that the call to F08SAF (DSYGV) will overwrite the upper triangle of the array B and leave the subdiagonal elements unchanged, whereas the call to F02AEF 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.
Replaced by F08NAF (DGEEV).

```
Old: CALL F02AFF(A,IA,N,RR,RI,INTGER,IFAIL)
New: CALL DGEEV('N','N',N,A,IA,RR,RI,VR,1,VI,1,     +           WORK,LWOR-
K,INFO)      IF (INFO.EQ.0) THEN      ....
```

where VR and VI are dummy arrays of length (1) (not used in this call), WORK is a ***double precision*** 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 F08NAF (DGEEV). Larger values of LWORK, up to some optimal value, may improve performance.

F02AGF
Withdrawn at Mark 18.
Replaced by F08NAF (DGEEV).

```
Old: CALL F02AGF(A,IA,N,RR,RI,VR,IVR,VI,IVI,INTGER,IFAIL)
New:   CALL   DGEEV(('N','V',N,A,IA,RR,RI,VL,LDVL,VR1,LDVR1,          +
WORK,LWORK,INFO)       IF (INFO.EQ.0) THEN C      Eigenvector infomation is
stored differently in VR1
C      VR(j)=VR1(j) if RI(j) = 0.0
C      VR(j)=VR1(j) and VI(j)=VR1(j+1) and
C       VR(j+1)=VR1(j) and VI(j+1) = - VR1(j+1) if RI(j)/= (not equals) 0
and
C      RI(j) = -RI(j+1)
      ....
```

where WORK is a ***double precision*** 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 F08NAF (DGEEV). Larger values of LWORK, up to some optimal value, may improve performance.

F02AJF

Withdrawn at Mark 18.
Replaced by F08NNF (ZGEEV).

```
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 ZGEEV('N','N',N,A,LDA,R,VL,1,VR,1,WORK,     +            LWORK,R-
WORK,INFO)
     IF (INFO.EQ.0) THEN         DO 30 I = 1, N
          RR(I) = real(R(I))
          RI(I) = imag(R(I))
 30    CONTINUE
        .... where A is a complex*16 array of dimension (LDA,N), LDA must be at
least max(1,N), R is a complex*16 array of dimension (N), VR and VL are dummy com-
plex*16 array of length (1) (not used in this call), RWORK is a double precision ar-
ray of length at least (2 × N), WORK is a complex*16 array of length at least
(2 × N) and LWORK is its actual length. Larger values of LWORK, up to some op-
timal value, may improve performance. The iteration counts (returned by
F02AJF in the array INTGER) are not available from F08NNF (ZGEEV).
```

F02AKF

Withdrawn at Mark 18.
Replaced by F08NNF (ZGEEV).

```
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 ZGEEV('N','V',N,A,LDA,R,VL,LDVL,VR1,LDVR,     +           WORK,-
LWORK,RWORK,INFO)      IF (INFO.EQ.0) THEN        DO 40 J = 1, N
          RR(J) = real(R(J))
          RI(J) = imag(R(J))
          DO 30 I = 1, N
             VR(I,J) = real(VR1(I,J))
             VI(I,J) = imag(VR1(I,J))
  30      CONTINUE
  40    CONTINUE          .... where A is a complex*16 array of dimension (LDA,N),
LDA is at least max(1,N), R is a complex*16 array of length (N), VL is a complex*16
array of dimension (1,1), LDVL is 1, VR1 is a complex*16 array of dimension
(LDVR,N), LDVR is at least max(1,N), RWORK is a double precision array of length at
least (2 × N), WORK is a complex*16 array of length at least (2 × N) and LWORK is
its actual length. Larger values of LWORK, up to some optimal value, may
improve performance. The iteration counts (returned by F02AKF in the array
INTGER) are not available from F08NNF (ZGEEV).
```

F02AMF

Withdrawn at Mark 18.
Replaced by F08JEF (DSTEQR).

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

where WORK is a ***double precision*** array of length at least $(2(\mathrm{N}-1))$.

F02ANF
Withdrawn at Mark 18.
Replaced by F08PSF (ZHSEQR).

```
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 ZHSEQR('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*16*** array of dimension (IH, N), R is a ***complex*16*** array of length (N), Z is a dummy ***complex*16*** array of length (1) (not used in this call), and WORK is a ***complex*16*** array of length at least (N).

F02APF
Withdrawn at Mark 18.
Replaced by F08PEF (DHSEQR).

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

where Z is a dummy ***double precision*** array of length (1) (not used in this call), and WORK is a ***double precision*** array of length at least (3 × N); the iteration counts (returned by F02APF in the array ICNT) are not available from F08PEF (DHSEQR).

F02AQF
Withdrawn at Mark 18.
Replaced by F08PEF (DHSEQR) and F08QKF (DTREVC).

```
Old: CALL F02AQF(N,K,L,EPS,H,IH,V,IV,RR,RI,INTGER,IFAIL)
New: CALL DHSEQR('S','V',N,K,L,H,IH,RR,RI,V,IV,WORK,1,INFO)
     CALL DTREVC('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 ***double precision*** array of length at least (3 × N); the iteration counts (returned by F02AQF in the array INTGER) are not available from F08PEF (DHSEQR); M is an integer which is set to N by F08QKF (DTREVC).

F02ARF
Withdrawn at Mark 18.
Replaced by F08PSF (ZHSEQR) and F08QXF (ZTREVC).

```
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 ZHSEQR('S','V',N,K,L,H,IH,R,V,IV,WORK,1,INFO)
     CALL ZTREVC('R','O',SELECT,N,H,IH,V,IV,V,IV,N,M,WORK,RWORK,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*16*** array of dimension (IH, N), R is a ***complex*16*** array of length (N), V is a ***complex*16*** array of dimension (IV, N), WORK is a ***complex*16*** array of length at least (2 × N) and

RWORK is a ***double precision*** array of length at least (N); M is an integer which is set to N by F08QXF (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 (ZGEHRD) and F08NTF (ZUNGHR). IH must be $\geq \max(1, \mathrm{N})$ and IV must be $\geq \max(1, \mathrm{N})$.

F02AVF
Withdrawn at Mark 18.
Replaced by F08JFF (DSTERF).

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

F02AWF
Withdrawn at Mark 18.
Replaced by F08FNF (ZHEEV).

```
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 ZHEEV('N','L',N,A,LDA,R,WORK,LWORK,RWORK,INFO)
     IF (INFO.EQ.0) THEN
     ...
```

where A is a ***complex*16*** array of dimension (LDA,N), LDA is at least max(1,N) RWORK is a ***double precision*** array of length at least $\max(1, 3 \times \mathrm{N} - 2)$, WORK is a ***complex*16*** array of length at least $(2 \times \mathrm{N})$ and LWORK is its actual length. Larger values of LWORK, up to some optimal value, may improve performance.

F02AXF
Withdrawn at Mark 18.
Replaced by F08FNF (ZHEEV).

```
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,LDA,V,LDV)
     CALL ZHEEV('V','L',N,V,LDV,R,WORK,LWORK,RWORK,INFO)
     IF (INFO.EQ.0) THEN
       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*16*** array of dimension (LDA, N), LDA is at least max(1,N), V is a ***complex*16*** array of dimension (LDV, N), LDV is at least max(1, N), RWORK is a ***double precision*** array of length at least $\max(1, 3 \times \mathrm{N} - 2)$, WORK is a ***complex*16*** array of length at least $(2 \times \mathrm{N})$ 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 may be omitted.

F02AYF
Withdrawn at Mark 18.
Replaced by F08JSF (ZSTEQR).

```
Old: CALL F02AYF(N,EPS,D,E,VR,IVR,VI,IVI,IFAIL)
New: CALL ZSTEQR('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*16*** array of dimension (IV, N), and WORK is a ***real*** array of length at least (2(N − 1)).

F02BBF
Withdrawn at Mark 19.
Replaced by F08FBF (DSYEVX).

```
Old:  CALL F02BBF(A,LDA,N,RLB,RUB,M,MM,R,V,LDV,D,E,E2,X,G,C,
     +             ICOUNT,IFAIL)
New:       CALL    DSYEVX('V','V','L',N,A,LDA,RLB,RUB,                 +
0,0,2*X02AMF(),MM,R,V,LDV,WORK,LWORK,IWORK,
     +              JFAIL,INFO)
 where R must have dimension at least max(1,N), WORK is a real array of length
at least (4×N), LWORK is its actual length, JFAIL is an integer array of
length at least max(1,N), 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).
Larger values of LWORK, up to some optimal value, may improve performance.
Arguments C, ICOUNT, X, G, E2, E and D are not used.
```

F02BCF
Withdrawn at Mark 19.
Replaced by F02ECF.

```
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
Withdrawn at Mark 19.
Replaced by F02GCF.

```
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*16*** array of dimension (IA, N), R is a ***complex*16*** array of dimension (N), V is a ***complex*16*** array of dimension (IV, M), WORK is a ***complex*16*** 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.
Replaced by F08JJF (DSTEBZ) and F08JKF (DSTEIN).

```
Old: CALL F02BEF(N,D,ALB,UB,EPS,EPS1,E,E2,M,MM,R,V,IV,ICOUNT,X,C,
    +            IFAIL)
New: CALL DSTEBZ(’V’,’B’,N,ALB,UB,0,0,EPS1,D,E(2),MM,NSPLIT,R,IBLOCK,
    +          ISPLIT,X,IWORK,INFO)
     CALL DSTEIN(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.
Replaced by F08JJF (DSTEBZ).

```
Old: CALL F02BFF(D,E,E2,N,M1,M2,MM12,EPS1,EPS,EPS2,IZ,R,WU)
New: CALL DSTEBZ(’I’,’E’,N,0.0D0,0.0D0,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 ***double precision*** array of length at least (4 × N), and IWORK is an integer array of length at least (3 × N).

F02BJF
Scheduled for withdrawal at Mark 23.
Replaced by F08WAF (DGGEV).

```
Old: CALL F02BJF(N,A,LDA,B,LDB,EPS1,ALFR,ALFI,BETA,MATV,V,LDV,ITER,IFAIL)
New: IF (MATV) THEN
       JOBVR = ’V’
     ELSE
       JOBVR = ’N’
     ENDIF
     CALL DGGEV(’N’,JOBVR,N,A,LDA,B,LDB,ALFR,ALFI,BETA,VL,LDVL,
    +            VR,LDVL,WORK,LWORK,INFO)
     IF (INFO.EQ.0) THEN
       ...
```

F02BKF
Withdrawn at Mark 18.
Replaced by F08PKF (DHSEIN).

```
Old: CALL F02BKF(N,M,H,IH,RI,C,RR,V,IV,B,IB,U,W,IFAIL)
New: CALL DHSEIN(’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 (DHSEIN) if there are complex conjugate pairs of eigenvalues.

F02BLF
Withdrawn at Mark 18.
Replaced by F08PXF (ZHSEIN).

```
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 ZHSEIN(’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*16*** array of dimension (IH, N), R is a ***complex*16*** array of length (N), V is a ***complex*16***, array of dimension (IV, M), M2 is an integer variable, WORK is a ***complex*16*** array of length at least (N × N), RWORK is a ***double precision*** array of length at least (N), and IFAILR is an integer array of length at least (N).

F02EAF
Scheduled for withdrawal at Mark 23.
Replaced by F08PAF (DGEES).

```
Old: CALL F02EAF(JOB,N,A,LDA,WR,WI,Z,LDZ,WORK,LWORK,IFAIL)
New: LOGICAL SELECT
     EXTERNAL SELECT
     ...
     IF (JOB.EQ.'N') THEN
       JOBVS = 'N'
     ELSE
       JOBVS = 'V'
     END IF
     CALL DGEES(JOBVS,'N',SELECT,N,A,LDA,0,WR,WI,Z,LDZ,WORK,
    +            LWORK,BWORK,INFO)
     IF (INFO.EQ.0) THEN
        ....
      LOGICAL FUNCTION SELECT(AR,AI)
     DOUBLE PRECISION AR, AI
     SELECT = .TRUE.
     RETURN
     ENDK
```

F02EBF
Scheduled for withdrawal at Mark 23.
Replaced by F08NAF (DGEEV).

```
Old: CALL F02EBF(JOB,N,A,LDA,WR,WI,VR,LDVR,VI,LDVI,WORK,LWORK,
    +            IFAIL)
New: IF (JOB.EQ.'N') THEN
       JOBVR = 'N'
     ELSE
       JOBVR = 'V'
     END IF
     CALL DGEEV('N',JOBVR,N,A,LDA,WR,WI,VL,LDVL,VR1,LDVR1,
    +            WORK,LWORK,INFO)
     IF (INFO.EQ.0) THEN
C      Eigenvector infomation is stored differently. C      For complex con-
jugate pairs (that is, corresponding C       to the j-th eigenvector such
that WI(j) is non-zero, C      and WI(j) = -WI(j+1)), the real and imaginary
parts C      of the first of the pair of eigenvectors are stored  C      as
consecutive columns of VR1: VR1(:,j), VR1(:,j+1). C      The second in the
pair is just the conjugate of the C      first, so can be constructed by ne-
gating the  C      elements in VR1(:,j+1). C      If the j-th eigenvector
is real (WI(j)=0), the  C      corresponding real eigenvector is stored in
the C      j-th column of VR1, VR1(1:N,j).
```

F02FAF
Scheduled for withdrawal at Mark 23.
Replaced by F08FAF (DSYEV).

```
Old: CALL F02FAF(JOB,UPLO,N,A,LDA,W,WORK,LWORK,IFAIL)
New: CALL DSYEV(JOB,UPLO,N,A,LDA,W,WORK,LWORK,INFO)
```

```
     IF (INFO.EQ.0) THEN
       ...
C    the workspace requirements are slightly different.
```

F02FCF

Scheduled for withdrawal at Mark 23.
Replaced by F08FBF (DSYEVX).

```
Old: CALL F02FCF(JOB,RANGE,UPLO,N,A,LDA,WL,WU,IL,IU,MEST,M,
    +            W,Z,LDZ,WORK,LWORK,IWORK,IFAIL)
New: CALL DSYEVX(JOB,RANGE,UPLO,N,A,LDA,WL,WU,IL,IU,ABSTOL,M,
    +            W,Z,LDZ,WORK,LWORK,IWORK,JFAIL,INFO)
     IF (INFO.EQ.0) THEN
       ...
C    the workspace requirements are slightly different.
```

F02FDF

Scheduled for withdrawal at Mark 23.
Replaced by F08SAF (DSYGV).

```
Old: CALL F02FDF(ITYPE,JOB,UPLO,N,A,LDA,B,LDB,W,WORK,LWORK,IFAIL)
New: CALL DSYGV(ITYPE,JOB,UPLO,N,A,LDA,B,LDB,W,WORK,LWORK,INFO)
     IF (INFO.EQ.0) THEN
       ...
C    the workspace requirements are slightly different.
```

F02FHF

Scheduled for withdrawal at Mark 23.
Replaced by F08UAF (DSBGV).

```
Old: CALL F02FHF(N,MA,A,LDA,MB,B,LDB,D,WORK,LWORK,IFAIL)
New: CALL DSBGV('N','U',N,MA,MB,A,LDA,B,LDB,D,Z,LDZ,WORK,INFO)
     IF (INFO.EQ.0) THEN
       ...
C    note that the eigenvalues appear in reverse order
C    note also the different workspace requirements
```

F02GAF

Scheduled for withdrawal at Mark 23.
Replaced by F08PNF (ZGEES).

```
Old: CALL F02GAF(JOB,N,A,LDA,W,Z,LDZ,RWORK,WORK,LWORK,IFAIL)
New: LOGICAL BWORK(1)       LOGICAL SELECT       EXTERNAL SELECT          ...
IF (JOB.EQ.'N') THEN          JOBVS = 'N'       ELSE          JOBVS = 'V'
END IF           CALL  ZGEES(JOBVS,'N',SELECT,N,A,LDA,0,W,Z,LDZ,        +
WORK,LWORK,RWORK,BWORK,INFO)      IF (INFO.NE.0) THEN         ... C    Note
also the different workspace requirements         ...      LOGICAL FUNCTION
SELECT(C)      COMPLEX*16 C      SELECT = .TRUE.      RETURN      END
```

F02GBF

Scheduled for withdrawal at Mark 23.
Replaced by F08NNF (ZGEEV).

```
Old: CALL F02GBF(JOB,N,A,LDA,W,V,LDV,RWORK,WORK,LWORK,IFAIL)
New: CALL ZGEEV('N',JOB,N,A,LDA,W,VL,LDVL,V,LDV,
    +            WORK,LWORK,RWORK,INFO)
     IF (INFO.EQ.0) THEN
       ...
```

F02GJF

Scheduled for withdrawal at Mark 23.
Replaced by F08WNF (ZGGEV).

```
Old: CALL F02GJF(N,AR,LDAR,AI,LDAI,BR,LDBR,BI,LDBI,EPS1,ALFR,
    +            ALFI,BETA,MATV,VR,LDVR,VI,LDVI,ITER,IFAIL)
New: IF (MATV) THEN
```

```
          JOBVR = 'V'
        ELSE
          JOBVR = 'N'
        END IF
      C
      C    Set A=AR + iAI and B = BR+iBI
      C
           CALL ZGGEV('N',JOBVR,N,A,LDA,B,LDB,ALPHA,BETA1,VL,LDVL,
          +            V,LDV,WORK,LWORK,RWORK,INFO)C
      C    Note results returned in COMPLEX*16 types, unlike F02GJF.
           IF (INFO.EQ.0) THEN
             ...
```

F02HAF
Scheduled for withdrawal at Mark 23.
Replaced by F08FNF (ZHEEV).

```
Old: CALL F02HAF(JOB,UPLO,N,A,LDA,W,RWORK,WORK,LWORK,IFAIL)
New: CALL ZHEEV(JOB,UPLO,N,A,LDA,W,WORK,LWORK,RWORK,INFO)
C    Note slightly different workspace requirements.
     IF (INFO.EQ.0) THEN
       ...
```

F02HCF
Scheduled for withdrawal at Mark 23.
Replaced by F08FPF (ZHEEVX).

```
Old: CALL F02HCF(JOB,RANGE,UPLO,N,A,LDA,WL,WU,IL,IU,MEST,M,
    +            W,Z,LDZ,WORK,LWORK,RWORK,IWORK,IFAIL)
New: CALL ZHEEVX(JOB,RANGE,UPLO,N,A,LDA,WL,WU,IL,IU,ABSTOL,M,
    +            W,Z,LDZ,WORK,LWORK,RWORK,IWORK,JFAIL,INFO)
C    Note slightly different workspace requirements.
     IF (INFO.EQ.0) THEN
       ...
```

F02HDF
Scheduled for withdrawal at Mark 23.
Replaced by F08SNF (ZHEGV).

```
Old: CALL F02HDF(ITYPE,JOB,UPLO,N,A,LDA,B,LDB,W,RWORK,WORK,
    +            LWORK,IFAIL)
New: CALL ZHEGV(ITYPE,JOB,UPLO,N,A,LDA,B,LDB,W,WORK,LWORK,
    +            RWORK,INFO)
C    Note slightly different workspace requirements.
     IF (INFO.EQ.0) THEN
       ...
```

F02SWF
Withdrawn at Mark 18.
Replaced by F08KEF (DGEBRD).

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) = 0.0D0
  10    CONTINUE
  20 CONTINUE
     CALL DGEBRD(N,N,A,LDA,D,E,TAUQ,TAUP,WORK,LWORK,INFO)
     IF (WANTQ) THEN
        CALL F06QFF('L',N,N,A,LDA,Q,LDQ)
        CALL DORGBR('Q',N,N,N,Q,LDQ,TAUQ,WORK,LWORK,INFO)
     END IF
     IF (NCOLY.GT.0) THEN
        CALL DORMBR('Q','L','T',N,NCOLY,N,A,LDA,TAUQ,Y,LDY,
    +              WORK,LWORK,INFO)
     END IF
```

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

F02SXF
Withdrawn at Mark 18.
Replaced by F08KFF (DORGBR) and F08KGF (DORMBR).

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

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

F02SYF
Withdrawn at Mark 18.
Replaced by F08MEF (DBDSQR).

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

where WORK is a ***double precision*** array of length at least $(4(N-1))$ unless NCOLB = NROWY = NCOLZ = 0.

F02SZF
Withdrawn at Mark 15.
Replaced by F08MEF (DBDSQR).

```
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 DBDSQR('U',N,NCVT,NRU,NCC,D,E(2),Z,NRZ,Y,NRY,B,N,WORK,INFO)
```

WORK must be a one-dimensional ***double precision*** array of length at least *lwork* given by:

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

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

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

F02UWF
Withdrawn at Mark 18.
Replaced by F08KSF (ZGEBRD).

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) = 0.0D0
  10    CONTINUE
  20 CONTINUE
     CALL ZGEBRD(N,N,A,LDA,D,E,TAUQ,TAUP,WORK,LWORK,INFO)
     IF (WANTQ) THEN
        CALL F06TFF('L',N,N,A,LDA,Q,LDQ)
        CALL ZUNGBR('Q',N,N,N,Q,LDQ,TAUQ,WORK,LWORK,INFO)
     END IF
     IF (NCOLY.GT.0) THEN
        CALL ZUNMBR('Q','L','C',N,NCOLY,N,A,LDA,TAUQ,Y,LDY,
    +                  WORK,LWORK,INFO)
     END IF
```

where TAUQ and TAUP are ***complex*16*** arrays of length at least (N), and LWORK is the actual length of WORK.

F02UXF
Withdrawn at Mark 18.
Replaced by F08KTF (ZUNGBR) or F08KUF (ZUNMBR).

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

```
Old: CALL F02UXF(N,A,LDA,NCOLY,Y,LDY,RWORK,CWORK,IFAIL)
New: IF (NCOLY.EQ.0) THEN
        CALL ZUNGBR('P',N,N,N,A,LDA,TAUP,CWORK,LWORK,INFO)
     ELSE
        CALL ZUNMBR('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.
Replaced by F08MSF (ZBDSQR).

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

where WORK is a ***double precision*** array of length at least $(4(\mathrm{N}-1))$ unless $\mathrm{NCOLB}=\mathrm{NROWY}=\mathrm{NCOLZ}=0$.

F02WAF
Withdrawn at Mark 16.
Replaced by F08KBF (DGESVD).

```
Old: CALL F02WAF(M,N,A,LDA,WANTB,B,SV,WORK,LWORK,IFAIL)
New: CALL DGESVD('N','O',M,N,A,LDA,SV,WORK,1,WORK,1,
    +                WORK,LWORK,INFO)      IF (INFO.EQ.0) THEN          ...
```

WORK must be a one-dimensional ***double precision*** array of length at least *lwork* given by: $\max(1, 3 \times \min(\mathrm{M},\mathrm{N}) + \max(\mathrm{M},\mathrm{N}), 5 \times \min(\mathrm{M},\mathrm{N}))$

Larger values of LWORK, up to some optimal value, may improve performance.

Please note that the facility to return $Q^{\mathrm{T}} \times b$ is not provided so arguments WANTB and B are not required. Instead, F08KBF (DGESVD) has an option to return the entire $\mathrm{M} * \mathrm{M}$ orthogonal matrix Q, referred to as U in its documentation, through its 8th argument.

F02WEF

Scheduled for withdrawal at Mark 23.
Replaced by F08KBF (DGESVD).

```
Old: CALL F02WEF(M,N,A,LDA,NCOLB,B,LDB,WANTQ,Q,LDQ,SV,WANTP,
    +            PT,LDPT,WORK,IFAIL)
New: IF (WANTQ) THEN
        JOBU = 'A'
     ELSE
        JOBU = 'N'
     END IF
     IF (WANTP) THEN
        JOBVT = 'A'
     ELSE
        JOBVT = 'N'
     END IF
C    Please note that the facility to return Q(t)B is not provided.
     CALL DGESVD(JOBU,JOBVT,M,N,A,LDA,SV,Q,LDQ,PT,LDPT,WORK,
    +            LWORK,INFO)
C    Note slightly different workspace requirements.
     IF (INFO.EQ.0) THEN
       ...
```

F02XEF

Scheduled for withdrawal at Mark 23.
Replaced by F08KPF (ZGESVD).

```
Old: CALL F02XEF(M,N,A,LDA,NCOLB,B,LDB,WANTQ,Q,LDQ,SV,WANTP,
    +            PH,LDPH,RWORK,CWORK,IFAIL)
New: IF (WANTQ) THEN
        JOBU = 'A'
     ELSE
        JOBU = 'N'
     END IF
     IF (WANTP) THEN
        JOBVT = 'A'
     ELSE
        JOBVT = 'N'
     END IF
C    Please note that the facility to return Q(h)B is not provided.
     CALL ZGESVD(JOBU,JOBVT,M,N,A,LDA,SV,Q,LDQ,PH,LDPH,CWORK,
    +            LWORK,RWORK,INFO)
C    Note slightly different workspace requirements.
     IF (INFO.EQ.0) THEN
        ...
```

F03 – Determinants

F03AGF

Withdrawn at Mark 17.
Replaced by F07HDF (DPBTRF).

```
Old: CALL F03AGF(N,M,A,IA,RL,IL,M1,D1,ID,IFAIL)
New: CALL DPBTRF('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 (DPBTRF), the array A holds the matrix packed using a different scheme to that used by F03AGF; see the routine document for details. F07HDF (DPBTRF) overwrites A with the Cholesky factor L (without reciprocating diagonal elements) rather than returning L in the array RL. F07HDF (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 (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 = 1.0D0
   ID = 0
   DO 30 I = 1, N
```

```
      D1 = D1*A(1,I)**2
10    IF (D1.GE.1.0D0) THEN
         D1 = D1*0.0625e0
         ID = ID + 4
         GO TO 10
      END IF
20    IF (D1.LT.0.0625e0) THEN
         D1 = D1*16.0D0
         ID = ID - 4
         GO TO 20
      END IF
30 CONTINUE
```

F03AHF
Withdrawn at Mark 17.
Replaced by F07ARF (ZGETRF).

```
Old: CALL F03AHF(N,A,IA,DETR,DETI,ID,RINT,IFAIL)
New: CALL ZGETRF(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 (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 (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 F07ARF (ZGETRF) 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(1.0D0,0.0D0)
   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.1.0D0) 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*16.0D0
         ID = ID - 4
         GO TO 20
      END IF
30 CONTINUE
   DETR = real(DET)
   DETI = imag(DET)
```

F03AMF
Withdrawn at Mark 17.
There is no replacement for this routine.

```
Old: CALL F01BNF(N,A,IA,P,IFAIL)
     CALL F03AMF(N,TEN,P,D1,D2)
New: CALL ZPOTRF('Upper',N,A,IA,IFAIL)
     D1 = 1.0D0
     D2 = 0.0D0
     DO 30 I = 1, N
        D1 = D1*real(A(I,I))**2
   10   IF (D1.GE.1.0D0) THEN
           D1 = D1*0.0625e0
           D2 = D2 + 4
           GO TO 10
        END IF
   20   IF (D1.LT.0.0625e0) THEN
           D1 = D1*16.0D0
           D2 = D2 - 4
           GO TO 20
```

```
          END IF
    30 CONTINUE
       IF (TEN) THEN
          I = D2
          D2 = D2*LOG10(2.0D0)
          D1 = D1*2.0D0**(I-D2/LOG10(2.0D0))
       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 (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 (ZPOTRF), array A contains the upper triangle of the matrix rather than the lower triangle.

F04 – Simultaneous Linear Equations

F04AAF

Scheduled for withdrawal at Mark 23.
Replaced by F07AAF (DGESV).

```
Old: CALL F04AAF(A,LDA,B,LDB,N,M,C,LDC,WKSPCE,IFAIL)
New: CALL DGESV(N,M,A,LDA,IPIV,B,LDB,INFO)
     IF (INFO.EQ.0) THEN
c      Answer now in B
       ...
```

F04ACF

Scheduled for withdrawal at Mark 23.
Replaced by F07HAF (DPBSV).

```
Old: CALL F04ACF(A,LDA,B,LDB,N,M,IR,C,LDC,RL,LDRL,M1,IFAIL)
New: CALL DPBSV('U',N,M,IR,AB,LDAB,B,LDB,INFO)
     IF (INFO.EQ.0) THEN
c      A and AB are stored differently.
c      AB may be regarded as the transpose of A, with the 'U' option.
c      Thus LDAB might be M+1
c      Answer now in B
       ...
```

F04ADF

Scheduled for withdrawal at Mark 23.
Replaced by F07ANF (ZGESV).

```
Old: CALL F04ADF(A,LDA,B,LDB,N,M,C,LDC,WKSPCE,IFAIL)
New: CALL ZGESV(N,M,A,LDA,IPIV,B,LDB,INFO)
     IF (INFO.EQ.0) THEN
c      Answer now in B
       ...
```

F04AKF

Withdrawn at Mark 17.
Replaced by F07ASF (ZGETRS).

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

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

F04ALF

Withdrawn at Mark 17.
Replaced by F07HEF (DPBTRS).

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

It is assumed that the matrix has been factorized by a call of F07HDF (DPBTRF) rather than F03AGF; see the F03 Chapter Introduction for details. A is the factorized matrix as returned by F07HDF (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 (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 (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.
Replaced by F06EFF (DCOPY), F06PJF (DTRSV) and F08AGF (DORMQR).

```
Old: CALL F04ANF(M,N,QR,IQR,ALPHA,IPIV,B,X,Z)
New: CALL DCOPY(N,ALPHA,1,QR,IQR+1)
     CALL DORMQR('L','T',M,1,N,QR,IQR,Y,B,M,Z,N,INFO)
     CALL DTRSV('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 ***double precision*** array as was used as the seventh 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 (DGEQPF) as shown above.

F04AQF

Withdrawn at Mark 16.
Replaced by F07GEF (DPPTRS) and F07PEF (DSPTRS).

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

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

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

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

```
Old: CALL F04AQF(N,M,RL,D,B,X)
New: CALL DCOPY(N,B,1,X,1)
     CALL DSPTRS('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 (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 (DCOPY), which simply copies array B to X, is not necessary. References to X in the calls of F07GEF (DPPTRS) and F07PEF (DSPTRS) may then be replaced by references to B, in which case B will be overwritten by the solution vector.

F04ARF

Scheduled for withdrawal at Mark 23.
Replaced by F07AAF (DGESV).

```
      Old: CALL F04ARF(A,LDA,B,N,C,WKSPCE,IFAIL)
      New: CALL DGESV(N,1,A,LDA,IPIV,B,N,INFO)
           IF (INFO.EQ.0) THEN
      c      Answer now in B
             ...
```

F04AWF

Withdrawn at Mark 17.
Replaced by F07FSF (ZPOTRS).

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

It is assumed that the matrix has been factorized by a call of F07FRF (ZPOTRF) rather than F01BNF; see the the F01 Chapter Introduction for details. A is the factorized matrix as returned by F07FRF (ZPOTRF). The array P is no longer required. INFO is an INTEGER diagnostic parameter; see the F07FSF (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 B and IX in the call of F07FSF (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.
Replaced by F07AEF (DGETRS).

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

It is assumed that the matrix has been factorized by a call of F07ADF (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.
Replaced by F07FEF (DPOTRS).

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

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

F04EAF

Scheduled for withdrawal at Mark 23.
Replaced by F07CAF (DGTSV).

```
      Old: CALL F04EAF(N,D,DU,DL,B,IFAIL)
      New: CALL DGTSV(N,1,DL(2),D,DU(2),B,N,INFO)
           IF (INFO.EQ.0) THEN
      c      Answer now in B
             ...
```

F04FAF

Scheduled for withdrawal at Mark 23.
Replaced by F07JAF (DPTSV), F07JDF (DPTTRF) and F07JEF (DPTTRS).

```
      Old: CALL F04FAF(JOB,N,D,E,B,IFAIL)
      New: IF (JOB.EQ.0)
             CALL DPTSV(N,1,D,E(2),B,1,INFO)
           IF (INFO.EQ.0) THEN
      c      Answer now in B
             ...
```

F04JAF

Scheduled for withdrawal at Mark 23.
Replaced by F08KAF (DGELSS).

```
Old: CALL F04JAF(M,N,A,LDA,B,TOL,SIGMA,IRANK,WORK,LWORK,IFAIL)
New: CALL DGELSS(M,N,1,A,LDA,B,1,S,RCOND,IRANK,WORK,LWORK,INFO)
c    Note workspace requirements are different.
     IF (INFO.EQ.0) THEN
C      Answer now in B
C      Singular values now in S, not WORK.
C      The standard error is not computed
       ...
```

F04JDF

Scheduled for withdrawal at Mark 23.
Replaced by F08KAF (DGELSS).

```
Old: CALL F04JDF(M,N,A,LDA,B,TOL,SIGMA,IRANK,WORK,LWORK,IFAIL)
New: CALL DGELSS(M,N,1,A,LDA,B,1,S,RCOND,IRANK,WORK,LWORK,INFO)
c    Note workspace requirements are different.
     IF (INFO.EQ.0) THEN
C      Answer now in B
C      Singular values now in S, not WORK.
C      The standard error is not computed
       ...
```

F04JLF

Scheduled for withdrawal at Mark 23.
Replaced by F08ZBF (DGGGLM).

```
Old: CALL F04JLF(M,N,P,A,LDA,B,LDB,D,X,Y,WORK,LWORK,IFAIL)
New: CALL DGGGLM(M,N,P,A,LDA,B,LDB,D,X,Y,WORK,LWORK,INFO)
C    Slight workspace differences
     IF (INFO.EQ.0) THEN
       ...
```

F04JMF

Scheduled for withdrawal at Mark 23.
Replaced by F08ZAF (DGGLSE).

```
Old: CALL F04JMF(M,N,P,A,LDA,B,LDB,C,D,X,WORK,LWORK,IFAIL)
New: CALL DGGLSE(M,N,P,A,LDA,B,LDB,C,D,X,WORK,LWORK,INFO)
C    Slight workspace differences
     IF (INFO.EQ.0) THEN
       ...
```

F04KLF

Scheduled for withdrawal at Mark 23.
Replaced by F08ZPF (ZGGGLM).

```
Old: CALL F04KLF(M,N,P,A,LDA,B,LDB,D,X,Y,WORK,LWORK,IFAIL)
New: CALL ZGGGLM(M,N,P,A,LDA,B,LDB,D,X,Y,WORK,LWORK,INFO)
     IF (INFO.EQ.0) THEN
       ...
```

F04KMF

Scheduled for withdrawal at Mark 23.
Replaced by F08ZNF (ZGGLSE).

```
Old: CALL F04KMF(M,N,P,A,LDA,B,LDB,C,D,X,WORK,LWORK,IFAIL)
New: CALL ZGGLSE(M,N,P,A,LDA,B,LDB,C,D,X,WORK,LWORK,INFO)
     IF (INFO.EQ.0) THEN
       ...
```

F04LDF
Withdrawn at Mark 18.
Replaced by F07BEF (DGBTRS).

```
Old: CALL F04LDF(N,M1,M2,IR,A,IA,AL,IL,IN,B,IB,IFAIL)
New: CALL DGBTRS('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 (DGBTRF) rather than F01LBF. The array AL and its associated dimension parameter IL are no longer required.

F04MAF
Withdrawn at Mark 19.
Replaced by F11JCF.

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
Withdrawn at Mark 19.
Replaced by F11GDF, F11GEF and F11GFF (or F11JCF or F11JEF).

If a user-defined preconditioner is required existing programs should be modified to call F11GDF, F11GEF and F11GFF. 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.
Replaced by F06SKF (ZTBSV) and F07BSF (ZGBTRS).

```
Old: CALL F04NAF(JOB,N,ML,MU,A,NRA,IN,B,TOL,IFAIL)
New: JOB = ABS(JOB)
     IF (JOB.EQ.1) THEN
        CALL ZGBTRS('No Transpose',N,ML,MU,1,A,NRA,IN,B,N,IFAIL)
     ELSE IF (JOB.EQ.2) THEN
        CALL ZGBTRS('Conjugate Transpose',N,ML,MU,1,A,NRA,IN,B,N,IFAIL)
     ELSE IF (JOB.EQ.3) THEN
        CALL ZTBSV('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 (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.

F11 – Large Scale Linear Systems

F11BAF
Withdrawn at Mark 21.
Replaced by F11BDF.

```
Old: CALL F11BAF(METHOD,PRECON,NORM,WEIGHT,ITERM,N,M,TOL,MAXITN,
    +            ANORM,SIGMAX,MONIT,LWREQ,IFAIL)
New: CALL F11BDF(METHOD,PRECON,NORM,WEIGHT,ITERM,N,M,TOL,MAXITN,
    +            ANORM,SIGMAX,MONIT,WORK,LWORK,LWREQ,IFAIL)
```

F11BDF contains two additional parameters as follows:

WORK(LWORK) – ***double precision*** array.

LWORK – INTEGER.

See the routine document for further information.

F11BBF
Withdrawn at Mark 21.
Replaced by F11BEF.

```
Old: CALL F11BBF(IREVCM,U,V,WORK,LWORK,IFAIL)
New: CALL F11BEF(IREVCM,U,V,WGT,WORK,LWORK,IFAIL)
```

WGT must be a one-dimensional ***double precision*** array of length at least n (the order of the matrix) if weights are to be used in the termination criterion, and 1 otherwise. Note that the call to F11BEF requires the weights to be supplied in WGT$(1:n)$ rather than WORK$(1:n)$. The minimum value of the parameter LWORK may also need to be changed.

F11BCF
Withdrawn at Mark 21.
Replaced by F11BFF.

```
Old: CALL F11BCF(ITN,STPLHS,STPRHS,ANORM,SIGMAX,IFAIL)
New: CALL F11BFF(ITN,STPLHS,STPRHS,ANORM,SIGMAX,WORK,LWORK,IFAIL)
```

F11BFF contains two additional parameters as follows:

WORK(LWORK) – ***double precision*** array.

LWORK – INTEGER.

See the routine document for further information.

F11GAF
Withdrawn at Mark 22.
Replaced by F11GDF.

```
Old: CALL F11GAF(METHOD,PRECON,SIGCMP,NORM,WEIGHT,ITERM,N,TOL,MAXITN,
    +            ANORM,SIGMAX,SIGTOL,MAXITS,MONIT,LWREQ,IFAIL)
New: CALL F11GDF(METHOD,PRECON,SIGCMP,NORM,WEIGHT,ITERM,N,TOL,MAXITN,
    +            ANORM,SIGMAX,SIGTOL,MAXITS,MONIT,LWREQ,WORK,LWORK,IFAIL)
```

F11GDF contains two additional parameters as follows:

WORK(LWORK) – ***double precision*** array.

LWORK – INTEGER.

See the routine document for further information.

F11GBF
Withdrawn at Mark 22.
Replaced by F11GEF.

```
Old: CALL F11GBF(IREVCM,U,V,WORK,LWORK,IFAIL)
New: CALL F11GEF(IREVCM,U,V,WGT,WORK,LWORK,IFAIL)
```

WGT must be a one-dimensional ***double precision*** array of length at least n (the order of the matrix) if weights are to be used in the termination criterion, and 1 otherwise. Note that the call to F11GEF requires the weights to be supplied in WGT$(1:n)$ rather than WORK$(1:n)$. The minimum value of the parameter LWORK may also need to be changed.

F11GCF
Withdrawn at Mark 22.
Replaced by F11GFF.

```
Old: CALL F11GCF(ITN,STPLHS,STPRHS,ANORM,SIGMAX,ITS,SIGERR,IFAIL)
New: CALL F11GFF(ITN,STPLHS,STPRHS,ANORM,SIGMAX,ITS,SIGERR,
    +            WORK,LWORK,IFAIL)
```

F11GFF contains two additional parameters as follows:

WORK(LWORK) – ***double precision*** array.

LWORK – INTEGER.

See the routine document for further information.

G01 – Simple Calculations on Statistical Data

G01BAF
Withdrawn at Mark 16.
Replaced by G01EBF.

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

G01BBF
Withdrawn at Mark 16.
Replaced by G01EDF.

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

G01BCF
Withdrawn at Mark 16.
Replaced by G01ECF.

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

G01BDF
Withdrawn at Mark 16.
Replaced by G01EEF.

```
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 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.
Replaced by G01FBF.

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

G01CBF
Withdrawn at Mark 16.
Replaced by G01FDF.

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

G01CCF
Withdrawn at Mark 16.
Replaced by G01FCF.

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

G01CDF
Withdrawn at Mark 16.
Replaced by G01FEF.

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

where TOL is set to the accuracy required.

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

G01CEF

Withdrawn at Mark 18.
Replaced by G01FAF.

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

G02 – Correlation and Regression Analysis

G02CJF

Withdrawn at Mark 16.
Replaced by G02DAF and G02DGF.

```
     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 ***double precision*** array and will not be referenced, e.g., SIGSQ could be used.

The array C no longer contains $\left(X^{\mathrm{T}}X\right)^{-1}$; however, $\left(X^{\mathrm{T}}X\right)^{-1}$ 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 $X^{\mathrm{T}}X$.

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

G04 – Analysis of Variance

G04ADF

Withdrawn at Mark 17.
Replaced by G04BCF.

```
     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,TABL,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 TABL starting at the second row, e.g., VAR(1) = TABL(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.

```
          double precision GMEAN
          INTEGER    IFAIL
          double precision C(NMAX,NMAX), EF(NMAX), TABL(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.
Replaced by G04BBF.

```
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,TABL,4,C,KMAX,NOBS,
    +           R,EF,0.0D0,0,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 TABL starting at row 2 and the values equivalent to those returned in the scalars F and FP are returned in TABL(2,4) and TABL(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.

```
 double precision BMEAN(1),C(KMAX,KMAX),EF(KMAX),R(NMAX),TABL(4,5),
+                WK(KMAX*KMAX+KMAX)
 INTEGER         IT(NMAX)
```

NMAX and KMAX are integers such that NMAX≥ N and KMAX≥ K.

G04AFF
Withdrawn at Mark 17.
Replaced by G04CAF.

```
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,TABL,ITOTAL,TMEAN,MAXT,E,
    +           IMEAN,SEMEAN,BMEAN,R,IWK,IFAIL)
```

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)         TABL(3,2)
     SS(2)         TABL(2,2)
     SS(i)         TABL(4,2)
     IDF(1)        TABL(3,1)
     IDF(2)        TABL(2,1)
     IDF(i)        TABL(4,1)
     F(1)          TABL(3,4)
     F(2)          TABL(2,4)
     F(3)          TABL(4,4)
     FP(1)         TABL(3,5)
     FP(2)         TABL(2,5)
     FP(3)         TABL(4,5)
```

Note how rows and columns have swapped.

The following additional declarations are required.

```
     double precision TABL(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 $\mathtt{NMAX} \geq M \times \mathrm{NR} \times \mathrm{NC}$ and $\mathrm{MAXT} \geq \mathrm{NR} + \mathrm{NC} + \mathrm{NR} \times \mathrm{NC}$.

G05 – Random Number Generators

G05CAF

Withdrawn at Mark 22.
Replaced by G05SAF.

```
Old:
    DO 20 I = 1, N
       X(I) = G05CAF(X(I))
20  CONTINUE
New: CALL G05SAF(N,STATE,X,IFAIL)
```

The integer array STATE in the call to G05SAF contains information on the base generator being used. This array must have been initialized prior to calling G05SAF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SAF is likely to be different from those produced by G05CAF.

G05CBF

Withdrawn at Mark 22.
Replaced by G05KFF.

```
Old: CALL G05CBF(I)
New: LSEED = 1
     SEED(1) = I
     GENID = 1
     SUBID = 1
     CALL G05KFF(GENID,SUBID,SEED,LSEED,STATE,LSTATE,IFAIL)
```

The integer array STATE in the call to G05KFF contains information on the base generator being used. The base generator is chosen via the integer parameters GENID and SUBID. The required length of the array STATE depends on the base generator chosen. Due to changes in the underlying code a sequence of values produced by using a random number generator initialized via a call to G05KFF is likely to be different to a sequence produced by a generator initialized by G05CBF, even if the same value for I is used.

G05CCF

Withdrawn at Mark 22.
Replaced by G05KGF.

```
Old: CALL G05CCF
New: GENID = 1
     SUBID = 1
     CALL G05KGF(GENID,SUBID,STATE,LSTATE,IFAIL)
```

The integer array STATE in the call to G05KGF contains information on the base generator being used. The base generator is chosen via the integer parameters GENID and SUBID. The required length of the array STATE depends on the base generator chosen.

G05CFF

Withdrawn at Mark 22.
Replaced by F06DFF.

```
Old: CALL G05CFF(IA,NI,XA,NX,IFAIL)
New: LSTATE = STATE(1)
     CALL F06DFF(LSTATE,STATE,1,CSTATE,1)
```

The state of the base generator for the group of routines G05KFF, G05KGF, G05KHF, G05KJF, G05NCF, G05NDF, G05PDF–G05PZF, G05RCF–G05RZF, G05S and G05T can be saved by simply creating a local copy of the array STATE. The first element of the STATE array contains the number of elements that are used by the random number generating routines, therefore either this number of elements can be copied, or the whole array (as defined in the calling program).

G05CGF

Withdrawn at Mark 22.
Replaced by F06DFF.

```
Old: CALL G05CGF(IA,NI,XA,NX,IFAIL)
New: LSTATE = CSTATE(1)
     CALL F06DFF(LSTATE,CSTATE,1,STATE,1)
```

The state of the base generator for the group of routines G05KFF, G05KGF, G05KHF, G05KJF, G05NCF, G05NDF, G05PDF–G05PZF, G05RCF–G05RZF, G05S and G05T can be restored by simply copying back the previously saved copy of the STATE array. The first element of the STATE array contains the number of elements that are used by the random number generating routines, therefore either this number of elements can be copied, or the whole array (as defined in the calling program).

G05DAF

Withdrawn at Mark 22.
Replaced by G05SQF.

```
Old: DO 10 I = 1, N
        X(I) = G05DAF(AA,BB)
  10 CONTINUE
New: A = MIN(AA,BB)
     B = MAX(AA,BB)
     IFAIL = 0
     CALL G05SQF(N,A,B,STATE,X,IFAIL)
```

The old routine G05DAF returns a single variate at a time, whereas the new routine G05SQF returns a vector of N values in one go. In G05SQF the minimum value must be held in the parameter A and the maximum in parameter B, therefore $A < B$. This was not the case for the equivalent parameters in G05DAF.

The integer array STATE in the call to G05SQF contains information on the base generator being used. This array must have been initialized prior to calling G05SQF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SQF is likely to be different from those produced by G05DAF.

G05DBF

Withdrawn at Mark 22.
Replaced by G05SFF.

```
Old: DO 10 I = 1, N
        X(I) = G05DBF(AA)
  10 CONTINUE
New: A = ABS(AA)
     IFAIL = 0
     CALL G05SFF(N,A,STATE,X,IFAIL)
```

The old routine G05DBF returns a single variate at a time, whereas the new routine G05SFF returns a vector of N values in one go. In G05SFF parameter A must be non-negative, this was not the case for the equivalent parameter in G05DBF.

The integer array STATE in the call to G05SFF contains information on the base generator being used. This array must have been initialized prior to calling G05SFF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SFF is likely to be different from those produced by G05DBF.

G05DCF
Withdrawn at Mark 22.
Replaced by G05SLF.

```
Old: DO 10 I = 1, N
          X(I) = G05DCF(A,BB)
   10 CONTINUE
New: B = ABS(BB)
     IFAIL = 0
     CALL G05SLF(N,A,B,STATE,X,IFAIL)
```

The old routine G05DCF returns a single variate at a time, whereas the new routine G05SLF returns a vector of N values in one go. In G05SLF the spread (parameter A) must be positive, this was not the case for the equivalent parameters in G05DCF.

The integer array STATE in the call to G05SLF contains information on the base generator being used. This array must have been initialized prior to calling G05SLF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SLF is likely to be different from those produced by G05DCF.

G05DDF
Withdrawn at Mark 22.
Replaced by G05SKF.

```
Old: DO 10 I = 1, N
          X(I) = G05DDF(XMU,SD)
   10 CONTINUE
New: VAR = SD**2
     IFAIL = 0
     CALL G05SKF(N,XMU,VAR,STATE,X,IFAIL)
```

The old routine G05DDF returns a single variate at a time, whereas the new routine G05SKF returns a vector of N values in one go. G05SKF expects the variance of the Normal distribution (parameter VAR), compared to G05DDF which expected the standard deviation.

The integer array STATE in the call to G05SKF contains information on the base generator being used. This array must have been initialized prior to calling G05SKF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SKF is likely to be different from those produced by G05DDF.

G05DEF
Withdrawn at Mark 22.
Replaced by G05SMF.

```
Old: DO 10 I = 1, N
          X(I) = G05DEF(XMU,SD)
   10 CONTINUE
New: VAR = SD**2
     IFAIL = 0
     CALL G05SMF(N,XMU,VAR,STATE,X,IFAIL)
```

The old routine G05DEF returns a single variate at a time, whereas the new routine G05SMF returns a vector of N values in one go. G05SMF expects the variance of the corresponding normal distribution (parameter VAR), compared to G05DEF which expected the standard deviation.

The integer array STATE in the call to G05SMF contains information on the base generator being used. This array must have been initialized prior to calling G05SMF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SMF is likely to be different to those produced by G05DEF.

G05DFF
Withdrawn at Mark 22.
Replaced by G05SCF.

```
Old: DO 10 I = 1, N
          X(I) = G05DFF(XMED,B)
  10 CONTINUE
New: SEMIQR = ABS(B)
     IFAIL = 0
     CALL G05SCF(N,XMED,SEMIQR,STATE,X,IFAIL)
```

The old routine G05DFF returns a single variate at a time, whereas the new routine G05SCF returns a vector of N values in one go. G05SCF expects the semi-interquartile range (parameter SEMIQR) to be non-negative, this was not the case for the equivalent parameter in G05DFF.

The integer array STATE in the call to G05SCF contains information on the base generator being used. This array must have been initialized prior to calling G05SCF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SCF is likely to be different to those produced by G05DFF.

G05DGF
Withdrawn at Mark 16.
Replaced by G05SJF.

```
Old: DO 20 I = 1, N
          X(I) = G05DGF(A,B,IFAIL)
20   CONTINUE
New: CALL G05SJF(N,A,B,STATE,X,IFAIL)
```

The old routine G05DGF returns a single variate at a time, whereas the new routine G05SJF returns a vector of N values in one go.

The integer array STATE in the call to G05SJF contains information on the base generator being used. This array must have been initialized prior to calling G05SJF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SJF is likely to be different from those produced by G05DGF.

G05DHF
Withdrawn at Mark 22.
Replaced by G05SDF.

```
Old: DO 10 I = 1, N
          X(I) = G05DHF(DF,IFAIL)
  10 CONTINUE
New: CALL G05SDF(N,DF,STATE,X,IFAIL)
```

The old routine G05DHF returns a single variate at a time, whereas the new routine G05SDF returns a vector of N values in one go.

The integer array STATE in the call to G05SDF contains information on the base generator being used. This array must have been initialized prior to calling G05SDF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SDF is likely to be different from those produced by G05DHF.

G05DJF
Withdrawn at Mark 22.
Replaced by G05SNF.

```
Old: DO 10 I = 1, N
          X(I) = G05DJF(DF,IFAIL)
  10 CONTINUE
New: CALL G05SNF(N,DF,STATE,X,IFAIL)
```

The old routine G05DJF returns a single variate at a time, whereas the new routine G05SNF returns a vector of N values in one go.

The integer array STATE in the call to G05SNF contains information on the base generator being used. This array must have been initialized prior to calling G05SNF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SNF is likely to be different from those produced by G05DJF.

G05DKF
Withdrawn at Mark 22.
Replaced by G05SHF.

```
Old: DO 10 I = 1, N
        X(I) = G05DKF(DF1,DF2,IFAIL)
 10 CONTINUE
New: CALL G05SHF(N,DF1,DF2,STATE,X,IFAIL)
```

The old routine G05DKF returns a single variate at a time, whereas the new routine G05SHF returns a vector of N values in one go.

The integer array STATE in the call to G05SHF contains information on the base generator being used. This array must have been initialized prior to calling G05SHF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SHF is likely to be different from those produced by G05DKF.

G05DLF
Withdrawn at Mark 16.
Replaced by G05SBF.

```
Old: DO 20 I = 1, N
         X(I) = G05DLF(A,B,IFAIL)
     CONTINUE
New: CALL G05SBF(N,A,B,STATE,X,IFAIL)
```

The old routine G05DLF returns a single variate at a time, whereas the new routine G05SBF returns a vector of N values in one go.

The integer array STATE in the call to G05SBF contains information on the base generator being used. This array must have been initialized prior to calling G05SBF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SBF is likely to be different from those produced by G05DLF.

G05DMF
Withdrawn at Mark 16.
Replaced by G05SBF.

```
Old: DO 20 I = 1, N
        X(I) = G05DMF(A,B,IFAIL)
     20 CONTINUE
```

There is no direct replacement for G05DMF. However, there are two methods for obtaining the same functionality.

Method 1:

```
New: CALL G05SBF(N,A,B,STATE,Y1,IFAIL)
     J = 0
     DO 20 I = 1, N
        IF (Y1(I).LT.1.0Exp;0) THEN
           J = J + 1
           X(J) = Y1(I)/(1.0D0-Y1(I))
        END IF
     20 CONTINUE
```

Method 2:

```
     New: CALL G05SBF(N,A,1.0D0,STATE,Y1,IFAIL)
          CALL G05SBF(N,B,1.0D0,STATE,Y2,IFAIL)
          J = 0
          DO 20 I = 1, N
             IF (Y2(I).LT.0.0Exp;0) THEN
                J = J + 1
                X(J) = Y1(I)/Y2(I)
             END IF
          20 CONTINUE
```

The old routine G05DMF returns a single variate at a time, whereas the routine G05SBF returns a vector of N values in one go.

The integer array STATE in the call to G05SBF contains information on the base generator being used. This array must have been initialized prior to calling G05SBF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. The sequence of values produced by the replacement code suggested above will be different to those produced by G05DMF.

G05DPF
Withdrawn at Mark 22.
Replaced by G05SSF.

```
     Old: DO 10 I = 1, N
             X(I) = G05DPF(A,B,IFAIL)
       10 CONTINUE
     New: CALL G05SSF(N,A,B,STATE,X,IFAIL)
```

The old routine G05DPF returns a single variate at a time, whereas the new routine G05SSF returns a vector of N values in one go.

The integer array STATE in the call to G05SSF contains information on the base generator being used. This array must have been initialized prior to calling G05SSF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SSF is likely to be different from those produced by G05DPF.

G05DRF
Withdrawn at Mark 22.
Replaced by G05TKF.

```
     Old: DO 10 I = 1, N
             X(I) = G05DRF(LAMDA,IFAIL)
       10 CONTINUE
     New: MODE = 3      CALL G05TJF(MODE,N,LAMBDA,R,LR,STATE,X,IFAIL)
```

The old routine G05DRF returns a single variate at a time, whereas the new routine G05TJF returns a vector of N values in one go. For efficiency, the new routine can make use of a reference vector, R. If, as in this case, the integer parameter MODE is set to 3, the real reference vector R is not referenced , and its length, LR, need only be at least one.

The integer array STATE in the call to G05TJF contains information on the base generator being used. This array must have been initialized prior to calling G05TJF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TJF is likely to be different to those produced by G05DRF.

G05DYF
Withdrawn at Mark 22.
Replaced by G05TLF.

```
     Old: DO 10 I = 1, N
             X(I) = G05DYF(AA,BB)
       10 CONTINUE
```

```
New: IFAIL = 0
     A = MIN(AA,BB)
     B = MAX(AA,BB)
     CALL G05TLF(N,A,B,STATE,X,IFAIL)
```

The old routine G05DYF returns a single variate at a time, whereas the new routine G05TLF returns a vector of N values in one go. In G05TLF the minimum value must be held in the parameter A and the maximum in parameter B, therefore $A \leq B$. This was not the case for the equivalent parameters in G05DYF.

The integer array STATE in the call to G05TLF contains information on the base generator being used. This array must have been initialized prior to calling G05TLF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TLF is likely to be different from those produced by G05DYF.

G05DZF
Withdrawn at Mark 22.
Replaced by G05TBF.

```
Old:       DO 20 I = 1, N
        X(I) = G05DZF(PP)
20   CONTINUE
New: P = MAX(0.0D0,MIN(PP,1.0D0))
     IFAIL = 0
     CALL G05TBF(N,P,STATE,X,IFAIL)
```

The old routine G05DZF returns a single variate at a time, whereas the new routine G05TBF returns a vector of N values in one go. The ***double precision*** parameter P in G05TBF must not be less than zero or greater than one, this was not the case for the equivalent parameter in G05DZF.

The integer array STATE in the call to G05TBF contains information on the base generator being used. This array must have been initialized prior to calling G05TBF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TBF is likely to be different to those produced by G05DZF.

G05EAF
Withdrawn at Mark 22.
Replaced by G05RZF.

```
Old: CALL G05EAF(XMU,M,C,LDC,EPS,R1,LR1,IFAIL)
New: MODE = 0
     CALL G05RZF(MODE,N,M,XMU,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

The old routine G05EAF sets up a reference vector for use by G05EZF. The functionality of both these routines has been combined into the single new routine G05RZF. Setting $\text{MODE} = 0$ in the call to G05RZF only sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EAF.

The length of the ***double precision*** reference vector, R, in G05RZF must be at least $\text{M} \times (\text{M} + 1) + 1$. This is longer than the equivalent parameter in G05EAF.

G05EBF
Withdrawn at Mark 22.
Replaced by G05TLF.

There is no direct replacement for routine G05EBF. G05EBF sets up a reference vector for use by G05EYF, this reference vector is no longer required. The replacement routine for G05EYF is G05TLF.

G05ECF
Withdrawn at Mark 22.
Replaced by G05TJF.

```
Old: CALL G05ECF(LAMBDA,R1,LR1,IFAIL)
     DO 10 I = 1, N
        X(I) = G05EYF(R1,LR1)
  10 CONTINUE
New: MODE = 2
     CALL G05TJF(MODE,N,LAMBDA,R,LR,STATE,X,IFAIL)
```

The old routine G05ECF sets up a reference vector for use by G05EYF. The replacement routine G05TJF is now used to both set up a reference vector and generate the required variates. Setting MODE = 0 in the call to G05TJF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05ECF. Setting MODE = 1 generates a series of variates from a reference vector mimicking the functionality of G05EYF for this particular distribution. Setting MODE = 2 initializes the reference vector and generates the variates in one go.

The routine G05EYF returns a single variate at a time, whereas the new routine G05TJF returns a vector of N values in one go.

The length of the ***double precision*** reference vector, R, in G05TJF, needs to be a different length from the equivalent parameter in G05ECF, see the documentation for more details.

The integer array STATE in the call to G05TJF contains information on the base generator being used. This array must have been initialized prior to calling G05TJF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TJF is likely to be different from those produced by a combination of G05ECF and G05EYF.

G05EDF
Withdrawn at Mark 22.
Replaced by G05TAF.

```
Old: CALL G05EDF(M,P,R1,LR1,IFAIL)
     DO 10 I = 1, N
        X(I) = G05EYF(R1,LR1)
  10 CONTINUE
New: MODE = 2
     CALL G05TAF(MODE,N,M,P,R,LR,STATE,X,IFAIL)
```

The old routine G05EDF sets up a reference vector for use by G05EYF. The replacement routine G05TAF is now used to both set up a reference vector and generate the required variates. Setting MODE = 0 in the call to G05TAF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EDF. Setting MODE = 1 generates a series of variates from a reference vector mimicking the functionality of G05EYF for this particular distribution. Setting MODE = 2 initializes the reference vector and generates the variates in one go.

The routine G05EYF returns a single variate at a time, whereas the new routine G05TAF returns a vector of N values in one go.

The length of the ***double precision*** reference vector, R, in G05TAF, needs to be a different length from the equivalent parameter in G05EDF, see the documentation for more details.

The integer array STATE in the call to G05TAF contains information on the base generator being used. This array must have been initialized prior to calling G05TAF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TAF is likely to be different from those produced by a combination of G05EDF and G05EYF.

G05EEF
Withdrawn at Mark 22.
Replaced by G05THF.

```
Old: CALL G05EEF(M,P,R1,LR1,IFAIL)
     DO 10 I = 1, N
```

```
           X(I) = G05EYF(R1,LR1)
     10 CONTINUE
   New: MODE = 2
        CALL G05THF(MODE,N,M,P,R,LR,STATE,X,IFAIL)
```

The old routine G05EEF sets up a reference vector for use by G05EYF. The replacement routine G05THF is now used to both set up a reference vector and generate the required variates. Setting MODE = 0 in the call to G05THF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EEF. Setting MODE = 1 generates a series of variates from a reference vector mimicking the functionality of G05EYF for this particular distribution. Setting MODE = 2 initializes the reference vector and generates the variates in one go.

The routine G05EYF returns a single variate at a time, whereas the new routine G05THF returns a vector of N values in one go.

The length of the ***double precision*** reference vector, R, in G05THF, needs to be a different length from the equivalent parameter in G05EEF, see the documentation for G05THF for more details.

The integer array STATE in the call to G05THF contains information on the base generator being used. This array must have been initialized prior to calling G05THF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05THF is likely to be different to those produced by a combination of G05EEF and G05EYF.

G05EFF
Withdrawn at Mark 22.
Replaced by G05TEF.

```
   Old: CALL G05EFF(NS,M,NP,R1,LR1,IFAIL)
        DO 10 I = 1, N
           X(I) = G05EYF(R1,LR1)
     10 CONTINUE
   New: MODE = 2
        CALL G05TEF(MODE,N,NS,NP,M,R,LR,STATE,X,IFAIL)
```

The old routine G05EFF sets up a reference vector for use by G05EYF. The replacement routine G05TEF is now used to both set up a reference vector and generate the required variates. Setting MODE = 0 in the call to G05TEF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EFF. Setting MODE = 1 generates a series of variates from a reference vector mimicking the functionality of G05EYF for this particular distribution. Setting MODE = 2 initializes the reference vector and generates the variates in one go.

The routine G05EYF returns a single variate at a time, whereas the new routine G05TEF returns a vector of N values in one go.

The length of the ***double precision*** reference vector, R, in G05TEF, needs to be a different length from the equivalent parameter in G05EFF, see the documentation for more details.

The integer array STATE in the call to G05TEF contains information on the base generator being used. This array must have been initialized prior to calling G05TEF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TEF is likely to be different to those produced by a combination of G05EFF and G05EYF.

G05EGF
Withdrawn at Mark 22.
Replaced by G05PHF.

```
   Old: CALL G05EGF(E,A,NA,B,NB,R,NR,VAR,IFAIL)
   New: AVAR = B(1)**2
        IQ = NB - 1
        IF (AVAR.GT.0.0D0) THEN
           DO 10 I = 1, IQ
              THETA(I) = -B(I+1)/B(1)
     10    CONTINUE
        ELSE
```

```
             DO 20 I = 1, IQ
                THETA(I) = 0.0D0
       20    CONTINUE
          END IF
          MODE = 0
          CALL G05PHF(MODE,N,E,NA,A,IQ,THETA,AVAR,R,LR,STATE,VAR,X,IFAIL)
```

The ***double precision*** vector THETA must be of length at least IQ = NB − 1.

The old routine G05EGF sets up a reference vector for use by G05EWF. The replacement routine G05PHF is now used to both set up a reference vector and generate the required variates. Setting MODE = 0 in the call to G05PHF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EGF. When MODE = 0, the integer array STATE in the call to G05PHF need not be set.

G05EHF
Withdrawn at Mark 22.
Replaced by G05NCF.

```
Old: CALL G05EHF(INDEX,N,IFAIL)
New: CALL G05NCF(INDEX,N,STATE,IFAIL)
```

The integer array STATE in the call to G05NCF contains information on the base generator being used. This array must have been initialized prior to calling G05NCF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05NCF is likely to be different from those produced by G05EHF.

G05EJF
Withdrawn at Mark 22.
Replaced by G05NDF.

```
Old: CALL G05EJF(IA,N,IZ,M,IFAIL)
New: CALL G05NDF(IA,N,IZ,M,STATE,IFAIL)
```

The integer array STATE in the call to G05NDF contains information on the base generator being used. This array must have been initialized prior to calling G05NDF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05NDF is likely to be different from those produced by G05EJF.

G05EWF
Withdrawn at Mark 22.
Replaced by G05PHF.

```
Old: CALL G05EGF(E,A,NA,B,NB,R,NR,VAR,IFAIL)
     DO 10 I = 1, N
        X(I) = G05EWF(R,NR,IFAIL)
  10 CONTINUE
New: AVAR = B(1)**2
     IQ = NB - 1
     IF (AVAR.GT.0.0D0) THEN
        DO 10 I = 1, IQ
           THETA(I) = -B(I+1)/B(1)
  10    CONTINUE
     ELSE
        DO 20 I = 1, IQ
           THETA(I) = 0.0D0
  20    CONTINUE
     END IF
     MODE = 2
     CALL G05PHF(MODE,N,E,NA,A,NB-1,THETA,AVAR,VAR,R,LR,STATE,X,IFAIL)
```

The ***double precision*** vector THETA must be of length at least IQ = NB − 1.

The old routine G05EGF sets up a reference vector for use by G05EWF. The replacement routine G05PHF is now used to both set up a reference vector and generate the required variates. Setting the integer parameter MODE to 0 in the call to G05PHF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EGF. Setting MODE to 1 generates a series of variates from a reference vector mimicking the functionality of G05EWF. Setting MODE to 2 initializes the reference vector and generates the variates in one go.

The integer array STATE in the call to G05PHF contains information on the base generator being used. This array must have been initialized prior to calling G05PHF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PHF is likely to be different to those produced by G05EGF.

G05EXF
Withdrawn at Mark 22.
Replaced by G05TDF.

```
Old: CALL G05EXF(P,NP,IP1,ITYPE,R1,LR1,IFAIL)
     DO 10 I = 1, N
        X(I) = G05EYF(R1,LR1)
  10 CONTINUE
New: MODE = 2
     CALL G05TDF(MODE,N,P,NP,IP1,ITYPE,R,LR,STATE,X,IFAIL)
```

The old routine G05EXF sets up a reference vector for use by G05EYF. The replacement routine G05TDF is now used to both set up a reference vector and generate the required variates. Setting MODE $= 0$ in the call to G05TDF sets up the ***double precision*** reference vector R and hence mimics the functionality of G05EXF. Setting MODE $= 1$ generates a series of variates from a reference vector mimicking the functionality of G05EYF for this particular distribution. Setting MODE $= 2$ initializes the reference vector and generates the variates in one go.

The routine G05EYF returns a single variate at a time, whereas the new routine G05TDF returns a vector of N values in one go.

The length of the ***double precision*** reference vector, R, in G05TDF, needs to be a different length from the equivalent parameter in G05EXF, see the documentation for more details.

The integer array STATE in the call to G05TDF contains information on the base generator being used. This array must have been initialized prior to calling G05TDF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05TDF is likely to be different from those produced by a combination of G05EXF and G05EYF.

G05EYF
Withdrawn at Mark 22.
Replaced by G05TDF.

There is no direct replacement routine for G05EYF.

G05EYF is designed to generate random draws from a distribution defined by a reference vector. These reference vectors are created by other routines in Chapter G05, for example G05EBF, which have themselves been superseded. In order to replace a call to G05EYF you must identify which NAG routine generated the reference vector being used and look up its replacement. For example, to replace a call to G05EYF preceded by a call to G05EBF, as in:

```
CALL G05EBF(M,IB,R,NR,IFAIL)
X = G05EYF(R,NR)
```

you would need to look at the replacement routine for G05EBF.

G05EZF
Withdrawn at Mark 22.
Replaced by G05RZF.

```
Old: CALL G05EAF(XMU,N,C,LDC,EPS,R1,LR1,IFAIL)
     DO 20 I = 1, N
```

```
         CALL G05EZF(CX,M,R,NR,IFAIL)
         DO 30 J = 1, M
            X(I,J) = CX(J)
30       CONTINUE
20    CONTINUE
New: MODE = 2
     CALL G05RZF(MODE,N,M,XMU,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

The old routine G05EAF sets up a reference vector for use by G05EZF. The functionality of both these routines has been combined into the single new routine G05RZF. Setting MODE = 2 in the call to G05RZF sets up the ***double precision*** reference vector R and generates the draws from the multivariate Normal distribution in one go.

The old routine G05EZF returns a single (M-dimensional vector) draw from the multivariate Normal distribution at a time, whereas the new routine G05RZF returns an N by M matrix of N draws in one go.

The integer array STATE in the call to G05RZF contains information on the base generator being used. This array must have been initialized prior to calling G05RZF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05RZF is likely to be different from those produced by G05EZF.

G05FAF
Withdrawn at Mark 22.
Replaced by G05SQF.

```
Old: CALL G05FAF(AA,BB,N,X)
New: A = MIN(AA,BB)
     B = MAX(AA,BB)
     IFAIL = 0
     CALL G05SQF(N,A,B,STATE,X,IFAIL)
```

In G05SQF the minimum value must be held in the parameter A and the maximum in parameter B, therefore $A \leq B$. This was not the case for the equivalent parameters in G05FAF.

The integer array STATE in the call to G05SQF contains information on the base generator being used. This array must have been initialized prior to calling G05SQF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SQF is likely to be different from those produced by G05FAF.

G05FBF
Withdrawn at Mark 22.
Replaced by G05SFF.

```
Old: CALL G05FBF(AA,N,X)
New: A = ABS(AA)
     IFAIL = 0
     CALL G05SFF(N,A,STATE,X,IFAIL)
```

In G05SFF parameter A must be non-negative, this was not the case for the equivalent parameter in G05FBF.

The integer array STATE in the call to G05SFF contains information on the base generator being used. This array must have been initialized prior to calling G05SFF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SFF is likely to be different from those produced by G05FBF.

G05FDF
Withdrawn at Mark 22.
Replaced by G05SKF.

```
Old: CALL G05FDF(XMU,SD,N,X)
New: VAR = SD**2
```

```
          IFAIL = 0
          CALL G05SKF(N,XMU,VAR,STATE,X,IFAIL)
```

G05SKF expects the variance of the normal distribution (parameter VAR), compared to G05FDF which expected the standard deviation.

The integer array STATE in the call to G05SKF contains information on the base generator being used. This array must have been initialized prior to calling G05SKF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SKF is likely to be different from those produced by G05FDF.

G05FEF
Withdrawn at Mark 22.
Replaced by G05SBF.

```
      Old: CALL G05FEF(A,B,N,X,IFAIL)
      New: CALL G05SBF(N,A,B,STATE,X,IFAIL)
```

The integer array STATE in the call to G05SBF contains information on the base generator being used. This array must have been initialized prior to calling G05SBF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SBF is likely to be different from those produced by G05FEF.

G05FFF
Withdrawn at Mark 22.
Replaced by G05SJF.

```
      Old: CALL G05FFF(A,B,N,X,IFAIL)
      New: CALL G05SJF(N,A,B,STATE,X,IFAIL)
```

The integer array STATE in the call to G05SJF contains information on the base generator being used. This array must have been initialized prior to calling G05SJF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SJF is likely to be different from those produced by G05FFF.

G05FSF
Withdrawn at Mark 22.
Replaced by G05SRF.

```
      Old: CALL G05FSF(VK,N,X,IFAIL)
      New: CALL G05SRF(N,VK,STATE,X,IFAIL)
```

The integer array STATE in the call to G05SRF contains information on the base generator being used. This array must have been initialized prior to calling G05SRF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05SRF is likely to be different from those produced by G05FSF.

G05GAF
Withdrawn at Mark 22.
Replaced by G05PXF.

```
      Old: CALL G05GAF(SIDE,INIT,M,N,A,LDA,WK,IFAIL)
      New: CALL G05PXF(SIDE,INIT,M,N,STATE,A,LDA,IFAIL)
```

The integer array STATE in the call to G05PXF contains information on the base generator being used. This array must have been initialized prior to calling G05PXF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PXF is likely to be different from those produced by G05GAF.

G05GBF
Withdrawn at Mark 22.
Replaced by G05PYF.

```
Old: CALL G05GBF(N,D,C,LDC,EPS,WK,IFAIL)
New: CALL G05PYF(N,D,EPS,STATE,C,LDC,IFAIL)
```

The integer array STATE in the call to G05PYF contains information on the base generator being used. This array must have been initialized prior to calling G05PYF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PYF is likely to be different from those produced by G05GBF.

G05HDF
Withdrawn at Mark 22.
Replaced by G05PJF.

```
Old: CALL G05HDF(MODE,K,IP,IQ,MEAN,PAR,LPAR,QQ,LDQQ,N,W,REF,LREF,
    +            IWORK,LIWORK,IFAIL)
New: IF (MODE.EQ.'S') THEN
        IMODE = 0
     ELSE IF (MODE.EQ.'C') THEN
        IMODE = 1
     ELSE IF (MODE.EQ.'R') THEN
        IMODE = 3
     END IF
     LL = 0
     DO 30 L = 1, IP
        DO 20 I = 1, K
           DO 10 J = 1, K
              LL = LL + 1
              PHI(I,J,L) = PAR(LL)
  10       CONTINUE
  20    CONTINUE
  30 CONTINUE
     DO 60 L = 1, IQ-1
        DO 50 I = 1, K
           DO 40 J = 1, K
              LL = LL + 1
              THETA(I,J,L) = PAR(LL)
  40       CONTINUE
  50    CONTINUE
  60 CONTINUE
     IF (MEAN.EQ.'M') THEN
        DO 70 I = 1, K
           LL = LL + 1
           XMEAN(I) = PAR(LL)
  70    CONTINUE
     ELSE
        DO 80 I = 1, K
           XMEAN(I) = 0.0D0
  80    CONTINUE
     END IF
       LDW = N        CALL G05PJF(IMODE,N,K,XMEAN,IP,PHI,IQ,THETA,QQ,LDQQ,-
REF,LREF,
    +            STATE,W,LDW,IWORK,LIWORK,IFAIL)
```

The integer parameter IMODE should be set to 0, 1 or 3 in place of the parameter MODE having settings of 'S', 'C' or 'R' respectively. The ***double precision*** array PHI should have length at least $\max(1, \mathrm{IP} \times (\mathrm{K} \times \mathrm{K}))$; if dimensioned as PHI(K, K, IP) (as in the above example) then PHI(i, j, l) will contain the element PAR$((l-1) \times k \times k + (i-1) \times k + j)$. The ***double precision*** array THETA should have length at least $\max(1, \mathrm{IQ} \times (\mathrm{K} \times \mathrm{K}))$; if dimensioned as THETA(K, K, IQ) (as in the above example) then THETA(i, j, l) will contain the element PAR$(\mathrm{IP} \times k \times k + (l-1) \times k \times k + (i-1) \times k + j)$. The ***double precision*** array XMEAN should have length at least K; if MEAN = 'M' then XMEAN(i) will contain the element PAR$(\mathrm{IP} + \mathrm{IQ} \times k \times k + i)$, otherwise XMEAN should contain an array of zero values.

The integer array STATE in the call to G05PJF contains information on the base generator being used. This array must have been initialized prior to calling G05PJF with a call to either G05KFF or G05KGF.

The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PJF is likely to be different from those produced by G05HDF.

G05HKF
Scheduled for withdrawal at Mark 24.
Replaced by G05PDF.

```
Old:    CALL    G05HKF(DIST,NUM,IP,IQ,THETA,GAMMA,DF,HT,ET,FCALL,RVEC,IGEN,-
ISEED,RWSAV,IFAIL)
New:   CALL   G05PDF(DIST,NUM,IP,IQ,THETA,GAMMA,DF,HT,ET,FCALL,R,LR,STATE,I-
FAIL)
```

The integer array STATE in the call to G05PDF contains information on the base generator being used. This array must have been initialized prior to calling G05PDF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PDF is likely to be different from those produced by G05HKF.

G05HLF
Scheduled for withdrawal at Mark 24.
Replaced by G05PEF.

```
Old:    CALL    G05HLF(DIST,NUM,IP,IQ,THETA,GAMMA,DF,HT,ET,FCALL,RVEC,IGEN,-
ISEED,RWSAV,IFAIL)
New:   CALL   G05PEF(DIST,NUM,IP,IQ,THETA,GAMMA,DF,HT,ET,FCALL,R,LR,STATE,I-
FAIL)
```

The integer array STATE in the call to G05PEF contains information on the base generator being used. This array must have been initialized prior to calling G05PEF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PEF is likely to be different from those produced by G05HLF.

G05HMF
Scheduled for withdrawal at Mark 24.
Replaced by G05PFF.

```
Old:    CALL    G05HMF(DIST,NUM,IP,IQ,THETA,GAMMA,DF,HT,ET,FCALL,RVEC,IGEN,-
ISEED,RWSAV,IFAIL)
New:   CALL   G05PFF(DIST,NUM,IP,IQ,THETA,GAMMA,DF,HT,ET,FCALL,R,LR,STATE,I-
FAIL)
```

The integer array STATE in the call to G05PFF contains information on the base generator being used. This array must have been initialized prior to calling G05PFF with a call to either G05KFF or G05KGF. The required length of the array STATE will depend on the base generator chosen during initialization. Due to changes in the underlying code the sequence of values produced by G05PFF is likely to be different from those produced by G05HMF.

G05HNF
Scheduled for withdrawal at Mark 24.
Replaced by G05PGF.

```
Old:  CALL  G05HNF(DIST,NUM,IP,IQ,THETA,DF,HT,ET,FCALL,RVEC,IGEN,ISEED,RWSA-
V,IFAIL)
New: CALL G05PGF(DIST,NUM,IP,IQ,THETA,DF,HT,ET,FCALL,RVEC,STATE,IFAIL)
```

G05KAF
Scheduled for withdrawal at Mark 24.
Replaced by G05SAF.

```
Old:
        DO 20 I = 1, N
           X(I) = G05KAF(IGEN,ISEED)
```

```
20      CONTINUE
New: CALL G05SAF(N,STATE,X,IFAIL)
```

G05KBF
Scheduled for withdrawal at Mark 24.
Replaced by G05KFF.

```
Old: G05KBF(IGEN,ISEED)
New:
IF (IGEN.EQ.0) THEN
   CALL G05KFF(1,1,ISEED,LSEED,STATE,LSTATE,IFAIL)
ELSE
   CALL G05KFF(2,IGEN,ISEED,LSEED,STATE,LSTATE,IFAIL)
END IF
```

G05KCF
Scheduled for withdrawal at Mark 24.
Replaced by G05KGF.

```
Old: CALL G05KCF(IGEN,ISEED)
New:
 IF (IGEN.EQ.0) THEN
   CALL G05KGF(1,1,STATE,LSTATE,IFAIL)
ELSE
   CALL G05KGF(2,IGEN,STATE,LSTATE,IFAIL)
END IF
```

G05KEF
Scheduled for withdrawal at Mark 24.
Replaced by G05TBF.

```
Old:
       DO 20 I = 1, N
          X(I) = G05KEF(P,IGEN,ISEED,IFAIL)
20     CONTINUE
New: CALL G05TBF(N,P,STATE,X,IFAIL)
```

G05LAF
Scheduled for withdrawal at Mark 24.
Replaced by G05SKF.

```
Old: CALL G05LAF(XMU,VAR,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SKF(N,XMU,VAR,STATE,X,IFAIL)
```

G05LBF
Scheduled for withdrawal at Mark 24.
Replaced by G05SNF.

```
Old: CALL G05LBF(DF,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SNF(N,DF,STATE,X,IFAIL)
```

G05LCF
Scheduled for withdrawal at Mark 24.
Replaced by G05SDF.

```
Old: CALL G05LCF(DF,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SDF(N,DF,STATE,X,IFAIL)
```

G05LDF
Scheduled for withdrawal at Mark 24.
Replaced by G05SHF.

```
Old: CALL G05LDF(DF1,DF2,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SHF(N,DF1,DF2,STATE,X,IFAIL)
```

G05LEF
Scheduled for withdrawal at Mark 24.
Replaced by G05SBF.

```
Old: CALL G05LEF(A,B,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SBF(N,A,B,STATE,X,IFAIL)
```

G05LFF
Scheduled for withdrawal at Mark 24.
Replaced by G05SJF.

```
Old: CALL G05LFF(A,B,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SJF(N,A,B,STATE,X,IFAIL)
```

G05LGF
Scheduled for withdrawal at Mark 24.
Replaced by G05SQF.

```
Old: CALL G05LGF(A,B,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SQF(N,A,B,STATE,X,IFAIL)
```

G05LHF
Scheduled for withdrawal at Mark 24.
Replaced by G05SPF.

```
Old: CALL G05LHF(XMIN,XMAX,XMED,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SPF(N,XMIN,XMED,XMAX,STATE,X,IFAIL)
```

G05LJF
Scheduled for withdrawal at Mark 24.
Replaced by G05SFF.

```
Old: CALL G05LJF(A,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SFF(N,A,STATE,X,IFAIL)
```

G05LKF
Scheduled for withdrawal at Mark 24.
Replaced by G05SMF.

```
Old: CALL G05LKF(XMU,VAR,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SMF(N,XMU,VAR,STATE,X,IFAIL)
```

G05LLF
Scheduled for withdrawal at Mark 24.
Replaced by G05SJF.

```
Old: CALL G05LLF(XMED,SEMIQR,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SCF(N,XMED,SEMIQR,STATE,X,IFAIL)
```

G05LMF
Scheduled for withdrawal at Mark 24.
Replaced by G05SSF.

```
Old: CALL G05LMF(A,B,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SSF(N,A,B,STATE,X,IFAIL)
```

G05LNF
Scheduled for withdrawal at Mark 24.
Replaced by G05SLF.

```
Old: CALL G05LNF(A,B,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SLF(N,A,B,STATE,X,IFAIL)
```

G05LPF

Scheduled for withdrawal at Mark 24.
Replaced by G05SRF.

```
Old: CALL G05LPF(VK,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SRF(N,VK,STATE,X,IFAIL)
```

G05LQF

Scheduled for withdrawal at Mark 24.
Replaced by G05SGF.

```
Old: CALL G05LQF(NMIX,A,WGT,N,X,IGEN,ISEED,IFAIL)
New: CALL G05SGF(N,NMIX,A,WGT,STATE,X,IFAIL)
```

G05LXF

Scheduled for withdrawal at Mark 24.
Replaced by G05RYF.

```
Old: CALL G05LXF(MODE,DF,M,XMU,C,LDC,N,X,LDX,IGEN,ISEED,R,LR,IFAIL)
New: CALL G05RYF(MODE,N,DF,M,XMU,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

G05LYF

Scheduled for withdrawal at Mark 24.
Replaced by G05RZF.

```
Old: G05LYF(MODE,M,XMU,C,LDC,N,X,LDX,IGEN,ISEED,R,LR,IFAIL)
New: G05RZF(MODE,N,M,XMU,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

G05LZF

Scheduled for withdrawal at Mark 24.
Replaced by G05RZF.

```
Old: CALL G05LZF(MODE,M,XMU,C,LDC,X,IGEN,ISEED,R,LR,IFAIL)
New:
N = 1
LDX = 1
CALL G05RZF(MODE,N,M,XMU,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

G05MAF

Scheduled for withdrawal at Mark 24.
Replaced by G05TLF.

```
Old: CALL G05MAF(A,B,N,X,IGEN,ISEED,IFAIL)
New: CALL G05TLF(N,A,B,STATE,X,IFAIL)
```

G05MBF

Scheduled for withdrawal at Mark 24.
Replaced by G05TCF.

```
Old: CALL G05MBF(MODE,P,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TCF(MODE,N,P,R,LR,STATE,X,IFAIL)
```

G05MCF

Scheduled for withdrawal at Mark 24.
Replaced by G05THF.

```
Old: CALL G05MCF(MODE,M,P,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05THF(MODE,N,M,P,R,LR,STATE,X,IFAIL)
```

G05MDF

Scheduled for withdrawal at Mark 24.
Replaced by G05TFF.

```
Old: CALL G05MDF(MODE,A,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TFF(MODE,N,A,R,LR,STATE,X,IFAIL)
```

G05MEF
Scheduled for withdrawal at Mark 24.
Replaced by G05TKF.

```
Old: CALL G05MEF(M,VLAMDA,X,IGEN,ISEED,IFAIL)
New: CALL G05TKF(M,VLAMBDA,STATE,X,IFAIL)
```

G05MJF
Scheduled for withdrawal at Mark 24.
Replaced by G05TAF.

```
Old: CALL G05MJF(MODE,M,P,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TAF(MODE,N,M,P,R,LR,STATE,X,IFAIL)
```

G05MKF
Scheduled for withdrawal at Mark 24.
Replaced by G05TJF.

```
Old: CALL G05MKF(MODE,LAMBDA,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TJF(MODE,N,LAMBDA,R,LR,STATE,X,IFAIL)
```

G05MLF
Scheduled for withdrawal at Mark 24.
Replaced by G05TEF.

```
Old: CALL G05MLF(MODE,NS,NP,M,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TEF(MODE,N,NS,NP,M,R,LR,STATE,X,IFAIL)
```

G05MRF
Scheduled for withdrawal at Mark 24.
Replaced by G05TGF.

```
Old: CALL G05MRF(MODE,M,K,P,N,X,LDX,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TGF(MODE,N,M,K,P,R,LR,STATE,X,LDX,IFAIL)
```

G05MZF
Scheduled for withdrawal at Mark 24.
Replaced by G05TDF.

```
Old: CALL G05MZF(MODE,P,NP,IP1,ITYPE,N,X,IGEN,ISEED,R,NR,IFAIL)
New: CALL G05TDF(MODE,N,P,NP,IP1,ITYPE,R,LR,STATE,X,IFAIL)
```

G05NAF
Scheduled for withdrawal at Mark 24.
Replaced by G05NCF.

```
Old: CALL G05NAF(INDEX,N,IGEN,ISEED,IFAIL)
New: CALL G05NCF(INDEX,N,STATE,IFAIL)
```

G05NBF
Scheduled for withdrawal at Mark 24.
Replaced by G05NDF.

```
Old: CALL G05NBF(IPOP,N,ISAMPL,M,IGEN,ISEED,IFAIL)
New: CALL G05NDF(IPOP,N,ISAMPL,M,STATE,IFAIL)
```

G05PAF
Scheduled for withdrawal at Mark 24.
Replaced by G05PHF.

```
Old:  CALL  G05PAF(MODE,XMEAN,IP,PHI,IQ,THETA,AVAR,VAR,N,X,IGEN,ISEED,R,NR,I-
FAIL)
New: CALL G05PHF(MODE,N,XMEAN,IP,PHI,IQ,THETA,AVAR,R,LR,STATE,VAR,X,IFAIL)
```

G05PCF

Scheduled for withdrawal at Mark 24.
Replaced by G05PJF.

```
Old:  CALL  G05PCF(MODE,K,XMEAN,IP,PHI,IQ,THETA,VAR,LDV,N,X,IGEN,ISEED,R,N-
R,IWORK,LIWORK,IFAIL)
New: CALL G05PJF(MODE,N,K,XMEAN,IP,PHI,IQ,THETA,VAR,LDV,R,LR,STATE,X,LDX,I-
FAIL)
```

G05QAF

Scheduled for withdrawal at Mark 24.
Replaced by G05PXF.

```
Old: CALL G05QAF(SIDE,INIT,M,N,A,LDA,IGEN,ISEED,WK,IFAIL)
New: CALL G05PXF(SIDE,INIT,M,N,STATE,A,LDA,IFAIL)
```

G05QBF

Scheduled for withdrawal at Mark 24.
Replaced by G05PYF.

```
Old: CALL G05QBF(N,D,C,LDC,EPS,IGEN,ISEED,WK,IFAIL)
New: CALL G05PYF(N,D,EPS,STATE,C,LDC,IFAIL)
```

G05QDF

Scheduled for withdrawal at Mark 24.
Replaced by G05PZF.

```
Old:  CALL  G05QDF(MODE,NROW,NCOL,TOTR,TOTC,X,LDX,IGEN,ISEED,R,NR,IW,LIW,I-
FAIL)
New: CALL G05PZF(MODE,NROW,NCOL,TOTR,TOTC,R,LR,STATE,X,LDX,IFAIL)
```

G05RAF

Scheduled for withdrawal at Mark 24.
Replaced by G05RDF.

```
Old: CALL G05RAF(MODE,M,C,LDC,N,X,LDX,IGEN,ISEED,R,LR,IFAIL)
New: CALL G05RDF(MODE,N,M,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

G05RBF

Scheduled for withdrawal at Mark 24.
Replaced by G05RCF.

```
Old: CALL G05RBF(MODE,DF,M,C,LDC,N,X,LDX,IGEN,ISEED,R,LR,IFAIL)
New: CALL G05RCF(MODE,N,DF,M,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

G05YAF

Scheduled for withdrawal at Mark 23.
Replaced by G05YLF and G05YMF.

Faure quasi random numbers

```
Old: CALL G05YAF(.TRUE.,'F',ISKIP,IDIM,QUAS,IREF,IFAIL)
New: CALL G05YLF(4,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YAF(.FALSE.,'F',ISKIP,IDIM,QUAS,IREF,IFAIL)
New: CALL G05YMF(1,2,QUAS,LDQUAS,IREF,IFAIL)
```

Sobol quasi random numbers

```
Old: CALL G05YAF(.TRUE.,'S',ISKIP,IDIM,QUAS,IREF,IFAIL)
New: CALL G05YLF(2,IDIM,IREF,LIREF,ISKIP,IFAIL)
```

```
Old: CALL G05YAF(.FALSE.,'S',ISKIP,IDIM,QUAS,IREF,IFAIL)
New: CALL G05YMF(1,2,QUAS,LDQUAS,IREF,IFAIL)
```

Neiderreiter quasi random numbers

```
Old: CALL G05YAF(.TRUE.,'N',ISKIP,IDIM,QUAS,IREF,IFAIL)
New: CALL G05YLF(3,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YAF(.FALSE.,'N',ISKIP,IDIM,QUAS,IREF,IFAIL)
New: CALL G05YMF(1,2,QUAS,LDQUAS,IREF,IFAIL)
```

G05YBF

Scheduled for withdrawal at Mark 23.
Replaced by G05YLF and either G05YJF or G05YKF.

This routine has been replaced by a suite of routines consisting of the relevant initialization routine followed by one of two possible generator routines.

Faure quasi random numbers with Gaussian probability:

```
Old: CALL G05YBF(.TRUE.,'F',.FALSE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YLF(4,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YBF(.FALSE.,'F',.FALSE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YJF(MEAN,STD,N,QUASI,IREF,IFAIL)
```

Sobol quasi random numbers with Gaussian probability:

```
Old: CALL G05YBF(.TRUE.,'S',.FALSE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YLF(2,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YBF(.FALSE.,'S',.FALSE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YJF(MEAN,STD,N,QUASI,IREF,IFAIL)
```

Neiderreiter quasi random numbers with Gaussian probability:

```
Old: CALL G05YBF(.TRUE.,'N',.FALSE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YLF(3,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YBF(.FALSE.,'N',.FALSE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YJF(MEAN,STD,N,QUASI,IREF,IFAIL)
```

Faure quasi random numbers with log Normal probability:

```
Old: CALL G05YBF(.TRUE.,'F',.TRUE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YLF(4,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YBF(.FALSE.,'F',.TRUE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YKF(MEAN,STD,N,QUASI,IREF,IFAIL)
```

Sobol quasi random numbers with log Normal probability:

```
Old: CALL G05YBF(.TRUE.,'S',.TRUE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YLF(2,IDIM,IREF,LIREF,ISKIP,IFAIL)


Old: CALL G05YBF(.FALSE.,'S',.TRUE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YKF(MEAN,STD,N,QUASI,IREF,IFAIL)
```

Neiderreiter quasi random numbers with log Normal probability:

```
Old: CALL G05YBF(.TRUE.,'N',.TRUE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
New: CALL G05YLF(3,IDIM,IREF,LIREF,ISKIP,IFAIL)
```

```
     Old: CALL G05YBF(.FALSE.,'N',.TRUE.,MEAN,STD,ISKIP,IDIM,QUASI,IREF,IFAIL)
     New: CALL G05YKF(MEAN,STD,N,QUASI,IREF,IFAIL)
```

G05YCF

Scheduled for withdrawal at Mark 24.
Replaced by G05YLF.

```
     Old: CALL G05YCF(IDIM,IREF,IFAIL)
     New: GENID = 4
          CALL G05YLF(GENID,IDIM,IREF,LIREF,ISKIP,IFAIL)
```

G05YDF

Scheduled for withdrawal at Mark 24.
Replaced by G05YMF.

```
     Old: CALL G05YDF(N,QUASI,IREF,IFAIL)
     New: CALL G05YMF(N,QUAS,LDQUAS,IREF,IFAIL)
```

G05YEF

Scheduled for withdrawal at Mark 24.
Replaced by G05YLF.

```
     Old: CALL G05YEF(IDIM, IREF, ISKIP, IFAIL)
     New: GENID = 2
          CALL G05YLF(GENID,IDIM,IREF,LIREF,ISKIP,IFAIL)
```

G05YFF

Scheduled for withdrawal at Mark 24.
Replaced by G05YMF.

```
     Old: CALL G05YFF(N,QUASI,IREF,IFAIL)
     New: CALL G05YMF(N,QUAS,LDQUAS,IREF,IFAIL)
```

G05YGF

Scheduled for withdrawal at Mark 24.
Replaced by G05YLF.

```
     Old: CALL G05YGF(IDIM,IREF,ISKIP,IFAIL)
     New: GENID = 3
          CALL G05YLF(GENID,IDIM,IREF,LIREF,ISKIP,IFAIL)
```

G05YHF

Scheduled for withdrawal at Mark 24.
Replaced by G05YMF.

```
     Old: CALL G05YHF(N,QUASI,IREF,IFAIL)
     New: CALL G05YMF(N,RCORD,QUAS,LDQUAS,IREF,IFAIL)
```

G05ZAF

Withdrawn at Mark 22.
There is no replacement for this routine.

G08 – Nonparametric Statistics

G08ABF

Withdrawn at Mark 16.
Replaced by G08AGF.

```
     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 = 0.0D0
          CALL G08AGF(N,Z,XME,'Lower-tail','No-zeros',W,WNOR,P,
         +            N1,W1,IFAIL)
```

W1 is a ***double precision*** work array of dimension (3 × N). The ***double precision*** array W2 is no longer required. WNOR returns the normalized Wilcoxon test statistic. The ***double precision*** array Z, of dimension (N), contains the difference between the paired sample observations, and by setting the ***double precision*** 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.
Replaced by G08AHF, G08AJF and G08AKF.

```
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 ***double precision*** arrays, of dimensions N1 and N2, where N2 $= N -$ N1, rather than consecutively in one array as in 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 ***double precision*** 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.
Replaced by G08CBF.

```
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 ***double precision*** array of dimension (2) 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.
Replaced by G13DMF.

```
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 ***double precision*** 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.

G13DCF

Scheduled for withdrawal at Mark 24.
Replaced by G13DDF.

```
Old:    CALL    G13DCF(K,N,IP,IQ,MEAN,PAR,NPAR,QQ,KMAX,W,PARHLD,EXACT,IPRINT,
*            CGETOL,MAXCAL,ISHOW,NITER,RLOGL,V,G,CM,LDCM,WORK,LWORK,       *
IW,LIW,IFAIL) New: CALL G13DDF(K,N,IP,IQ,MEAN,PAR,NPAR,QQ,KMAX,W,PARHLD,EX-
ACT,IPRINT,      *            CGETOL,MAXCAL,ISHOW,NITER,RLOGL,V,G,CM,LDCM,I-
FAIL)
```

The workspace arguments WORK, LWORK, IW and LIW are no longer required in the call to G13DDF.

H – Operations Research

H02BAF

Withdrawn at Mark 15.
Replaced by H02BBF.

```
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)
        double precision BIGBND, TOLFES, TOLIV, ROPT
        double precision RA(MM,N), RX(N), CVEC(N), BL(N+M), BU(N+M),
                    RWORK(LRWORK)
        DO 10 J = 1, N
           INTVAR(J) = 1
           CVEC(J) = A(1,J)
           RX(J) = 1.0D0
           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) = 0.0D0
           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 = 0.0D0
        TOLFES = 0.0D0
        MAXDPT = 3*N/2
        IFAIL = 0
        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 and Searching

M01AJF
Withdrawn at Mark 16.
Replaced by M01CAF, M01DAF and M01ZAF.

```
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.
Replaced by M01CAF, M01DAF and M01ZAF.

```
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.

M01APF
Withdrawn at Mark 16.
Replaced by M01CAF.

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

P01 – Error Trapping

P01ABF
Scheduled for withdrawal at Mark 24.
There is no replacement for this routine.