1 SUMMARY

This subroutine **determines rapidly whether points are internal or external to a two-dimensional region.** The boundary of the region may be either polygonal or given by a pair of parametric piecewise cubics, $x(t)$, $y(t)$ which are continuous with continuous first derivatives. The subroutine is most likely to be useful when the number of vertices of the polygon (or joins of the cubics) is very large.

The subroutine calculates the local maxima and minima, y_i, i=1,2, . . .s, of the y coordinate of points on the boundary and stores information at each intersection of a line y=y_i with the boundary. The amount of working storage needed therefore depends on the number, τ, of such intersections.

ATTRIBUTES — Version: 1.0.0. **Types:** GA15A, GA15AD. **Calls:** FD05, KB05, NB01. **Original date:** April 1985. 385 lines. **Origin:** J.K.Reid, Harwell. **Remark:** This is a slightly rewritten version of GA05 and supersedes it.

2 HOW TO USE THE PACKAGE

2.1 Argument list

To specify the region:

The single precision version

```
CALL GA15A(N,KIND,WORK,IW,IND,LND,X,Y,T,XD,YD,LP)
```

The double precision version

```
CALL GA15AD(N,KIND,WORK,IW,IND,LND,X,Y,T,XD,YD,LP)
```

To test a point:

The single precision version

```
CALL GA15B(XP,YP,RESULT,N,KIND,WORK,IW,IND,LND,X,Y,T,XD,YD,LP)
```

The double precision version

```
CALL GA15BD(XP,YP,RESULT,N,KIND,WORK,IW,IND,LND,X,Y,T,XD,YD,LP)
```

N is an INTEGER variable which must be set by the user to specify that the number of vertices of the polygon or joins of the cubics is N-1. The first vertex is also regarded as the N th vertex. It is not altered. **Restriction:** N≥4.

KIND is an INTEGER variable which the user must set to specify the kind of region involved. It may take these values:

1. A polygon
2. A region whose boundary is given by piecewise cubics in x and y, these functions being functions of a parameter t. **Restriction:** KIND=1 or KIND=2.

WORK is a REAL (DOUBLE PRECISION in the D version) array of length IW, which must be at least 4τ for polygons and at least 2N+4τ for piecewise cubics. The value of τ is output in IND(2,1). WORK must not be altered between a call to GA15A/AD and a corresponding call to GA15B/BD.

IW is an INTEGER variable which must be set by the user to specify the dimension of WORK. **Restriction:** for polygon IW>4τ; for piecewise cubics IW>2N+4τ.

IND is an INTEGER workspace of dimensions (2,LND), where LND≥τ. IND(1,1) is set negative after a successful entry to GA15A or to a positive number on a failure. IND must not be altered between a call to GA15A/AD and a corresponding call to GA15B/BD.

On failure the subroutine prints a diagnostic message and sets IND(1,1) to one of the following values:–

(1) when IW is too small,
(2) when LND is too small,
(3) when the data is not periodic (see X,Y),
(4) when a failure in NB01 occurs,
(5) when N<4 (see N),
(6) when KIND≠1 or KIND≠2,
(7) when a value of T is out of order,
(8) when XD or YD is not periodic.

LND is an INTEGER variable specifying the second dimension of IND. It is not altered. **Restriction:** LND≥τ.

X,Y are REAL (DOUBLE PRECISION in the D version) arrays of dimension N which the user must set to the *x* and *y* coordinates of the vertices (or joins). They are not altered. **Restriction:** X(1)=X(N), Y(1)=Y(N).

T is a REAL (DOUBLE PRECISION in the D version) array of dimension N required when KIND=2. T must be set by the user to contain the *t* values at the ends of each pair of piecewise cubics $x(t)$, $y(t)$ specifying the boundary. **Restriction:** T(1)<T(2)<. . .<T(N).

XD,YD are REAL (DOUBLE PRECISION in the D version) arrays of dimension N, required when KIND=2. They must be set by the user to hold $x'(t),y'(t)$ at t=T(1), T(2), . . .T(N). They are not altered. **Restriction:** XD(1)=XD(N), YD(1)=YD(N).

LP is an INTEGER variable which must be set by the user to a unit number for printing or to a non-positive number if printing is to be suppressed.

XP,YP are REAL (DOUBLE PRECISION in the D version) variables which the user must set to the *x* and *y* coordinates of the point to be tested. They are not altered.

RESULT is a REAL (DOUBLE PRECISION in the D version) variable which the subroutine sets to a positive value if the point lies within the region, to zero if the point is on the boundary and to a negative value if it lies outside the region. The distance to the boundary along the line y=YP is at least ABS(RESULT).

2.2 Polar coordinates

Polar coordinates r and θ may be used instead of x and y provided the range of θ is less than 2π. The call is identical except that X,Y,XD,YD,XP,YP should contain r and θ coordinates and their derivatives.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly: KB05, NB01, and the private subroutine GA15C/CD.

Input/output: Output is under the control of argument LP.

Restrictions: N≥4, IW>4τ (for polygon), IW>2N+4τ (for piecewise cubics), LND>τ, X(1)=X(N), Y(1)=Y(N), KIND=1 or KIND=2, T(1)<T(2) < . . . <T(N), XD(1)=XD(N), YD(1)=YD(N).

4 METHOD

In preparation for the testing of a large number of points GA15A/AD computes and stores the following:

1. (Piecewise cubic case only). For each interval between joins, the maximum and minimum values of x.
2. The local maxima and minima of y, regarded as functions of t, ordered in increasing y values. Let these be $y_1<y_2< . . .<y_s$.
3. All the values of t for which $y(t)=y_{i,}$ $1\le i\le s$, ordered in increasing t values. We call the corresponding boundary values 'critical points'.
4. Pointers to permit the rapid identification of all the intervals between adjacent critical points corresponding to

the interval y_{i-1}, y_i, i=2,3, . . .s. We call such intervals ‘boundary segments’.

5. For each boundary segment, the maximum and minimum values of x.

GA15B/BD uses this information to determine whether the point P=(XP,YP) is internal as follows:

1. Perform a binary search on the set $y_1, y_2, . . . y_s$ to find an interval such that y_k<YP≤y_{k+1}.
2. On the corresponding boundary segment there is a single point of intersection of the boundary and the line y=YP. To find which side of P it lies we compare XP with the maximum and minimum values of x for the segment; if this does not give us a decision we find between which pair of adjacent vertices (or joins) the point of intersection lies by a binary search; in the polygonal case we then find the point itself and in the piecewise cubic case we use the known maximum and minimum values of x for this interval between joins and if this still does not give us a decision we solve the appropriate equation to find the point itself.
3. Determine whether there are an even or odd number of intersections of y=YP with the boundary on one side of P, using the results of step 2.

For further details see J.K.Reid, A note on the approximation of plane regions, Comp.J.**14** (1971), 307-308 and J.K.Reid, AERE R7298 (1972).

5 EXAMPLE OF USE

As a very simple example the following code calls GA15AD to define a simple region and return the value of τ and then calls GA15BD to see if three points are within the region. The region is the square with vertices (0,1), (1,0), (0,–1), (–1,0). This has one maximum and one minimum and hence the value of τ is two.

```
      DOUBLE PRECISION RESULT,XP,YP,WORK(20),X(5),Y(5),XD(5),YD(5),T(5)
      INTEGER IND(2,5)
      DOUBLE PRECISION XPTEST(3),YPTEST(3)
      DATA N/5/,KIND/1/,IW/20/,LND/5/,LP/6/
      DATA X/0.0000D0,1.0000D0,0.0000D0,-1.0000D0,0.0000D0/
      DATA Y/1.0000D0,0.0000D0,-1.0000D0,0.0000D0,1.0000D0/
      DATA XPTEST/0.0000D0,0.5000D0,1.0000D0/
      DATA YPTEST/0.0000D0,0.5000D0,1.0000D0/
C  DEFINE THE REGION AND FIND THE VALUE OF TAU
      CALL GA15AD(N,KIND,WORK,IW,IND,LND,X,Y,T,XD,YD,LP)
      WRITE(LP,90)IND(2,1)
90    FORMAT(' RETURN FROM GA15AD, TAU = ',I10)
C  TEST POINTS
      DO 150 I=1,3
        XP=XPTEST(I)
        YP=YPTEST(I)
        WRITE(LP,100)
100     FORMAT(/,' RETURN FROM GA15BD  POINT ')
        CALL GA15BD(XP,YP,RESULT,N,KIND,WORK,IW,IND,LND,X,Y,T,XD,YD,
     *LP)
        IF(RESULT.GT.0.0D0) WRITE(LP,120) XP,YP
120     FORMAT(1H+,25X,2F8.4,3X,'IS INSIDE THE BOUNDARY ')
        IF(RESULT.LT.0.0D0) WRITE(LP,130) XP,YP
130     FORMAT(1H+,25X,2F8.4,3X,'IS OUTSIDE THE BOUNDARY ')
        IF(RESULT.EQ.0.0D0) WRITE(LP,140) XP,YP
140     FORMAT(1H+,25X,2F8.4,3X,'IS ON THE BOUNDARY ')
150     CONTINUE
        STOP
        END
```

This produces the following output

```
 RETURN FROM GA15AD, TAU =          2

 RETURN FROM GA15BD  POINT  0.0000  0.0000   IS INSIDE THE BOUNDARY

 RETURN FROM GA15BD  POINT  0.5000  0.5000   IS ON THE BOUNDARY

 RETURN FROM GA15BD  POINT  1.0000  1.0000   IS OUTSIDE THE BOUNDARY
```