NAG Fortran Library Routine Document

E02ZAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

E02ZAF sorts two-dimensional data into rectangular panels.

2 Specification

```
 SUBROUTINE E02ZAF(PX, PY, LAMDA, MU, M, X, Y, POINT, NPOINT, ADRES,
1                  NADRES, IFAIL)
 INTEGER           PX, PY, M, POINT(NPOINT), NPOINT, ADRES(NADRES),
1                  NADRES, IFAIL
 real              LAMDA(PX), MU(PY), X(M), Y(M)
```

3 Description

A set of m data points with rectangular Cartesian co-ordinates x_r, y_r are sorted into panels defined by lines parallel to the y and x axes. The intercepts of these lines on the x and y axes are given in LAMDA(i), for $i = 5, 6, \ldots, \mathrm{PX} - 4$ and MU(j), for $j = 5, 6, \ldots, \mathrm{PY} - 4$, respectively. The subroutine orders the data so that all points in a panel occur before data in succeeding panels, where the panels are numbered from bottom to top and then left to right, with the usual arrangement of axes, as shown in the diagram. Within a panel the points maintain their original order.

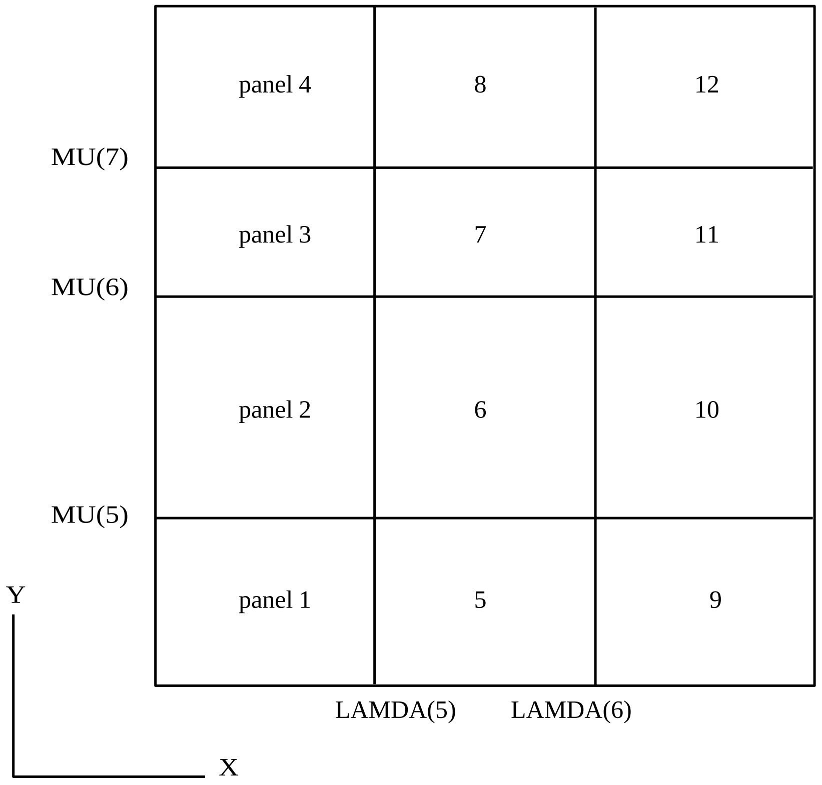

Figure 1

A data point lying exactly on one or more panel sides is taken to be in the highest-numbered panel adjacent to the point. The subroutine does not physically rearrange the data, but provides the array POINT which contains a linked list for each panel, pointing to the data in that panel. The total number of panels is $(\mathrm{PX}-7) \times (\mathrm{PY}-7)$.

4 References

None.

5 Parameters

1: PX – INTEGER *Input*

2: PY – INTEGER *Input*

On entry: PX and PY must specify eight more than the number of intercepts on the x axis and y axis, respectively.

Constraint: $\mathrm{PX} \geq 8$ and $\mathrm{PY} \geq 8$.

3: LAMDA(PX) – ***real*** array *Input*

On entry: LAMDA(5) to LAMDA$(\mathrm{PX}-4)$ must contain, in non-decreasing order, the intercepts on the x axis of the sides of the panels parallel to the y axis.

4: MU(PY) – ***real*** array *Input*

On entry: MU(5) to MU$(\mathrm{PY}-4)$ must contain, in non-decreasing order, the intercepts on the y axis of the sides of the panels parallel to the x axis.

5: M – INTEGER *Input*

On entry: the number m of data points.

6: X(M) – ***real*** array *Input*

7: Y(M) – ***real*** array *Input*

On entry: the co-ordinates of the rth data point (x_r, y_r), for $r = 1, 2, \ldots, m$.

8: POINT(NPOINT) – INTEGER array *Output*

On exit: for $i = 1, 2, \ldots,$ NADRES, POINT$(m+i) = \mathrm{I1}$ is the index of the first point in panel i, POINT$(\mathrm{I1}) = \mathrm{I2}$ is the index of the second point in panel i and so on.

POINT(IN) $= 0$ indicates that X(IN),Y(IN) was the last point in the panel.

The co-ordinates of points in panel i can be accessed in turn by means of the following instructions:

```
   IN = M + I
10 IN = POINT(IN)
   IF (IN.EQ. 0) GOTO 20
   XI = X(IN)
   YI = Y(IN)
       .
       .
       .
   GOTO 10
20 ...
```

9: NPOINT – INTEGER *Input*

On entry: the dimension of the array POINT as declared in the (sub)program from which E02ZAF is called.

Constraint: $\mathrm{NPOINT} \geq \mathrm{M} + (\mathrm{PX}-7) \times (\mathrm{PY}-7)$.

10: ADRES(NADRES) – INTEGER array *Workspace*

11: NADRES – INTEGER *Input*

On entry: the value (PX – 7) × (PY – 7), the number of panels into which the (x, y) plane is divided.

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

The intercepts in the array LAMDA, or in the array MU, are not in non-decreasing order.

IFAIL $= 2$

On entry, PX < 8,
or PY < 8,
or M ≤ 0,
or NADRES $\neq$ (PX – 7) × (PY – 7),
or NPOINT $<$ M + (PX – 7) × (PY – 7).

7 Accuracy

Not applicable.

8 Further Comments

The time taken by this routine is approximately proportional to $m \times \log(\text{NADRES})$.

This subroutine was written to sort two dimensional data in the manner required by routine E02DAF. The first 9 parameters of E02ZAF are the same as the parameters in E02DAF which have the same name.

9 Example

This example program reads in data points and the intercepts of the panel sides on the x and y axes; it calls E02ZAF to set up the index array POINT; and finally it prints the data points in panel order.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E02ZAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, MAXPX, MAXPY, NADMAX, NPOINT
      PARAMETER        (MMAX=20,MAXPX=12,MAXPY=12,NADMAX=(MAXPX-7)
     +                  *(MAXPY-7),NPOINT=MMAX+NADMAX)
      INTEGER           NIN, NOUT
      PARAMETER         (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER           I, IADRES, IFAIL, M, NADRES, PX, PY
*     .. Local Arrays ..
      real              LAMDA(MAXPX), MU(MAXPY), X(MMAX), Y(MMAX)
      INTEGER           ADRES(NADMAX), POINT(NPOINT)
*     .. External Subroutines ..
      EXTERNAL          E02ZAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02ZAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*) M
      IF (M.GT.0 .AND. M.LE.MMAX) THEN
         READ (NIN,*) PX, PY
         IF (PX.LE.MAXPX .AND. PY.LE.MAXPY) THEN
            NADRES = (PX-7)*(PY-7)
*           Read data points and intercepts of panel sides
            READ (NIN,*) (X(I),Y(I),I=1,M)
            IF (PX.GT.8) READ (NIN,*) (LAMDA(I),I=5,PX-4)
            IF (PY.GT.8) READ (NIN,*) (MU(I),I=5,PY-4)
*           Sort points into panel order
            IFAIL = 0
*
            CALL E02ZAF(PX,PY,LAMDA,MU,M,X,Y,POINT,NPOINT,ADRES,NADRES,
     +                  IFAIL)
*
*           Output points in panel order
            DO 60 I = 1, NADRES
               WRITE (NOUT,*)
               WRITE (NOUT,99999) 'Panel', I
               IADRES = M + I
   40          IADRES = POINT(IADRES)
               IF (IADRES.GT.0) THEN
                  WRITE (NOUT,99998) X(IADRES), Y(IADRES)
                  GO TO 40
               END IF
   60       CONTINUE
            GO TO 20
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I4)
99998 FORMAT (1X,2F7.2)
      END
```

9.2 Program Data

```
E02ZAF Example Program Data
  10
   9
  10
0       0.77
0.70    1.06
1.44    0.33
0.21    0.44
1.01    0.50
1.84    0.02
0.71    1.95
1.00    1.20
0.54    0.04
1.53    0.18
1.00
0.80
1.20
0
```

9.3 Program Results

```
 E02ZAF Example Program Results

 Panel   1
    0.00   0.77
    0.21   0.44
    0.54   0.04

 Panel   2
    0.70   1.06

 Panel   3
    0.71   1.95

 Panel   4
    1.44   0.33
    1.01   0.50
    1.84   0.02
    1.53   0.18

 Panel   5

 Panel   6
    1.00   1.20
```