1 SUMMARY

To find $\mathbf{X}=x_1,x_2,...,x_n$ that **minimizes a quadratic function** of the form

$$Q(\mathbf{x})=\tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}-\mathbf{b}^T\mathbf{x}$$

where $\mathbf{A}=\{a_{ij}\}_{n\times n}$ is symmetric matrix, $\mathbf{b}$ is a vector and $\mathbf{x}$ is subject to **bounds** $l_i \le x_i \le u_i$ i=1, 2,..., n.

The subroutine calculates the solution $\mathbf{x}=\xi$, the minimum value $Q(\xi)$, and the gradient $\mathbf{g}(\xi)$ (note $\mathbf{g}(\mathbf{x})=\mathbf{A}\mathbf{x}-\mathbf{b}$). This problem is a special case of quadratic programming for which the subroutine `VE02` exists. `VE04` is more efficient and reliable for solving problems of this special form.

The subroutine can be used to solve linear least squares data fitting problems in cases when the variables are subject to bounds. For this application there is an extra entry point which provides the variance covariance matrix for the fit.

The method is that of R.Fletcher and M.P.Jackson (1973), AERE TP.528.

ATTRIBUTES — **Version:** 1.0.0. **Types:** `VE04A; VE04AD.` **Calls:** `FD05`. **Original date:** April 1973. **Origin:** R.Fletcher, Harwell.

2 HOW TO USE THE PACKAGE

2.1 An application

An application of the subroutine is to *weighted* linear least squares data fitting subject to bounds. If it is required to minimize

$$S(\mathbf{x})=(\mathbf{B}\mathbf{x}-\mathbf{y})^T\mathbf{W}(\mathbf{B}\mathbf{x}-\mathbf{y})$$

subject to the above bounds, where $\mathbf{W}$ is an $m\times m$ diagonal matrix of weights ($w_{i,i}$>0, then set $\mathbf{A}=2\mathbf{B}^T\mathbf{W}\mathbf{D}$ and $\mathbf{b}=2\mathbf{B}^T\mathbf{W}\mathbf{y}$. Statistical calculations for this problem are described in Section 4, including an additional entry point to enable the variance-covariance matrix to be calculated.

2.2 Argument List

The single precision version

```
CALL VE04A(N,A,IA,B,BL,BU,X,Q,LT,K,G)
```

The double precision version

```
CALL VE04AD(N,A,IA,B,BL,BU,X,Q,LT,K,G)
```

`N` is an `INTEGER` variable which must be set by the user to n the number of variables.

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) two dimensional array, each dimension at least n; the elements in the upper triangle `A(I,J)`, `I` ≤ `J` ≤ `N`, must be set by the user to the corresponding a_{ij} in $\mathbf{A}$, and will remain untouched by the subroutine. Elements `A(I,J)`, `N` ≥ `I` ≥ `J`, are used as working space.

`IA` is an `INTEGER` variable giving the first dimension of `A` in the statement which assigns space to `A`.

`B` is a `REAL` (`DOUBLE PRECISION` in the D version) array of at least n elements. The user must set `B(i)` to the b_i. `B` is not overwritten by the subroutine.

`BL` is a `REAL` (`DOUBLE PRECISION` in the D version) array of at least n elements. The user must set BL(i) to the lower bound l_i on the i-th variable. If the bound is non-existent, set l_i to a very small number like –|large| (large is a close approximation to the largest finite negative number which may be stored in the machine). `BL` is not overwritten by VE04A/AD.

BU is a REAL (DOUBLE PRECISION in the D version) array of at least N elements. The user must set BU(i) to the upper bound u_i on the i-th variable. If the bound is non-existent, set u_i to a very large number (see large in argument BL). BU is not overwritten by VE04A/AD.

X is a REAL (DOUBLE PRECISION in the D version) array of at least n elements. VE04A/AD returns the solution ξ_i in X(i).

Q is a REAL (DOUBLE PRECISION in the D version) variable in which VE04 returns the value of $Q(\xi)$.

LT is an INTEGER array of at least n elements, set by VE04 to a permutation of the integers 1, 2,..., n (see K and G below).

K is an INTEGER variable set by VE04A/AD to the number of free variables at the solution ξ (those not on their bounds). These are the variables LT(1), LT(2), ..., LT(K).

G is a REAL (DOUBLE PRECISION in the D version) array of at least $3n$ elements, G(1), G(2), ..., G(N) are set by VE04A/AD to the gradient $g(\xi)$. G is indirectly addressed so that G(I) contains the gradient with respect to the LT(I) variable, whence G(1), G(2), ..., G(K) will be found to be zero. G(N+1), G(N+2), ..., G(3*N) are used by VE04A/AD as working space.

3 GENERAL INFORMATION

Use of common: None

Workspace: Approximately $\frac{1}{2}n^2$ words (half of **A**) + $2n$ words in G and n integers in LT.

Other routines called directly: FD05.

Input/output: None

Restrictions: None

4 METHOD

That of Fletcher, R. and Jackson, M.P. (1973) ‘Minimization of a quadratic function of many variables subject only to lower and upper bounds’, TP 528. This method combines generality (any **A**), efficiency (times comparable to those required for a factorization of **A**) and stability (uses partial $\mathbf{LDL}^T$ factorizations).

Statistical Calculations

When a sum of squares is being minimized as in §2.1, the certain statistical quantities can readily be calculated. Firstly, of course, $S(\xi)$ is given by $Q(\xi)+\mathbf{y}^T\mathbf{Wy}$. If it is assumed that the bound variables located at VE04A/AD are exact in the underlying model, then an estimate of the residual variance is given by $S(\xi)/(m-k)$. To estimate variances and covariances, an additional entry point is provided. This calculates $\{\mathbf{A}\}$ where $\{\mathbf{A}\}$ indicates the submatrix $\{A_{ij}\}$ where i and j index only the free variables. The appropriate variance-covariance matrix for the free variables is then $\sigma^2\{\mathbf{A}\}^{-1}$. Estimates of standard deviations of the free variables are given by the square roots of the diagonal element s of this matrix. Because the bound variables are known exactly, they have zero variance and covariance.

The entry point VE04B/BD is essentially written as a separate subroutine. It calculates $\{\mathbf{A}\}^{-1}$ and is used as follows:

The single precision version

```
CALL VE04B(N,A,IA,G,K)
```

The double precision version

```
CALL VE04BD(N,A,IA,G,K)
```

N,A,IA,G and K must be passed on unchanged after exit from VE04A/AD. VE04B/BD, sets the following:

`A` The off diagonal elements of $\{\mathbf{A}\}^{-1}$ are set in `A(I,J)` for `J` < `I` ≤ `K`. The elements are indirectly addressed so that `A(I,J)` contains $\{\mathbf{A}\}^{-1}_{r,s}$ where r=`LT(I)` and s=`LT(J)`.

`G` The diagonal elements of $\{\mathbf{A}\}^{-1}$ are set in `G(N+1), G(N+2),..., G(N+K)`. They are again indirectly addressed so that `G(N+1)` contains $\{\mathbf{A}\}^{-1}_{r,r}$ where r=`LT(I)`.

Timing

The time required depends upon how many free variables k there are at the solution. Typical figures for (k/n number of multiplications) are (.1,n^3/12), (.5,n^3/6), (.75,n^3/2).