1 SUMMARY

This subroutine **solves a convex quadratic programming problem.** It seeks to minimize the quadratic function

$$F(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{G}\,\mathbf{x} + \mathbf{x}^T\mathbf{g}$$
$$= \tfrac{1}{2}\sum_{ij} g_{ij}x_i x_j + \sum_i x_i g_i$$

subject to the m linear constraints

$$\sum_{j=1}^{n} a_{jk}x_j = b_k, \qquad k=1,2,...,m_e$$

$$\sum_{j=1}^{n} a_{jk}x_j \geq b_k, \qquad k=m_e+1,...,m,$$

and the simple bound constraints

$$l_i \leq x_i \leq u_i, \qquad i=1,2,...,n,$$

where $\mathbf{G}$ should be positive definite ($\mathbf{x}^T\mathbf{G}\mathbf{x} \geq 0$ for any vector $\mathbf{x}$). If it is not, $\mathbf{G}$ is replaced by $\mathbf{G}+\delta\mathbf{I}$ with δ suitably chosen. Any of the bounds l_i and/or u_i may be infinite. If no simple bounds are present, the alternative subroutine VE07A/AD should be used instead of VE17A/AD.

ATTRIBUTES — **Version:** 1.0.0. **Remark:** This subroutine is intended to complement VE07A/AD. **Types:** VE17A, VE17AD. **Original date:** April 1987. **Origin:** M.J.D.Powell, Cambridge. **Licence:** A third-party licence for this package is available without charge.

2 HOW TO USE THE PACKAGE

2.1 The Argument List and Calling Sequence

The single precision version

```
      CALL VE17A(N,M,MEQ,BDL,BDU,A,IDIMA,B,GRAD,G,IDIMG,
     *           X,NACT,IACT,INFO,DIAG,W)
```

The double precision version

```
      CALL VE17AD(N,M,MEQ,BDL,BDU,A,IDIMA,B,GRAD,G,IDIMG,
     *            X,NACT,IACT,INFO,DIAG,W)
```

N is an INTEGER variable which must be set by the user to the number n of variables. This argument is not altered by the subroutine. **Restriction:** $n > 0$.

M is an INTEGER variable which must be set by the user to the number m of constraints. This argument is not altered by the subroutine. **Restriction:** $m \geq 0$.

MEQ is an INTEGER variable which must be set by the user to the number m_e of equality constraints. This argument is not altered by the subroutine. **Restriction:** $m_e \geq 0$.

BDL is a REAL (DOUBLE PRECISION in the D version) array of length n that the user must set to the vector of lower bounds on the variables, l. If there is no lower bound on the k-th variable, BDL(K) should be set to some large negative number. It is not altered by the subroutine.

BDU is a REAL (DOUBLE PRECISION in the D version) array of length n that the user must set to the vector of upper

bounds on the variables, *u*. If there is no upper bound on the *k*-th variable, BDU(K) should be set to some large positive number. It is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version) array whose first dimension is IDIMA, which the user must set to hold the constraint matrix $\mathbf{A}=(a_{jk})$. This argument is not altered by the subroutine.

IDIMA is an INTEGER variable that the user must set to the first dimension of array A. It is not altered by the subroutine. **Restriction:** IDIMA $\geq n$.

B is a REAL (DOUBLE PRECISION in the D version) array of length *m* that the user must set to the vector of right-hand sides of the constraints. It is not altered by the subroutine.

GRAD is a REAL (DOUBLE PRECISION in the D version) array of length *n* that the user must set to the vector **g** (equal to the gradient of *F* at **x**=**0**). It is not altered by the subroutine.

G is a REAL (DOUBLE PRECISION in the D version) array whose first dimension is IDIMG, which the user must set to hold the second derivative matrix $\mathbf{G}=(g_{ij})$. It must be symmetric (i.e. G(I,J)=G(J,I), I=1,N, J=1,N). It is not altered by the subroutine.

IDIMG is an INTEGER variable that the user must set to the first dimension of array G. It is not altered by the subroutine. **Restriction:** IDIMG $\geq n$.

X is a REAL (DOUBLE PRECISION in the D version) array of length *n* that the user need not set. On return it contains the final vector of variables.

NACT is an INTEGER variable that the user need not set. On return it contains the number of constraints active at the solution (i.e. treated as equalities).

IACT is an INTEGER array of length *n* that need not be set by the user. On return IACT(K), K=1,NACT, hold the indices of the constraints that are active and the variables that are on one of their bounds at the solution. If $1 \leq$ IACT(K) $\leq m$, the IACT(K)-th linear constraint is active, if $m+1 \leq$ IACT(K) $\leq m+n$, the (IACT(K)$-m$)-th variable is on its lower bound and if $m+n+1 \leq$ IACT(K) $\leq m+2n$, the (IACT(K)$-m-n$)-th variable is on its upper bound.

INFO is an INTEGER variable that need not be set by the user. On return it has one of the following values:

(i) INFO > 0. The calculation has been successful (though an altered problem will have been solved if DIAG $\neq 0$) and the value of INFO indicates the number of iterations performed.

(ii) INFO = 0. The machine accuracy is insufficient. There may be some avoidable constraint violations, though **x** will be optimal for the perturbed problem obtained by changing the corresponding values of b_i to avoid the violations.

(iii) INFO < 0. The constraint with index IABS(INFO) and the constraints whose indices are IACT(K), K=1,NACT are inconsistent.

DIAG is a REAL variable that need not be set by the user. On return it contains the multiple δ of the unit matrix that was added to G to give a positive definite-matrix, so normally its final value is zero.

W is a REAL array that need not be set by the user and is used by the subroutine for workspace. Its length must not be less than $\frac{3}{2}n^2+\frac{15}{2}n+m$. On return with INFO > 0, the Lagrange multipliers of the final active constraints are held in W(K), K=1,NACT.

3 GENERAL INFORMATION

Workspace: provided by the user, see argument W.

Use of common: none.

Other routines called directly: FD05.

Input/output: none.

Restrictions:

$n>0$,
$m\geq 0$,
$m_e\geq 0$,
IDIMA$\geq n$,
G symmetric,
IDIMG$\geq n$.

4 METHOD

The algorithm is that of Goldfarb and Idnani (Mathematical Programming **27**, (1983), 1-33) and the implementation is due to Powell ('ZQPCVX A Fortran subroutine for convex quadratic programming', report DAMTP/1983/NA17, Dept of Applied Maths and Theoretical Physics, University of Cambridge and 'On the quadratic programming algorithm of Goldfarb and Idnani', Mathematical Programming Study **27**, (1985), 46-61). The method is essentially the same as that implemented in VE07A/AD but allows the user to specify simple bound constraints separately from the general linear constraints, thereby enabling savings to be made in storage. It begins by performing a Choleski decomposition of **G** (using this to check whether **G** is positive definite and if necessary to choose δ) and uses this to solve the unconstrained problem. This is the first of a sequence of quadratic programs that it solves with objective function **F** and constraints which are subsets $S^{(i)}$, i=1,2,... of the given set of constraints. $S^{(1)}$ is the empty set and later sets are chosen so that the sequence of function values $F(\mathbf{x}^{(i)})$ increases monotonically until the required problem is solved or it is discovered that the constraints are inconsistent.

5 EXAMPLE OF USE

Calculate the least value of the function of three variables

$$F(x)=\frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{x_i x_j}{i+j}-\tfrac{1}{5}x_1$$

subject to the constraints

$$x_1+x_2+x_3=1$$
$$x_i\geq 0,\quad i=1,2,3.$$

The following main program is suitable. It prints out the number of iterations, the final solution and the indices of the constraints that are active at the solution. The exact solution is (0.7,0,0.3).

```
      REAL BDL(3),BDU(3),A(3,1), B(1), GRAD(3), G(3,3), X(3), W(37)
      INTEGER IACT(3)
      N = 3
      M = 1
      MEQ = 1
      IDIMA = 3
      IDIMG = 3
      DO 20 I=1,N
        GRAD(I) = 0.0
        DO 10 J=1,N
          G(I,J) = 1.0/FLOAT(I+J)
   10   CONTINUE
        A(I,1) = 1.0
        BDL(I) = 0.0
        BDU(I) = 1.0E+20
```

```
20 CONTINUE
   GRAD(1) = -0.2
   B(1) = 1.0
   CALL VE17A(N, M, MEQ, BDL, BDU, A, IDIMA, B, GRAD, G, IDIMG,
  *           X, NACT, IACT, INFO, DIAG, W)
   WRITE (6,30) INFO
30 FORMAT (/5X, 21HNUMBER OF ITERATIONS=, I3)
   WRITE (6,40) (X(I),I=1,N)
40 FORMAT (/5X, 26HFINAL VECTOR OF VARIABLES=/1P, 3E14.6)
   WRITE (6,50) (IACT(I),I=1,NACT)
50 FORMAT (/5X, 36HTHE FOLLOWING CONSTRAINTS ARE ACTIVE, 4I3)
   STOP
   END
```

This produces the following output

```
 NUMBER OF ITERATIONS=  4

 FINAL VECTOR OF VARIABLES=
6.999997E-01  0.000000E+00 3.000001E-01

 THE FOLLOWING CONSTRAINTS ARE ACTIVE  3  1
```