1 SUMMARY

To find the minimum of a function $F(\mathbf{x})$ of many variables $(x_1,x_2,...,x_n)$, given that the gradient vector $(\partial F/\partial x_1,\partial F/\partial x_2,...,\partial F/\partial x_n)$ can be calculated. The subroutine is intended for problems that are so large that an $n\times n$ matrix cannot be stored conveniently. The use of VA13 is recommended for smaller problems.

ATTRIBUTES — **Version:** 1.0.0. **Types:** Real (single, double). **Original date:** July 2001. **Remark:** VA34 is a threadsafe version of VA14. **Origin:** M.J.D. Powell, Harwell. **Licence:** A third-party licence for this package is available without charge.

2 HOW TO USE THE PACKAGE

2.1 Initialization

The VE34I/ID entry may be called to initialize the control array ICNTL to its default values.

The single precision version

```
CALL VA34I(ICNTL)
```

The double precision version

```
CALL VA34ID(ICNTL)
```

ICNTL is the INTEGER array of length 5 which will be used to pass optional user controls to the VA34A/AD entry.

2.2 The argument list for the calculation entry

The single precision version

```
CALL VA34A(FUNCT,N,X,F,G,DFN,ACC,W,ICNTL,INFO)
```

The double precision version

```
CALL VA34AD(FUNCT,N,X,F,G,DFN,ACC,W,ICNTL,INFO)
```

FUNCT is the name of the subroutine provided by the user whose purpose is defined in Section 2.3. This name must be declared in an EXTERNAL statement.

N ia an INTEGER whose value must be set to the number of variables, which is called n in Section 1.

X is a REAL (DOUBLE PRECISION in the D version) array of at least n elements, set by the user to initial values of the variables $(x_1,x_2,...,x_n)$. Usually computing time is saved if these estimates are close to the final solution. They are changed automatically to the values that give the least calculated value of the objective function.

F is a REAL (DOUBLE PRECISION in the D version) variable that is set automatically to the least calculated value of the objective function.

G is a REAL (DOUBLE PRECISION in the D version) array of at least n elements that are set automatically to the components of the first derivative vector of $F(\mathbf{x})$ for the final values of the variables. Small values indicate a successful calculation.

DFN is a REAL (DOUBLE PRECISION in the D version) variable that must be set to a positive value on entry to the subroutine. The value should be the expected reduction in $F(\mathbf{x})$ on the first iteration and a suitable value is often one tenth of the total expected reduction in $F(\mathbf{x})$. Any value that is not in error by a factor of about one thousand will usually suffice.

ACC is a REAL (DOUBLE PRECISION in the D version) set to a number that controls the normal exit from the subroutine. This exit occurs when a gradient vector `G(.)` is found such that the sum of squares of its components is less than `ACC.`

`W` is a `REAL` (`DOUBLE PRECISION` in the D version) array of at least 6*n* elements that is used as working space.

`ICNTL` is an `INTEGER` array of length 5 which is used to pass optional control information.

`ICNTL(1)` (default value 6) may be set to the stream number to be used for printed output from `VA34A`.

`ICNTL(2)` may be set to a non-zero value to instruct `VA34A` to produce printing every `ABS(ICNTL(2))` iterations. To restrict the output to just the iteration number and function value set `ICNTL(2)` < 0, or set `ICNTL(2)` > 0 to also include the elements of `X(.)` and `G(.).` When `ICNTL(2)` set to its default value of zero none of this printing is done.

`ICNTL(3)` may be set to the maximum number of function evaluations allowed. Its value must be positive and `VA34A` finishes as soon as the user subroutine has been called `ICNTL(3)` times. The default value is zero, in which case it does not restrict the calculation.

`ICNTL(4)` and `ICNTL(5)` are not used.

`INFO` is an `INTEGER` array of length 5 in which the subroutine returns information.

`INFO(1)` is set at the end of the calculation to inform the user of the success or otherwise of the calculation. The possible return values are:

0 the calculation has been successful,

1 the calculation was terminated before completion because `ICNTL(3)` function values have been calculated,

2 indicates an error return because it seems that the calculation can make no further progress.

When `INFO(1)` > 0 a suitable diagnostic message is printed on the output stream `ICNTL(1)` ignoring the value of `ICNTL(2)`.

`INFO(2)` to `INFO(5)` are not used.

2.3 The user supplied subroutine

The user must supply a subroutine

The single precision version

```
SUBROUTINE FUNCT(N,X,F,G)
REAL X(N),G(N)
```

The double precision version

```
SUBROUTINE FUNCT(N,X,F,G)
DOUBLE PRECISION X(N),G(N)
```

The name of this subroutine must be the first argument of the statement that calls `VA34A/AD.`

`N,` `X`, `F` and `G` correspond to the arguments with these names in Section 2.2. `N` and the elements of `X` are given to the user subroutine by `VA34A/AD.` The user subroutine must set `F` and `G(i),i=1,2,...,n,` to the value of the objective function and the components of its first derivative for the given vector of variables `X(.).` The components of `X(.)` must not be changed. The user should check the accuracy of his derivatives independently, because if they are wrong then the convergence of `VA34A/AD` becomes much slower than necessary, or `VA34A/AD` may fail to find a minimum of the objective function.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly: None.

Input/output: Information and error messages on unit `ICNTL(1)` (see also `ICNTL(2)` and `INFO(1)`).

4 METHOD

The method is described in AERE Report CSS.24 by M.J.D. Powell (November, 1975). It is a conjugate gradient method that includes an automatic restart procedure. Thus this routine is sometimes much faster than `VA08A.` Numerical calculations suggest that usually `VA34` is preferable to `VA08,` but note that `VA08` requires less working space.

5 EXAMPLE OF USE

We consider the test problem

$$\text{minimize } 100(x_1^2-x_2)^2+(x_1-1)^2.$$

The solution to this problem is at the point (1,1) and the optimal function value is 0. We may solve this problem using the following code:

```
      INTEGER ICNTL(5),INFO(5)
      DOUBLE PRECISION F
      DOUBLE PRECISION G(2),X(2),W(12)
      EXTERNAL FUNCT,VA34AD
      DATA X/-1.2D0,1.D0/
      CALL VA34ID(ICNTL)
      ICNTL(3) = 200
      CALL VA34AD(FUNCT,2,X,F,G,1.0D-1,1.0D-6,W,ICNTL,INFO)
      WRITE( 6, 2000 ) F, X(1), X(2)
      STOP
 2000 FORMAT( ' Final function value = ', 1P, D12.4, /,
     *        ' Solution components  = ', 2D12.4 )
      END
```

where the required external function evaluation subroutine is:

```
      SUBROUTINE FUNCT(N,X,F,G)
      DOUBLE PRECISION F
      INTEGER N
      DOUBLE PRECISION G(*),X(*)
      DOUBLE PRECISION R,S
      R = 10.D0* (X(1)**2-X(N))
      S = X(1) - 1.D0
      F = R**2 + S**2
      G(1) = 40.D0*X(1)*R + 2.D0*S
      G(N) = -20.D0*R
      RETURN
      END
```

The output is as follows:

```
 Final function value =   3.0420D-10
 Solution components  =   9.9999D-01  9.9999D-01
```