1 SUMMARY

Given

a) a data point (X,Y) and the associated weights W_x and W_y, and

b) a function $f(x)$ and

c) a starting approximation x_0, the subroutine finds a point $(x, f(x))$ which is 'closest' to the data point by minimising the function

$$SS(x) = W_x(X-x)^2 + W_y(Y-f(x))^2.$$

First and second derivatives of $f(x)$ must be supplied through a subroutine provided by the user.

ATTRIBUTES — **Version:** 1.0.0. **Types:** `VD03A`, `VD03AD`. **Calls:** `FD05`. **Original date:** May 1964. **Origin:** M.J.D. Powell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version

```
CALL VD03A(FN,XD,WX,YD,WY,XO,FX,PX,SSERR,MAXIT)
```

The double precision version

```
CALL VD03AD(FN,XD,WX,YD,WY,XO,FX,PX,SSERR,MAXIT)
```

`FN` is the name of the function subroutine supplied by the user. It must be declared `EXTERNAL` in the calling program, see 2.3.

`XD` is a `REAL` (`DOUBLE PRECISION` in the D version) set by the user to the x data value.

`WX` is a `REAL` (`DOUBLE PRECISION` in the D version) set by the user to a weight associated with the x data value (`WX` > 0).

`YD` is a `REAL` (`DOUBLE PRECISION` in the D version) is set by the user to the y data value.

`WY` is a `REAL` (`DOUBLE PRECISION` in the D version) set by the user to a weight associated with the y data value (`WY` > 0).

`XO` is a `REAL` (`DOUBLE PRECISION` in the D version) set by the user to a starting approximation for x. On return, this value will be over-written with the x value which minimizes the weighted sum of squares $SS(x)$.

`FX` is a `REAL` (`DOUBLE PRECISION` in the D version) which on return will be set to the function value corresponding to the returned x value returned in `XO`.

`PX` is a `REAL` (`DOUBLE PRECISION` in the D version) is set by the user to the required absolute precision in the x value. If set to 0.1, machine precision will be achieved. If set to a larger value, some function evaluations will usually be saved. **Restriction:** `PX` ≥ 0.0.

`SSERR` is a `REAL` (`DOUBLE PRECISION` in the D version) set by the user to an estimate of the error in the calculation of $SS(x)$. If set to 0.0, machine precision will be used. If set to a negative number, a calculated error bound will be used. See Section 2.6.

`MAXIT` is an `INTEGER` set by the user to the maximum number of iterations to be allowed. If set to 0, iteration will only be terminated by convergence.

2.2 Use of COMMON

The routine `VD03A` uses a common area `(VD03B)`, which the user may reference

The single precision version

```
COMMON/VD03B/SS,SD,EPS,LP,IERR
```

The double precision version

```
COMMON/VD03BD/SS,SD,EPS,LP,IERR
```

`SS` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which on return contains the value of the weighted sum of squares $SS(x)$ corresponding to the returned x value.

`SD` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which on return contains the value of $\partial^2(SS(x))/\partial x^2$ corresponding to the returned x value.

`EPS` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which may be set by the user to the largest number such that 1.0 + `EPS` = 1.0. This value is machine dependent. If `EPS` < 0 the routine uses `FD05A/AD` to set the value. The use can override this by setting `EPS` > 0.

`LP` is an `INTEGER` variable which may be set by the user to the logical unit number of the output device to which error messages are sent. It is set by default to 6. If set to zero, no messages will be printed.

`IERR` is an `INTEGER` variable used as an error flag. It is initialized to 0 and then set to a non-zero value in the event of an error, see Section 2.7. The flag is not cleared on subsequent calls.

2.3 Other routines

The user supplied subroutine `FN` must return

1) the function value, $f(x)$
2) the first derivative with respect to x, $f'(x)$, and
3) the second derivative with respect to x, $f''(x)$.

The argument list must be:

```
SUBROUTINE FN(X,FX,FD,SD)
```

`X` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which will contain the x value set by `VD03A`.

`FX` is a `REAL` (`DOUBLE PRECISION` in the D version) variable to be set by the user subroutine to the function value $f(x)$.

`FD` is a `REAL` (`DOUBLE PRECISION` in the D version) variable to be set by the user subroutine to the value of the first derivative $f'(x)$.

`SD` is a `REAL` (`DOUBLE PRECISION` in the D version) variable to be set by the user subroutine to the value of the second derivative $f''(x)$.

Any other parameters of the function required by the subroutine must be supplied through `COMMON`.

Note: The user is not restricted to using `FN` as the name of the subroutine. Any suitable name may be chosen but must agree with the first argument of the call as specified in Section 2.1.

2.4 Weights

Positive weights associated with the x and y data values should ideally be measured or estimated values of $1/V_x$ and $1/V_y$ where V_x and V_y are the variances associated with the measurement of the data values.

2.5 The starting approximation

In the absence of any better information, the x data value XD can be used as a starting approximation.

2.6 Error in the sum of squares

The algorithm initially searches for a minimum of the sum of squares, $SS(x)$. When the reduction in the sum of squares is less than the value of SSERR, the program refines the solution by iterating to a zero of, $SS(x)/x$ which is usually better determined as it does not suffer from cancellation. If SSERR is set to zero, machine precision in the sum of squares will be achieved but the code to locate a zero of $SS(x)/x$ will never be invoked. For problems where $SS(x)$ changes very little over a relatively wide interval of x values, and where SSERR = 0.0 has been specified as the error, round-off error in the calculated sum of squares can result in premature termination at a spurious minimum. This can be avoided by setting SSERR < 0.0 which causes the subroutine to calculate an error bound for the sum of squares. Hence the refinement to a zero of $SS(x)/x$ is automatically invoked when the change in the sum of squares is less than the error bound. In the absence of any precise information about the probable error in the sum of squares, it is safer to set SSERR < 0.0.

2.7 Error messages

Whenever argument restrictions are violated, the subroutine sets the flag IERR in common to a non-zero value and prints an error report (to logical unit LP) of the form

```
VD03A ERROR - ARGUMENT RESTRICTION VIOLATED
ERROR  FLAG VALUE = n
ARGUMENT VALUES ARE --
    ARG 2  = ......
    ARG 3  = ......
    ARG 4  = ......
    ARG 5  = ......
    ARG 6  = ......
    ARG 7  = ......
    ARG 8  = ......
    ARG 9  = ......
    ARG 10 = ......

LABELLED COMMON

    EPS = ......
```

where the error flag can have the following values –

IERR = 1 if WTX ≤ 0.0

IERR = 2 if WTY ≤ 0.0

IERR = 3 if PX < 0.0

A maximum of ten messages will be printed. In the case of an error, argument values other than IERR are left unaltered. The user's program should check that the flag is zero on return.

3 GENERAL INFORMATION

Use of common: labelled common is used (see section 2.2), named VB03B/BD.

Input/output: violation of argument restrictions results in an error message if LP in labelled common has not been set to zero. See sections 2.1 and 2.6.

Restrictions: there are restrictions on the allowable values of the input arguments WTX, WTY, and PX.

4 METHOD

A Newton-Raphson iteration is used, but the x value is bounded at all times to ensure convergence. Other features include

a) if the value of $\partial SS(x)/\partial x$ is not halved in any iteration, the next iteration bisects the bounds;

b) slow oscillatory convergence is prevented by an upper limit to the step length of [Upper bound – Lower bound]/2;

c) slow initial convergence is controlled by ensuring that the step length is greater than or equal to the precision `PX` of x.