



1 SUMMARY

This subroutine **draws contours of a function** $f(x,y)$ of two variables whose values are specified over an $n \times m$ rectangular mesh. Specifically, the values of $f(x,y)$ must be provided at the mesh points $(x_s + i\Delta x, y_s + j\Delta y)$, $0 \leq i \leq m-1$, $0 \leq j \leq n-1$, where (x_s, y_s) are the coordinates of the bottom left-hand corner of the mesh and where Δx and Δy are the mesh spacings along the x and y axes respectively.

The contours at the boundary of the rectangular region can be improved if the user can provide extra values of $f(x,y)$ immediately outside the region, i.e. along any or all of the lines $x=x_s - \Delta x$, $x=x_s + m\Delta x$, $y=y_s - \Delta y$ and $y=y_s + n\Delta y$. The user must also provide the heights h_i , $i=1,2,\dots,k$ at which the contours are to be drawn. The subroutine uses the GHOST graphics package to draw the contours but does not initialise plotting, draw axes, perform mapping functions or do any labelling. It communicates with GHOST through the Harwell Library subroutine OB13 and is easily adapted for other graphics packages.

ATTRIBUTES — **Version:** 1.0.0. **Types:** OB14A **Calls:** OB13A **Original date:** August 1977. **Origin:** S. Marlow and M.J.D. Powell, 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 OB14A(F, IF, IX, JX, IY, JY, H, K, IERR, DX, DY, XST, YST)
```

F is a REAL array of two dimensions which must hold the values of $f(x,y)$ at the mesh points. These values may be put into a sub array of F defined by the arguments **IX**, **JX**, **IY** and **JY** described below. Specifically, the user must put the values $f(x_s + i\Delta x, y_s + j\Delta y)$ into the array elements $F(IF + i, IY + j)$ $i=0,1,\dots,m-1$, $j=0,1,\dots,n-1$. There is an option for setting the values of $f(x,y)$ immediately outside the region in order to improve the contours at the boundaries, see the description of variable **IBD** in Section 3.

IF is an INTEGER variable and must be set by the user to the first dimension of the array F.

IX, **JX**, **IY**, **JY** are INTEGER variables which must be set by the user to the extents of the sub-array of F for which contours are to be plotted, see argument F. Note that with reference to $JX = IX + m-1$ and $JY = IY + n-1$ and thus define **n** and **m**.

H is a REAL array which must be set by the user to contain the values of contour heights h_i , $i=1,2,\dots,k$, which are to be plotted.

K is an INTEGER variable which must be set by the user to the number k of contour heights.

IERR is an INTEGER variable which is used as an error return parameter and set by the routine, it has one of the following values:

IERR = 0: successful return

IERR = -1: inadequate number of mesh lines in the x or y direction or incorrect values of **IX**, **JX**, **IY**, **JY**.

IERR = -2: the number of contour heights requested was not positive.

DX, **DY** are REAL variables which must be set by the user to the mesh spacings Δx and Δy along the x and y axes respectively.

XST, **YST** are REAL variables which must be set by the user to the coordinates x_s and y_s , of the bottom left-hand corner of the mesh.

2.2 Use of labelled COMMON

The subroutine uses a labelled common block

```
COMMON/OB14C/LP,IBD
```

If the user wishes to change the default values of the labelled COMMON variables then the above statement must be inserted in the calling program and the variables reset.

LP is an INTEGER variable and gives the stream number for diagnostic printing. The default value is 6. To suppress printing set LP to zero.

IBD is an INTEGER variable used as a parameter which can be set to indicate the availability of the values of $f(x,y)$ immediately outside the rectangular region which are useful for contouring near the boundaries of the region. This is defaulted to the value 0 by BLOCK DATA indicating that they are not available.

If any exterior values are available, IBD should be calculated as follows:

First set IBD = 0

- (i) If values of $f(x,y)$ along mesh line $x=x_s-\Delta x$ are available add 1 to IBD.
- (ii) If values of $f(x,y)$ along mesh line $y=y_s-\Delta y$ are available add 2 to IBD.
- (iii) If values of $f(x,y)$ along mesh line $x=x_s+m\Delta x$ are available add 4 to IBD.
- (iv) If values of $f(x,y)$ along mesh line $y=y_s+n\Delta y$ are available add 8 to IBD.

If for example $f(x,y)$ for $x=x_s-\Delta x$ and $x=x_s+m\Delta x$ only are available [for $y=y_s+j\Delta y$ $0\leq j\leq n-1$] then IBD will have the value $1 + 4 = 5$.

The values of $f(x,y)$ must be set in the appropriate positions of the array F, $f(x_s-\Delta x), \dots, f(x_s+m\Delta x), \dots, f(y_s-\Delta y), f(y_s+n\Delta y)$ being set in $F(IX-1, \dots)$, $F(JX+1, \dots)$, $F(\dots, IY-1)$ and $F(\dots, JY+1)$ respectively.

Note that if the user specifies values of IX=1 then (i) above cannot occur and if IY=1 (ii) cannot occur.

3 GENERAL INFORMATION

Use of common: : See Section 2.2.

Workspace: : Two real arrays of length 127 are used for workspace. They do not restrict the use of OB14 in any way. **Calls:** : Subroutine OB13 is called.

Input/output: : Diagnostic printing, See Section 2.2. **Original date:** : October 1977

4 METHOD The rectangular mesh is divided up into rectangles with a typical rectangle being bounded by the mesh lines $x=x_s+i\Delta x$, $x=x_s+(i+1)\Delta x$, $y=y_s+j\Delta y$ and $y=y_s+(j+1)\Delta y$ [i,j satisfying $0\leq i\leq m-1$, $0\leq j\leq n-1$]. Each rectangle is split into eight (similar) triangles by the two lines $x=x_\alpha$, $y=y_\alpha$ and the four lines [taking the four sign combinations] $2\Delta x[y-y_\alpha]\pm 2\Delta y[x-x_\alpha]\pm \Delta x\Delta y=0$ where $x_\alpha=x_s+(i+\frac{1}{2})\Delta x$ and $y_\alpha=y_s+(j+\frac{1}{2})\Delta y$. Approximations for the values of $f(x,y)$ at the vertices and midpoints of the sides of the triangle are then calculated by fitting a piecewise quadratic with continuous first derivatives over the rectangle, using only the 16 values of $f(x,y)$ at the points $x=x_s+i_\alpha\Delta x$, $i_\alpha=i-1, i, i+1, i+2$ and $y=y_s+j_\alpha\Delta y$, $j_\alpha=j-1, j, j+1, j+2$. For further details see AERE Report TP531 *Piecewise Quadratic Surface Fitting for Contouring* by M.J.D. Powell. Each triangle is then applied to subroutine OB13 which plots the part of a conic that is inside the triangle. For further details of this stage see AERE Report R-8336 *A Fortran Subroutine for Plotting the Part of a Conic that is Inside a Given Triangle*.

5 EXAMPLE

Suppose that we want to draw contours of the function $f(x,y)=4(\sin 2x \sin y - \sin x \sin 2y + \cos 4x + \cos 4y)$ for the region $0 \leq x \leq 1$, $0 \leq y \leq 1$ and contour heights $0, \pm 1, \pm 2, \dots, \pm 8$. The example below shows how to use OB14 to draw contours where the mesh spacing is .2 in both the x and y directions with values of the function also being supplied immediately outside the boundary.

```

DIMENSION F(8,8),H(17),S1P(8),S2P(8),C4P(8)
COMMON/OB14C/LP,IBD
IF=8                      set first dimension of array
PI=4.0*TAN(1.)
CALL PAPER(1)             switch on graph plotter
CALL PSPACE(.05,.95,.05,.95) set plotter space
CALL MAP(0.,1.,0.,1.)    set mathematical space
K=17                      number of contour heights
DO 1 I=1,K1
H(I)=FLOAT(I-9)          calculate contour heights
DO 2 I=1,8
X=(I-2)*.2*PI
S1P(I)=SIN(X)            calculate values of sub-expressions of f(x,y)
S2P(I)=SIN(2.*X) C4P(I)=COS(4.*X)
DO 3 I=1,8                calculate f(x,y) on mesh point
DO 3 J=1,83
F(I,J)=(S2P(I)*S1P(J)-S1P(I)*S2P(J)+C4P(J))*4.0
IBD=15                   set IBD to indicate boundary values available
DX=.2                    set mesh spacing
DY=.2
XST=0                    set coordinates of bottom left hand
YST=0                    corner of mesh
CALL OB14A(F,IF,2,7,2,7,H,K,IERR,DX,DY,XST,YST)
CALL BOX(0.,1.,0.,1.)    draw box around contours
CALL GREND               switch off graph plotter
STOP
END

```

An example of the output from this program is given on the following page.