1 SUMMARY

This subroutine **solves a system of nonlinear equations**

$$r_i(x_1,x_2,...,x_n) = f_i(x_1,x_2,...,x_n) + \sum_{j=1}^{n} a_{ij}x_j = 0, \quad i=1,2,...,m, m \geq n,$$

in the sense that it searches for a local minimum of the sum of squares

$$S = \sum_{i=1}^{m} [\, r_i(x_1,x_2,...,x_n)\,]^2.$$

It is closely related to VA27 and is designed specifically for the case where the matrices $\{\partial f_i/\partial x_j\}$ and $\{a_{ij}\}$ are sparse (that is, contain a large number of zeros). It uses the Marquardt algorithm (D. W. Marquardt 'An algorithm for least squares estimation of nonlinear parameters', J. SIAM, **11**, 1963, 431-441). The user must supply a_{ij}, an initial estimate of $x_1,x_2,...,x_n$, and code that calculates each f_i and (optionally) each $\partial f_i/\partial x_j$.

ATTRIBUTES — **Version:** 2.0.0. (23rd May 2022) **Types:** Real (double). **Calls:** _DOT, MA57, TD22. **Original date:** June 2001. **Remark:** NS23 is a threadsafe version of NS13. **Origin:** J. K. Reid, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

'Reverse communication' is employed to permit the user to access any data needed to calculate the functions f_i and their derivatives $\partial f_i/\partial x_j$. To use NS23, he or she must call it repeatedly under the control of the parameter IFLAG. The initial call must be with IFLAG set to unity, an initial approximate solution in the array X, the corresponding function values in the array F, and the corresponding derivatives (if being provided) in the array DERIV. On a return with IFLAG > 0, code for placing in F the values of the functions at the point then held in X must be executed and the subroutine recalled. If derivatives are being provided, they must be set each time in the array DERIV. All other arguments must not be altered. A return with IFLAG=0 indicates successful termination and the error conditions are marked by IFLAG < 0.

The main entry, NS23AD, solves a system of nonlinear equations. Following a successful solution, the subsidiary entry NS23FD may be called to obtain a column of the inverse of the normal matrix

$$\mathbf{N} = \{\partial f_i/\partial x_j + a_{ij}\}^T \{\partial f_i/\partial x_j + a_{ij}\}$$

at the solution. If $m > n$ and the residuals r_i at the solution are independent random variables with variance σ^2, the variances and covariances of the components of the solution are given by the matrix $\sigma^2 \mathbf{N}^{-1}$. An estimate of σ^2 is $S/(m–n)$.

Before any calls are made to NS23AD and NS23FD the NS23ID entry must be called to initialize the control and workspace arrays.

2.1.1 Initialization entry

```
CALL NS23ID(ICNTL,CNTL,KEEP,RKEEP,LKEEP)
```

ICNTL is an INTEGER array of length 5 that need not be set by the user. On return it contains default values (see Section 2.2 for details).

CNTL is a DOUBLE PRECISION array of length 5 that need not be set by the user. On return it contains default values (see Section 2.2 for details).

KEEP is an INTEGER array of length 130 that need not be set by the user. It used by NS23 as private workspace and should normally not be altered by the user. KEEP(71:90) holds the array ICNTL of MA57. Non-default values for these entries may be set by an expert user after calling NS23ID.

RKEEP is a DOUBLE PRECISION array of length 72 that need not be set by the user. It used by NS23 as private workspace and should normally not be altered by the user. RKEEP(48:52) holds the array CNTL of MA57. Non-default values for these entries may be set by an expert user after calling NS23ID.

LKEEP is an LOGICAL array of length 10 that need not be set by the user. It used by NS23 as private workspace and must not be altered by the user.

2.1.2 Main entry – solve a system of nonlinear equations

```
 CALL NS23AD(M,N,SAC,STPMIN,MAXFUN,IPRINT,IRN,IP,A,IRNA,
*            IPA,HMAX,LIW,IW,LW,W,X,F,DERIV,IFLAG,
*            ICNTL,CNTL,INFO,RINFO,KEEP,RKEEP,LKEEP)
```

M is an INTEGER variable that must be set by the user to the number m of equations. It is not altered by the subroutine. **Restriction:** $M \geq 1$.

N is an INTEGER variable that must be set by the user to the number n of unknowns. It is not altered by the subroutine. **Restriction:** $M \geq N \geq 1$.

SAC is a DOUBLE PRECISION variable that must be set by the user to an acceptable value for the sum of squares S. Full details of the convergence tests are give in section 2.3. The user may set SAC to zero. It is not altered by the subroutine.

STPMIN is a DOUBLE PRECISION variable that must be set by the user to indicate the required accuracy for the solution. Full details of the convergence tests are given in section 2.3. The user may set STPMIN to zero. It is not altered by the subroutine.

MAXFUN is an INTEGER variable that must be set by the user to the maximum number of function evaluations allowed. An error return is made if the number is insufficient. It is not altered by the subroutine.

IPRINT is an INTEGER variable that must be set by the user to indicate the level of monitor printing required:–

- 0 No monitor printing.
- >0 Scalar information (see section 2.5) on the first and last iterations and every IPRINT iterations in between.
- <0 Full information on the first and last iterations and every |IPRINT| iterations in between. By default, printing is performed on unit 6, but this may be altered through the control variable ICNTL(1) (see section 2.2).

IPRINT is not altered by the subroutine.

IRN is an INTEGER array of size the number of nonzero derivatives $\{\partial f_i/\partial x_j\}$. It must be set by the user to hold the row indices of these nonzeros, stored by columns. It is not altered by the subroutine.

IP is an INTEGER array of size N+1 that must be set by the user so that IP(J) is the position in IRN of the start of column J, J=1,2,...,N, and IP(N+1) is the position of the first unused location in IRN. It is not altered by the subroutine.

A is a DOUBLE PRECISION array of size the number of nonzeros in the matrix $\{a_{ij}\}$ that must be set by the user to hold these nonzeros, stored by columns. It is not altered by the subroutine.

IRNA is an INTEGER array of size the number of nonzeros in the matrix $\{a_{ij}\}$ that must be set by the user to hold the row indices of these nonzeros, stored by columns. It is not altered by the subroutine.

IPA is an INTEGER array of size N+1 that must be set by the user so that IPA(J) is the position in IRNA of the start of column J, J=1,2,...,N, and IPA(N+1) is the position of the first unused location in IRNA. It is not altered by the subroutine.

HMAX is a DOUBLE PRECISION variable that must be set by the user to an upper limit for step lengths to be used when

differencing, or to zero if analytic derivatives are being provided. Normally, forward differences with step-length HMAX*$\sqrt{\varepsilon}$, where ε is the relative precision, are used. If a different step length is desired, it may be specified in CNTL(5), see Section 2.2. HMAX is not altered by the subroutine.

LIW is an INTEGER variable that must be set by the user to the size of array IW. What size is necessary depends on how much fill-in happens, but IW must be at least 44+M+11N+τ_j+4τ_n, where τ_j is the number of nonzeros in the Jacobian matrix $\mathbf{J} = \{\partial f_i/\partial x_j + a_{ij}\}$ and τ_n is the number of nonzeros in the normal matrix $\mathbf{J}^T\mathbf{J}+\lambda\mathbf{I}$. After a successful call, a minimum suitable value is returned in INFO(4) (see section 2.2). LIW is not altered by the subroutine.

IW is an INTEGER array of size LIW that is used as a work array. It must not be altered by the user after a return with IFLAG positive.

LW is an INTEGER variable that must be set by the user to the size of array W. What size is necessary depends on how much fill-in happens, but LW must be at least 4M+3N+1+τ_j+τ_n, where τ_j is the number of nonzeros in the Jacobian matrix $\mathbf{J} = \{\partial f_i/\partial x_j + a_{ij}\}$ and τ_n is the number of nonzeros in the normal matrix $\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}$. Unless LIW is too small, a minimum suitable value for LW is returned in INFO(5) (see section 2.2). LW is not altered by the subroutine.

W a DOUBLE PRECISION array of size LW that is used as a work array. It must not be altered by the user after a return with IFLAG positive.

X is a DOUBLE PRECISION array of size N that must be set initially by the user to an approximation to the solution. It is altered by the routine between intermediate calls and finally contains the best solution found.

F is a DOUBLE PRECISION array of size N whose elements must be set by the user initially and before every intermediate call to $f_i(x_1, x_2,..., x_n)$, i=1,2,...,m. It is not altered by the subroutine.

DERIV is a DOUBLE PRECISION array of size equal to the number of nonzeros in the matrix of derivatives $\{\partial f_i / \partial x_j\}$. If derivatives are being calculated analytically (HMAX=0), they must be stored by columns in DERIV in the same order as that of their indices in IRN; in this case, DERIV is not be altered by the subroutine. Otherwise, DERIV need not be set before the initial call and must not be altered between intermediate calls; the subroutine calculates approximate derivatives and places them in DERIV.

IFLAG is an INTEGER variable which must be set by the user to the value 1 on the first entry to the subroutine. On return, the value 0 indicates that the calculation is complete. Positive values indicate that the subroutine should be recalled after calculating $f_i(x_1, x_2,..., x_n)$, i=1,2,...,m and placing them in F, and calculating the derivatives and placing them in DERIV (if they are being provided), without changing other arguments. A negative value on return indicates one of the following error conditions:

- –1 More than MAXFUN function evaluations needed.
- –2 LIW is too small. INFO(4) (see section 2.2) is set to a size for LIW that will enable processing to proceed further.
- –3 LIW is too small, LW is too small, or both. INFO(4) and INFO(5) (see section 2.2) are set to the minimum suitable values for LIW and LW.
- –4 The restriction M ≥ N ≥ 1 is violated.

ICNTL is an INTEGER array of length 5 that contains control parameters. Default values for the elements are set by NS23ID. Details of the control parameters are given in Section 2.2. This argument is not altered by the subroutine.

CNTL is a DOUBLE PRECISION array of length 5 that contains control parameters. Default values for the elements are set by NS23ID. Details of the control parameters are given in Section 2.2. This argument is not altered by the subroutine.

INFO is an INTEGER array of length 10 that need not be set by the user. On return, it holds information on the calculation, see Section 2.2.

RINFO is an DOUBLE PRECISION array of length 5 that need not be set by the user. On return, it holds information on

the calculation, see Section 2.2.

KEEP is an INTEGER array of length 130 used by NS23 for private workspace. It must be initialized by calling NS23ID and and should normally not be accessed by the user. KEEP(91:130) holds the array INFO of MA57. Values for these entries may be accessed by an expert user.

RKEEP is a DOUBLE PRECISION array of length 72 used by NS23 for private workspace. It must be initialized by calling NS23ID and should normally not be accessed by the user. RKEEP(53:72) holds the array RINFO of MA57. Values for these entries may be accessed by an expert user.

LKEEP is an LOGICAL array of length 10 used by NS23 for private workspace. It must be initialized by calling NS23ID and must not be altered by the user.

2.1.3 Auxiliary entry – obtain a column of the inverse of the normal matrix

Warning: If analytic derivatives are not provided, the derivatives are approximated by differences and then updated at each step. This means that they may be rather inaccurate at the solution. We therefore advise the user to perform a fresh calculation with NS23AD, giving it the calculated solution as its starting point.

```
    CALL NS23FD(M,N,IRN,IP,A,IRNA,IPA,HMAX,LIW,IW,LW,W,
   *            ICOL,V,X,F,DERIV,IFLAG,
   *            ICNTL,INFO,RINFO,KEEP,RKEEP,LKEEP)
```

M,N,IRN,IP,A,IRNA,IPA,HMAX,LIW,IW,LW,W must have types and sizes exactly as for the call of NS23AD and must have values as on return from NS23AD. They are not altered by the subroutine.

ICOL is an INTEGER variable that must be set by the user to specify the required column of the variance-covariance matrix. It is not altered by the subroutine. **Restriction:** $1 \leq$ ICOL $\leq$ N.

V is a DOUBLE PRECISION array of size N in which is returned the required column of the variance-covariance matrix.

X is a DOUBLE PRECISION array of size N that must be as on return from NS23AD. It is not altered by the subroutine.

F is a DOUBLE PRECISION array of size M whose elements must be set by the user initially and before every intermediate call to $f_i(x_1, x_2, ..., x_n)$, i=1,2,...,m. It is not altered by the subroutine.

DERIV is a DOUBLE PRECISION array of size equal to the number of nonzeros in the matrix of derivatives $\{\partial f_i / \partial x_j\}$. If derivatives are being calculated analytically (HMAX=0), they must be stored by columns in DERIV in the same order as that of their indices in IRN; in this case, DERIV is not be altered by the subroutine. Otherwise, the subroutine calculates approximate derivatives and places them in DERIV.

IFLAG is an INTEGER variable which must be set by the user to the value 1 on the first entry to the subroutine. On return, the value 0 indicates that the calculation is complete. Positive values indicate that the subroutine should be recalled after calculating $f_i(x_1, x_2, ..., x_n)$, i=1,2,...,m and placing them in F, and calculating the derivatives and placing them in DERIV (if they are being provided), without changing other arguments. The value –5 indicates an error return because ICOL does not satisfy the restriction $1 \leq$ ICOL $\leq$ N.

ICNTL is the INTEGER array of length 5 that was passed to the NS23AD entry – it is not altered by the subroutine.

INFO is the INTEGER array of length 10 that was passed to the NS23AD entry, see Section 2.2.

RINFO is the DOUBLE PRECISION array of length 5 that was passed to the NS23AD entry, see Section 2.2.

KEEP is the INTEGER array of length 130 that was passed to the NS23AD entry and must not be altered by the user.

RKEEP is the DOUBLE PRECISION array of length 72 that was passed to the NS23AD entry and must not be altered by the user.

LKEEP is the LOGICAL array of length 10 that was passed to the NS23AD entry and must not be altered by the user.

2.2 The control and information arrays

The arrays ICNTL and CNTL can be used to control the action of the subroutines. Default values are set by NS23ID.

ICNTL(1) specifies the unit number to be used to output error messages (see sections 2.4 and 2.5). A zero or negative value will suppress output. It has a default value of 6.

ICNTL(2) has default value 0 and controls the test for termination (see section 2.3).

ICNTL(3) to ICNTL(5) are not used at present and should not be altered.

CNTL(1) has default value 0.25 and is set to the algorithm parameter FAIM (see section 4).

CNTL(2) has default value 0.25 and is set to the algorithm parameter RHO (see section 4).

CNTL(3) has default value 0.75 and is set to the algorithm parameter SIG (see section 4).

CNTL(4) has default value 0. If f is not calculated with full precision, the relative accuracy should be set in CNTL(4). A negative value is taken as zero.

CNTL(5) has default value 0. If it is set to a positive value, the step length CNTL(5) is normally used for forward differencing. A negative value is taken as zero.

The INFO and RINFO arrays are used to return information back to the user.

INFO(1) is used as an error indicator. It is set to zero when there are no errors otherwise it is set to the negative value returned in IFLAG (see Section 2.4).

INFO(2) is used to return a supplementary value associated with the error indicated by INFO(1) (see Section 2.4).

INFO(3) is set on to the number of function evaluations.

INFO(4) is set to the minimum size for the array IW or to a size that will enable processing to proceed further.

INFO(5) is set to the minimum size for the array W.

INFO(6) to INFO(10) are not used at present.

RINFO(1) is set to the current value of the Marquardt parameter (see section 4).

RINFO(2) to RINFO(5) are not used at present.

The dependent subroutines in TD22 and MA57 also use control and information arrays and NS23 holds these in its KEEP and RKEEP work arrays at the offsets:

```
TD22's  ICNTL(J), J = 1,5     are in KEEP(61) to KEEP(65)
TD22's  INFO(J),  J = 1,5     are in KEEP(66) to KEEP(70)
TD22's  CNTL(J),  J = 1,7     are in RKEEP(41) to RKEEP(47)

MA57's  ICNTL(J), J = 1,20    are in KEEP(71) to KEEP(90)
MA57's  INFO(J),  J = 1,40    are in KEEP(91) to KEEP(130)
MA57's  CNTL(J),  J = 1,5     are in RKEEP(48) to RKEEP(52)
MA57's  RINFO(J), J = 1,20    are in RKEEP(53) to RKEEP(72)
```

2.3 Convergence tests

Normally, termination occurs if the sum of squares of residuals S is less than SAC or if the norm of the change to the solution is less than STPMIN. It also occurs if the step length becomes less than ε times the norm of the solution vector $\mathbf{x}$, where ε is the relative precision. If ICNTL(2) (see section 2.2) is changed from its default value of zero, the first reason for termination is that both the sum of squares of residuals S is less than SAC and the norm of the change to the solution is less than STPMIN.

2.4 Error Messages

There are five error returns, indicated by the following values of IFLAG (and INFO(1)):

–1 More than MAXFUN function evaluations needed. INFO(2) holds the value of MAXFUN.
–2 LIW is too small. INFO(2) holds the value of LIW.
–3 LW is too small. INFO(2) holds the value of LW.
–4 The restriction M ≥ N ≥ 1 is violated. INFO(2) is set to 0.
–5 ICOL does not satisfy the restriction 1 ≤ ICOL ≤ N.

If ICNTL(1) is positive, a message is output on stream ICNTL(1).

2.5 Monitor printing

If scalar monitor printing is requested (IPRINT > 0), the iteration number, the Marquardt parameter λ, the sum of squares of residuals S, the number of function evaluations, the norm of the current approximate solution, the norm of the last change made to it, and the norm of the vector $\mathbf{v} = \{\partial r_i/\partial x_j\}^T\mathbf{r}$ (which is the gradient of $S/2$) are printed.

If vector monitor printing is requested (IPRINT < 0), the current solution, current residual vector **r**, and current gradient vector **v** are also printed.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly: DDOT, MA57AD, MA57BD, MA57CD, NS23CD, NS23DD, NS23ED, NS23HD, TD22AD.

Input/output: Diagnostic messages on unit ICNTL(1) (see §2.2).

Restrictions: M ≥ N ≥ 1, 1 ≤ ICOL ≤ N.

4 METHOD

A typical step of Marquardt's algorithm involves solving the linear least-squares problem

$$\begin{pmatrix} \mathbf{J} \\ \lambda^{1/2}\mathbf{I} \end{pmatrix} \delta = \begin{pmatrix} -\mathbf{r} \\ \mathbf{0} \end{pmatrix}$$

where **J** is the Jacobian matrix $\{\partial f_i/\partial x_j + a_{ij}\}$, **r** is the residual vector $\{r_i\}$, λ is a scalar known as the Marquardt parameter, and δ gives a correction to the current solution. This problem is solved by the formation and solution of the normal equation

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\,\delta = -\mathbf{J}^T\mathbf{r}.$$

If λ is large, δ is approximately $-\lambda^{-1}\mathbf{J}^T\mathbf{r}$, which is an incremental step along the direction of steepest descent of the sum of squares S. On the other hand, if $\lambda = 0$, the correction is that of the Gauss-Newton method where the nonlinear problem is approximated by its linearization

$$\mathbf{r}(\mathbf{x}+\delta) = \mathbf{r} + \mathbf{J}\,\delta.$$

The method aims to adjust λ so that the length of the step is always such that the linearization is a satisfactory approximation.

The subroutine uses an adaptation of Fletcher's version of the algorithm (R. Fletcher, 'A modified Marquardt subroutine for nonlinear least squares', Harwell Report R-6799, 1971) and is described in detail by Reid (J. K. Reid, 'Fortran subroutines for the solution of sparse systems of nonlinear equations, 1972). The code was originally called NS03AD. It has been modified for better readability, conformance with Fortran 77, reverse communication and for threadsafe execution.

The parameters RHO and SIG indicate a range $[\rho, \sigma]$ within which the ratio R of the actual to expected reduction in the sum of squares S is regarded as acceptable. If derivatives are not provided analytically, they are approximated

initially by using subroutine TD22 and are revised by an updating formula as the iteration progresses. TD22 is called again later if $R<\rho$ although the length of the last change to **x** is less than FAIM* d_{aim}, where d_{aim} is a length calculated by the algorithm. The value of d_{aim} is intended to be sufficiently small for there to be good agreement between the expected and actual reduction in the sum of squares S whenever the length of the change to **x** is less than d_{aim}. Forward differences with steps HMAX*$\sqrt{\varepsilon}$, where ε is the relative precision, are used unless the improvement to S immediately following a call of TD22 is poor, in which case TD22 is recalled to use central differences with automatic step adjustment.

5 EXAMPLE OF USE

We consider the solution of the following set of nonlinear equations

$$r_i(\mathbf{x}) = x_i + 0.5x_{i+1}^2 - 1.5 = 0, \quad i=1,2,...,n-1, \qquad r_n(\mathbf{x}) = x_n - 1.5 = 0.$$

We take

$$f_i(\mathbf{x}) = 0.5x_{i+1}^2 - 1.5, \quad i=1,2,...,n-1, \qquad f_n(\mathbf{x}) = -1.5$$

and $\{a_{ij}\}$ equal to the identity matrix **I**. Using the initial solution $\mathbf{x}=\mathbf{0}$, the following code suffices for the case $n=10$.

```
      EXTERNAL NS23AD, FUNC
      INTEGER M, N, LIW, LW
      PARAMETER (M=10, N=10, LIW=500, LW=500)
      DOUBLE PRECISION X(N), F(N), DERIV(N-1), SAC, STPMIN, A(N), HMAX,
     *        W(LW)
      INTEGER MAXFUN, IPRINT, IRN(N), IP(N+1), IRNA(N), IPA(N+1),
     *        IW(LIW), IFLAG, I
      PARAMETER (SAC=1D-10, STPMIN=1D-10, MAXFUN=30, IPRINT=0)
      INTEGER ICNTL(5),INFO(10),KEEP(130)
      DOUBLE PRECISION CNTL(5),RINFO(5),RKEEP(72)
      LOGICAL LKEEP(10)
      CALL NS23ID(ICNTL,CNTL,KEEP,RKEEP,LKEEP)
      WRITE(6,'(A,I3,A,1P,E11.4,A,E11.4)')
     *     ' N =',N, ',  SAC =',SAC, ',  STPMIN =',STPMIN
C  Set initial solution
      DO 10 I=1,N
         X(I)=0.0D0
   10 CONTINUE
C  Set pattern of the nonzero derivatives of f
      IP(1) = 1
      DO 20 I=1,N-1
         IRN(I) = I
         IP(I+1) = I
   20 CONTINUE
      IP(N+1) = N
C  Set matrix A
      DO 30 I=1,N
         A(I)=1.0D0
         IRNA(I) = I
         IPA(I) = I
   30 CONTINUE
      IPA(N+1) = N + 1
      IFLAG = 1
      HMAX = 0.0D0
   40 CONTINUE
         CALL FUNC( N, X, F, DERIV)
         CALL NS23AD( M, N, SAC, STPMIN, MAXFUN, IPRINT, IRN, IP,
     *        A, IRNA, IPA, HMAX, LIW, IW, LW, W, X, F, DERIV, IFLAG,
     *        ICNTL,CNTL,INFO,RINFO,KEEP,RKEEP,LKEEP)
      IF(IFLAG.GT.0) GO TO 40
      IF (IFLAG.EQ.0) THEN
          WRITE(6,'(A,/,(5F9.5))') ' Solution is',(X(I),I=1,N)
      ELSE
          WRITE(6,'(A,I4)') ' Failure. INFO(1:5) =',(INFO(I),I=1,5)
      END IF
      END

      SUBROUTINE FUNC ( N, X, F, DERIV)
      INTEGER N
      DOUBLE PRECISION X(N), F(N), DERIV(N-1)
      INTEGER I
      DO 10 I = 1, N-1
         F(I) = 0.5D0*X(I+1)**2 - 1.5D0
         DERIV(I) = X(I+1)
   10 CONTINUE
      F(N) = -1.5D0
      END
```

This produces the following output

```
 N = 10,  SAC = 1.0000E-10,  STPMIN = 1.0000E-10
 Solution is
```

```
   0.61544  1.33008  0.58295  1.35429  0.53984
   1.38576  0.47800  1.42969  0.37500  1.50000
```