1 SUMMARY

To find the **minimum of a general function** $F(\mathbf{x})$ of n variables $\mathbf{x} = x_1, x_2, ..., x_n$. Values of the **derivatives are not required.**

A variation of the simple method of changing one variable at a time is used. The method is such that when applied to a quadratic form, it causes conjugate directions of search to be chosen, so when applied to a general function the ultimate rate of convergence is fast, see, M.J.D. Powell, Computer Journal, Vol. 7, No. 2, 1964.

The method is iterative and requires the user to give an initial estimate of the minimum position. The required accuracies in the solution must be specified and a subroutine to calculate values of $F(\mathbf{x})$ provided.

ATTRIBUTES — **Version:** 1.0.0. **Types:** `VA24A`; `VA24AD`. **Original date:** February 1964. **Origin:** M.J.D.Powell, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument lists

The single precision version

        CALL VA24A(X,E,N,F,ESCALE,IPRINT,ICONV,MAXIT,W,CALCFX)

The double precision version

        CALL VA24AD(X,E,N,F,ESCALE,IPRINT,ICONV,MAXIT,W,CALCFX)

X   is a REAL (DOUBLE PRECISION in the D version) array of length at least n which must be set by the user to contain the initial approximation to the n variables $x_1, x_2, ..., x_n$. On return X(i) will contain the value of x_i at the minimum position to the required accuracy.

E   is a REAL (DOUBLE PRECISION in the D version) array of length at least n which must be set by the user to contain the absolute accuracies required in the final values of the variables. The accuracy required in x_i should be set in E(i). It is assumed that the accuracies specified are roughly proportional to the final x_is.

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

F   is a REAL (DOUBLE PRECISION in the D version) variable set by the subroutine to the value of $F(\mathbf{x})$ at the minimum position.

ESCALE is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to limit the length of step taken by the subroutine at each iteration. X(I) will be changed by not more than ESCALE*E(I). This parameter is used to prevent the subroutine moving away from the required minimum and possibly converging to a neighbouring one.

IPRINT is an INTEGER variable which is set by the user to control the printed output from the subroutine. The possible values are:

       0   no printing

       1   the values of the variables and function are printed at every search along a line — approximately every other function value.

       2   the values of the variables and function are printed at every iteration, i.e. every $n+1$ searches along a line.

ICONV   is an INTEGER variable which is set by the user to 1 or 2 to control the ultimate convergence criterion, see

§2.3. Normally `ICONV` will be set to 1, but set `ICONV` to 2 if a more thorough check on the ultimate convergence is required (at the expense of computation time).

`MAXIT` is an `INTEGER` variable which must be set by the user to limit the number of iterations. The subroutine will return to the calling program regardless after `MAXIT` iterations.

`W` is a `REAL` (`DOUBLE PRECISION` in the D version) array used by the routine as workspace. It must be of length at least $n(n+3)$.

`CALCFX` is the name of the user-supplied subroutine described in §2.2. It should be declared in an `EXTERNAL` statement.

2.2 The user subroutine

The user must provide a subroutine to compute and return values of the function. The name of the subroutine must be passed as the 10th argument of `VA24A/AD`. The choice of name is for the user – `CALCFX` is used here only as an example.

The single precision version

```
      SUBROUTINE CALCFX(N,X,F)
      REAL X(N),F
```

The double precision version

```
      SUBROUTINE CALCFX(N,X,F)
      DOUBLE PRECISION X(N),F
```

`N`   is an `INTEGER` variable which will be set to *n* the number of variables.

`X`   is a `REAL` (`DOUBLE PRECISION` in the D version) array which will contain the current values of the variables x_i, i=1, 2,..., *n*.

`F`   is a `REAL` (`DOUBLE PRECISION` in the D version) variable which `CALCFX` must set to the function value of $f(\mathbf{x})$ at the point given in `X`.

2.3 Ultimate convergence criterion

The convergence criterion will normally be satisfactory, i.e. `ICONV` is set to 1. However if low accuracy is required, or if it is suspected that the required accuracy is not being achieved, `ICONV` should be set to 2. In this case a more thorough check will be made, possibly increasing the computation time by as much as 30%.

With `ICONV` set to 1 convergence will be assumed when an iteration changes each variable by less than 10% of the required accuracy. With `ICONV` set to 2 such a point is found and it is often displaced by 10 times the required accuracy in each variable. Minimisation is then continued from the new point until a change of less than 10% is again made by an iteration. The two estimates of the minimum are then compared.

2.4 Recommendations

The following points are recommended:

(i)   set `ESCALE` as large as is reasonable, remembering that it should prevent the maximum step jumping from one *valley* to another.

(ii)   set the required accuracy so that `ESCALE` > 100.

(iii)   if the results appear unreasonable, try with different initial values of the variables set in `X(J)`, `J=1,N`.

4 METHOD

The method is described in M.J.D.Powell, Computer Journal, 1964, Vol. 7, No. 2.

The minimum will almost never be found in less than n iterations, each iteration using at least $2n$ values of the function. The iteration is such than each iteration causes the function to decrease, except when the ultimate convergence criterion is being applied with ICONV set to 2.