

## 1 SUMMARY

A general-purpose subroutine to **produce a graph on a line printer** of several dependent variables against one independent variable. On the line printer, up to 10 separately scaled variables can be graphed to 1 percent accuracy. All dependent variables are identified by symbolic names chosen by the user. By plotting dependent variables (ordinates) across the page, and independent variable (abscissa) down the page, OC02A can put out graphs of arbitrary length. A graph may be put out in sections as data becomes available, and different graphs (from different calls of OC02A) can be put out on different output streams. Where different variables on the same graph fall on top of each other, the symbols representing them are cyclically permuted so that they can all be identified. For each graph, the variables to be plotted can be selected from a larger table of variable values. In the special case where only a single dependent variable is tabulated for graphing, the use of OC02A is slightly simplified.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** OC02A. **Original date:** December 1969. **Origin:** A.R. Curtis, Harwell.

## 2 HOW TO USE THE PACKAGE

### 2.1 The argument list and calling sequence

There is only a single-precision version of this routine:

```
CALL OC02A (IND,NAME,X,H,A,B,DAT,TITLE)
```

**IND** is an INTEGER array containing general information about the graph now required, including which variables from **DAT** are to be graphed. See below for details.

**NAME** is a CHARACTER\*4 array containing symbolic names for the dependent variables to be graphed.

**X** is a REAL variable giving the value of the independent variable for the first set of values of the dependent variables in **DAT**.

**H** is a REAL variable giving the increment in the independent variable for successive sets of values of the dependent variables.

**A, B** are real arrays containing lower and upper bounds to be used in scaling the dependent variables for plotting.

**DAT** is a real two-dimensional array, in which  $DAT(I,J)$  is the value of the  $I$ th dependent variable for abscissa value  $X+(J-1)H$ . The ordinate plotted will be  $(DAT(I,J)-A(I))/(B(I)-A(I))$ . If only a single dependent variable is tabulated in **DAT**, then **DAT** can be a one-dimensional array, and **NAME**, **A**, and **B** can be scalars. In this case,  $IND(4)$  and  $IND(8)$  should both be set to 1 (see below).

**TITLE** is a CHARACTER variable which may optionally contain a title for the graph. The title length is given by  $IND(18)$ , see next section.

### 2.2 Values to be Entered in Array IND

The following values must be set in **IND** before entering OC02A:

**IND(1)** A negative number (to get the heading) if starting a new graph. On return, OC02A saves the number of lines of graph (i.e. the number of abscissa values) in  $IND(1)$ , and this should be left undisturbed if more is to be added to the graph by further calls to OC02A (see below, 4). [Note: **NAME**, **X**, **H** and **TITLE** are referenced only if  $IND(1)$  is negative].

**IND(2)** To output a graph for the line printer on Fortran stream number  $K$ , set  $IND(2) = K$  (e.g. 6 for the standard output stream).

- IND(3) The number  $L$  of simultaneous dependent variables to be graphed (up to 10). If  $L \leq 0$ , OC02A does nothing; if  $L$  is too large, only the first 10 variables are graphed.
- IND(4) The first dimension of DAT, i.e. the spacing of consecutive values of the same dependent variable.
- IND(5) The number  $N$  of lines of graph (i.e. of abscissa values) to be printed. Thus on return IND(1) will have been increased by  $N$  (from zero if initially negative). This value should be set in relation to the amount of data in DAT.
- IND(6) The first value  $J1$  to be used of the second subscript of DAT, corresponding to the first abscissa value to be plotted. This is normally set to 1, unless IND(7) > 1, when IND(6) may have to be reset on subsequent calls (if any).
- IND(7) The increment  $JD$  in the second subscript of DAT to be used for successive lines of graph (thus the abscissa increment is  $H$  times  $JD$ ). This value will usually be 1. DAT must contain values in at least the first  $(J1+JD(N-1))$  columns.
- IND(8), IND(9), ..., Indices  $I1, I2, \dots, IL$  pointing to rows of DAT containing dependent IND(L+7) variables to be plotted in this graph. For each  $Ik$ , the values  $DAT(Ik, J1), DAT(Ik, J1+JD), DAT(Ik, J1+2JD), \dots$ , will be plotted using a symbol identified by NAME(Ik).
- IND(18) should specify the length of the title in TITLE. If IND(18) = 0 no title is printed. If IND(18) is greater than the length of TITLE the length of TITLE is used, i.e. if the title is passed as a literal constant set IND(18) very large to avoid having to count the number of characters in the title.

IND must have dimension at least 18.

### 2.3 Graphs in Sections and Multiple Graphs

If the user's array DAT is not large enough to hold all the data to be plotted (as may easily happen if several graphs, each with several dependent variables, are required, or if it is desired to graph only a few of a large number of variables which have to be computed and tabulated), then each graph may be put out in sections. A section can contain any number of lines of graph (i.e. abscissa values). In particular, a graph may be put out one line at a time if desired, for example as values at a new abscissa point become available. If this is done, DAT need have only one column (or can be one-dimensional). To avoid interleaving graphs with other output, each graph should be put on a separate stream if they are put out in sections. The user should set up a separate IND array for each graph, initialising IND(1) to a negative number. OC02A may then be called once for each graph, to output the heading and any lines of graph for which values are available in DAT. Subsequently, arrays A, B should not be changed, nor should any of the IND (except IND(5) and IND(6) if necessary). Values in DAT may be computed for a further sequence of abscissa values and OC02A may be called again for each graph (i.e. each IND array); and so on. By using values of IND(7) greater than unit, different graphs can be plotted from the same DAT with different abscissa spacings. If DAT contains  $N1$  columns for one call of OC02A, where

$$J1 + JD \cdot (N-1) \leq N1 < J1 + JD \cdot N,$$

and if the first column of DAT on a subsequent call follows immediately after the last of these  $N1$  columns, then for this subsequent call IND(6) will have the value  $(J1+JD \cdot N-N1)$  in order to give proper continuity from section to section. IND(5) will, of course, be set in accordance with the amount of data now in DAT; if this time it is  $N2$  columns, then we must have

$$\frac{N2-IND(6)}{JD} < IND(5) \leq \frac{N2-IND(6)}{JD} + 1$$

which clearly defines IND(5) uniquely. Note that in the normal case  $JD = 1$ , we have  $IND(6) = 1$  and  $IND(5) = N2$ .

No harm will be done if  $IND(5) = 0$  on entry. If IND(1) is negative, the heading will be printed and IND(1) will be left zero; otherwise, OC02A will do nothing.

**3 Variables Changed by OC02A**

OC02A uses no COMMON storage and changes no arguments except IND(1). It calls only standard Fortran subroutines.

4 Sample of Output

A sample graph prepared by OC02A for a printer is given below.

TEST OF OC02A

INDEP VARIABLE X = 0.000E+00+ 1.000E-01\*ABSCISSA

EACH DEP VARIABLE Y(NAME) = OFFSET+SCALE\*ORDINATE(SYMBOL)

NAME	SIN2	COS2
SYMBOL	A	B
OFFSET	0.00E+00	0.00E+00
SCALE	1.00E+00	1.00E+00

