1 SUMMARY

To **find the minimum of a general function** $f(x)$ **of one variable** given that first derivative values of $f(x)$ are available. An iterative method is used which uses cubic approximations to $f(x)$ based on function and derivative values at two previous points. To start the iterations off an initial estimate of the minimum position and the length of the first step must be provided by the user. The minimum found will normally be the nearest one to the initial estimate going in a downhill direction. An accuracy requirement on the minimum position and a limit on the number of iterations must be specified and the user must provide code to evaluate the function $f(x)$ and its derivative function $g(x)=f'(x)$.

ATTRIBUTES — **Version:** 1.0.0. **Types:** `VD02A`, `VD02AD` **Original date:** May 1964. **Origin:** M.J.D. Powell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

Reverse communication is used, as shown below

```
        K=2 (or 3 if F and G have been set initially)
 6      CALL VD02A(K,X,F,G,MAXFUN,ABSACC,RELACC,STEP)
               ------
        GO TO (1,2,3,4,5),K
 1      set F = f(x) and G = f'(x),  x given in X
               ------
        GO TO 6
 2      minimum found to required accuracy
               ------
 3      accuracy cannot be achieved
               ------
 4      more than MAXFUN function values are needed
               ------
 5      a turning point at which the gradient is zero has been found
               ------
```

`K` is is an `INTEGER` variable used to control the calling sequence. It is set initially by the user to either 2 or 3 to indicate the start of the search. It should be 3 if values of $f(x)$ and $f'(x)$ at the initial point x_0 have already been calculated and are provided in `F` and `G;` if `K` is 2 the routine will arrange these calculations. Once the iterations are in progress the routine will set `K` to one of the five possible values illustrated in the calling sequence given above. If a return is made with `K` set to 5 indicating $f'(x)=0$ it is up to the user to identify the type of turning point and restart if that is what is required. Note the routine cannot be re-entered with `K` > 3.

`X` is is a `REAL` (`DOUBLE PRECISION` in the D version) variable set initially by the user to an estimate x_0 of the minimum position. During the iterations the routine will return to the calling program for function values $f(x)$ and $f'(x)$ and the value of x will be found in `X.` On the final return `X` will have been set to the point corresponding to the function value returned in F. When `K=2` this will give the required minimum position.

`F,` `G` are `REAL` (`DOUBLE PRECISION` in the D version) variables used to hold the value of the function $f(x)$ and its derivative $g(x)=f'(x)$. On the first call the user has the option of setting `F` and `G` to the values corresponding to the initial point. During the iteration, on a return to the calling program with `K=1, F` and `G` must be set by the user to the values of $f(x)$ and $g(x)$ at the point x specified in `X.` On a final return, `K` > 1, `F` will contain the lowest value of $f(x)$ obtained so far and G the corresponding derivative value.

`MAXFUN` is an `INTEGER` and is set by the user to a limit on the number of evaluations of $f(x)$. If `MAXFUN` evaluations are made and the accuracy still not achieved a return with `K=4` is made and `F` set to the lowest value of $f(x)$ obtained so far.

ABSACC, RELACC are REAL (DOUBLE PRECISION in the D version) variables which must be set by the user to the absolute (ABSACC) and relative (RELACC) accuracy required in the final minimum position. If the current position is x_m and the next predicted position is x_{m+1}, the minimum is accepted if $|x_m - x_{m+1|} < |ABSACC|$ or $|x_m - x_{m+1} < |RELACC\, x_{m+1}|$. To select only one of these accuracy requirements the unwanted one may be prevented from operating by setting it to zero.

STEP is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to a reasonable change to make to the variable to start off the iterations. It should ideally be an estimate of the error in the initial estimate x_0 of the true minimum position. A gross overestimate may result in a minimum being found which is not the one nearest to x_0. Any bad estimate will cause more iterations to be taken, but should not affect the final convergence.

3 GENERAL INFORMATION

Use of common: : none

Workspace: : none

Restrictions: : none **Original date:** : May 1964

4 METHOD

To predict the position of the minimum the cubic defined by two function values and their derivatives is used. Precautions are built in to ensure that the search is always downhill from the lowest calculated function value and to prevent an excessive step being taken. The subroutine has been written so that it will converge even if unreasonably high accuracy is demanded.