UR1985_41

1/5

1985/41. PTCOORD - A FORTRAN program for measuring the co-ordinates of points on a map

R.G. Richardson

Abstract

Provided that the co-ordinates of three points (two of them with the same northing) are known, other points may be numbered and the co-ordinates scaled off using a digitiser.

USING THE PROGRAMS

Ensure that the digitiser is in point mode and type PTCOORD. The program then requests the user to digitise three points, the first two on the same horizontal (user) line and the third not on the same line.

The easting and northing of the first point, the easting of the second point, and the northing of the third point are then entered.

The co-ordinates of points may now be measured by entering the point number (O<N<99999) using the buttons on the digitiser cursor and then pressing the * button on the cursor to store the co-ordinates. Measurement is terminated by pushing the # button. The co-ordinates are output on the printer.

THE PROGRAM

PTCOORD (Appendix 1)

The program performs three point scaling and deskewing of the plan. The cursor buttons 0-9 are used to enter the point number with the * button indicating that the point co-ordinates are to be stored and the # button signalling the end of the run.

[22 July 1985]

APPENDIX 1

Program PTCOORD

```
*PTCOORD.CSS - TO RUN PTCOORD
L PTCOORD;  AS 6,PR:;  ST
$EXIT
```

```
$TITL  PTCOORD.FTN - MEASURE THE COORDINATES OF POINTS
C PTCOORD.FTN
      LOGICAL FOUND
C USED TO WORK OUT WHETHER TO PRINT A HEADER AND WHETHER TO PRINT THE
C DUMMY LINE AT THE END
      COMMON/IO/LUNO,LUNI
C FOR COMPATIBILITY WITH DIGTRD
      OPEN(UNIT=5,FILE='CON:')
C OPEN THE CONSOLE AND,BY DEFAULT,THE DIGITISER AS LOGICAL UNIT 5
      LUNI=5
      LUNO=5
      FOUND=.FALSE.
      WRITE(5,100)
  100 FORMAT(' COORDINATE MEASUREMENT PROGRAM')
C
      WRITE(5,103)
  103 FORMAT(' ENSURE THAT THE DIGITISER IS IN POINT MODE'/
     .       ' DIGITISE TWO KNOWN POINTS ON A HORIZONTAL LINE'/
     .       ' AND THEN A THIRD KNOWN POINT NOT ON THE SAME LINE')
      CALL DIGTRD(IBTN,X0,Y0)
      CALL DIGTRD(IBTN,XC,YC)
      CALL DIGTRD(IBTN,XV,YV)
C HAVE NOW READ ENOUGH POINTS TO ALLOW CALCULATION OF X AND Y SCALES
      WRITE(5,105)
  105 FORMAT(' USER COORDINATES OF FIRST POINT AS X,Y?')
      READ(5,*) X0MAP,Y0MAP
      WRITE(5,106)
  106 FORMAT(' USER X COORDINATE OF SECOND POINT?')
      READ(5,*) XCMAP
      WRITE(5,107)
  107 FORMAT(' USER Y COORDINATE OF THIRD POINT?')
      READ(5,*) YVMAP
C NOW CALCULATE THE SKEWING
      XD=XC-X0
      YD=YC-Y0
      XDIGIT=SQRT(XD*XD+YD*YD)
C DISTANCE ALONG HORIZONTAL LINE IN DIGITISER UNITS
      THETA=ATAN(YD/XD)
C THE SKEW OF THE USER GRID (RADIAN)
      CTHETA=COS(THETA)
      STHETA=SIN(THETA)
C NEEDED FOR COORDINATE TRANSFORMATION
C USE TRANSFORM ASSUMING A COMMON ORIGIN BY
C     REMOVING DIGITISER OFFSET  X0,Y0
C     ROTATING
C            XR=XD*COS(THETA)+YD*SIN(THETA)
C            YR=-XD*SIN(THETA)+YD*COS(THETA)
C     THE ROTATED COORDINATES ARE THEN SCALED USING XSCALE,YSCALE
C     AND IF DESIRED MADE RELATIVE TO THE USER ORIGIN BY ADDING
C     X0MAP,Y0MAP
C NOW NEED THE VERTICAL DISTANCE BETWEEN THE HORIZONTAL LINE AND THE
C THIRD POINT
      XD=XV-X0
      YD=YV-Y0
C REMOVE DIGITISER OFFSET
      YR=-XD*STHETA+YD*CTHETA
C NOW HAVE DESKEWED AND GOT Y DISTANCE IN DIGITSER UNITS
      DMAPX=ABS(XCMAP-X0MAP)
      DMAPY=YVMAP-Y0MAP
```

```
C THE KNOWN MAP DISTANCES
      XSCALE=DMAPX/XDIGIT
      YSCALE=ABS(DMAPY/YR)
C
C NOW HAVE ALL PARAMETERS FOR GETTING USER COORDINATES
C
C NOW PROMPT FOR COORDS
      WRITE(5,100)
  100 FORMAT(' ENTER THE NUMBER OF THE POINT THEN '/
     .       ' PUSH THE * BUTTON TO REGISTER THE POINT'/
     .       ' PUSH THE # BUTTON TO STOP')
   30 ILAST=0
   35 CALL DIGTRD(IBTN,XD,YD)
      IF (IBTN .EQ. 11) GOTO 40
C IF PUSH # ARE AT END OF JOB
      IF (IBTN .NE. 10) THEN
C PUSHED * BUTTON SO HAVE THE POINT COORDS
      ILAST=ILAST*10+IBTN
      GOTO 35
      ENDIF
C BUILD THE POINT NUMBER
C
C N.B.  IF ACCURACY IS A PROBLEM USE DOUBLE PRECISION
C
      XD=XD-X0
      YD=YD-Y0
C TAKE OFF THE DIGITISER OFFSET
      XF=XD*CTHETA+YD*STHETA
      YF=-XD*STHETA+YD*CTHETA
C ROTATE OUT ANY SKEW
      XF=XF*XSCALE
      YF=YF*YSCALE
C SCALE TO USER UNITS
      XF=XF+X0MAP
      YF=YF+Y0MAP
C ADD ON THE USER ORIGIN
      IF (.NOT. FOUND) THEN
      WRITE(6,200)
  200 FORMAT(' REF    EAST',6X,'NORTH')
      FOUND=.TRUE.
      ENDIF
      WRITE(6,201) ILAST,XF,YF
  201 FORMAT(I6,2F11.3)
      GOTO 30
   40 CONTINUE
      IF (FOUND) WRITE(6,202)
  202 FORMAT(1X)
      END
$INCLUDE SYS3:DIGTRD.FTN,-
```

```
      SUBROUTINE DIGTRD(IBTN,X,Y)
C FOR READING ONE NUMBER AT A TIME FROM THE DIGITISER ALTHOUGH
C THE DIGITISER PUTS OUT BLOCKS OF 4
C THE VALUES OF X AND Y ARE IN INCHES
C IBTN IS THE CURSOR BUTTON PRESSED 0 1 2 3 4 5 6 7 8 9  *  #
C VALUE                             0 1 2 3 4 5 6 7 8 9 10 11
C IBTN=-4 IMPLIES STREAM MODE (AS DISTINCT FROM SWITCHSTREAM).
      INTEGER*4 IBTN,IIX(4),IIY(4),IIBTN(4),POINT
      REAL*4 X,Y
C UNBLOCKED AND BLOCKED X,Y AND BUTTON VALUES
      DATA NREAD/5/,POINT/1H./
C NREAD IS THE NEXT NUMBER OF DATA VALUES OF THE FOUR TO READ
C SET TO 5 TO FORCE READ TO START WITH
      SAVE IIX,IIY,IIBTN
C CANT SAVE NREAD AS IT IS SET TO ZERO INITIALLY BY THE SAVE
      COMMON /IO/ LUNO,LUNI
C COMMON BLOCK IS USED FOR COMPATABILITY WITH DIGITISER SOFTWARE
C AND SHOULD BE SET PRIOR TO CALLING THIS ROUTINE
C LUNI=LUNO=LOGICAL UNIT NUMBER OF DIGITISER
   10 IF (NREAD .EQ. 5) THEN
C READ A FULL BLOCK OF FOUR OR FIRST TIME
      READ(LUNI,100) (IIBTN(I),IIX(I),IIY(I),I=1,4)
  100 FORMAT(4(A1,I5,I5))
C READ ONE BLOCK OF FOUR RECORDS
      NREAD=1
      ENDIF
C
      IF (IIBTN(NREAD) .EQ. POINT) THEN
      NREAD=5
      GOTO 10
      ENDIF
C REMAINDER IS FILLED BLOCK SO GET A NEW BLOCK
C NOW GET THE VALUES
      X=IIX(NREAD)*0.001
      Y=IIY(NREAD)*0.001
C CONVERTED TO INCHES HERE
C NOW GET THE RIGHT BUTTON NUMBER
      CALL ILBYTE(IBTN,IIBTN(NREAD),0)
C PICK UP THE CURSOR BUTTON AS A DECIMAL BUTTON
C AND
      IBTN=IBTN-48
C SUBTRACT 48 (THE DECIMAL VALUE OF THE CHARACTER 0) TO GET THE BUTTON
C NUMBER
      NREAD=NREAD+1
      RETURN
      END
```