1/8

UR1981-49

1981/49. A revised FORTRAN program for the drift correction of gravity data.

R.G. Richardson

Abstract

This program removes a linear drift from gravity readings made at times between the occupation of two base stations. The base stations need not be the same.

THE PROGRAM

GRVDFT (*Appendix 1*)

The original program (Richardson and Leaman, 1981) provided output only to the input device. To provide a record of all data entry, output is now to the printer as well as the input device.

This program reads station numbers, station times, and meter readings on logical unit 5 and prints drift and datum corrected differences on logical units 6 and 7. All times are in 24 hour notation, with a decimal point separating the hours and minutes (*e.g.* 10.25). Non-standard subroutine calls are:

CALL SYSCOM (I, 'AS 0622, 0720, 0899*') sets the logical units for the printer, optional output, and core buffer.

CALL SYSCOM (I, 'AS 0700*') sends the optional output to the null device.

CALL SYSCOM (I, 'AS 0620, 0720, 0820*') resets the logical units to the system console.

Station numbers are read in free integer format using subroutine INFREE, and all other data is read in F10.1 format.

The data for each run is:

SNUM	base 1 station number
TIME1	the time of reading base 1
BASE1	the meter reading at base 1
SHIFT1	any shift to be applied to base 1 to correct it to the datum (in the same units as BASE1)
SNUM	base 2 station number
TIME2	the time of reading base 2
BASE2	the meter reading at base 2 (in the same units as BASE1)
SHIFT2	any shift to be applied to base 2 to correct it to the same datum as base 1 (in the same units as BASE1)

The data for each station is:

SNUM	the station number - if SNUM = 0 the program stops
STIME	the time of reading the station
STNVAL	the station reading (in the same units as BASE1)

Output is:

Logical Unit 6

The base station data, and then for each station the station number, the drift (and datum) corrected difference, the reading time, and the uncorrected reading.

Logical Unit 7

The drift (and datum) corrected difference. This output appears only if the user requests the differences to be printed immediately.

REFERENCE

RICHARDSON, R.G.; LEAMAN, D.E. 1981. FORTRAN program for the drift correction of gravity data. *Unpubl.Rep.Dep.Mines Tasm.* 1981/37.

[30 October 1981]

APPENDIX 1

Program GRVDFT

```
C TITLE GRVDFT
C A PROGRAM TO REMOVE A LINEAR DRIFT FROM GRAVITY READINGS
C RELATIVE TO TWO BASE STATIONS WHICH MAY OR MAY NOT BE THE SAME
C AND MAY OR MAY NOT BE THE SURVEY DATUM.
C TIMES MUST BE IN 24 HOUR NOTATION
C OUTPUT TO THE PRINTER IS STORED IN ARRAYS AND PRINTED AT THE END OF
C THE RUN
      INTEGER*2 SNUM(150), SNUM1, SNUM2, K, AOK, YES, I
      LOGICAL IER
      REAL TIME1, TIME2, BASE1, BASE2, SHIFT1, SHIFT2, DRIFT, STIME,
     . STNVAL(150), VALUE(150), ATIME(150), ATIME1, ATIME2
      DATA YES/1HY/
      K=1
      CALL SYSCOM(I, 'AS 0622, 0720, 0899*')
C SET LOGICAL UNIT 6 TO PRINTER, 7 TO INPUT DEVICE, 8 TO CORE BUFFER
      WRITE(5, 110)
  110 FORMAT(' PRINT DIFFERENCES IMMEDIATELY?')
      READ(5, 111) AOK
  111 FORMAT(A1)
      IF (AOK .NE. YES) CALL SYSCOM(I, 'AS 0700*')
C IF OUTPUT NOT WANTED SET LOGICAL UNIT 7 TO A NULL DEVICE
      WRITE(5, 96)
   96 FORMAT(' BASE 1 NUMBER?')
      CALL INFREE(SNUM1, IER, 5)
   55 WRITE(5, 100)
  100 FORMAT(' BASE 1 TIME?')
      READ(5, 101) TIME1
  101 FORMAT(F10.1)
C READ TIME AS E.G. 10.56 FOR TEN FIFTY-SIX
C MUST BE IN 24 HOUR FORMAT
      IF (TIME1 .GT. 24.0) GOTO 55
C NO DECIMAL POINT IN TIME1
      ATIME1=TIME1
C SAVE FOR LATER PRINT
      CALL HOURS(TIME1)
C CORRECTION TO HOURS
      WRITE(5, 102)
  102 FORMAT(' BASE 1 READING?')
      READ(5, 101) BASE1
      WRITE(5, 103)
  103 FORMAT(' BASE 1 SHIFT?')
      READ(5, 101) SHIFT1
      WRITE(5, 97)
   97 FORMAT(' BASE 2 NUMBER?')
      CALL INFREE(SNUM2, IER, 5)
   56 WRITE(5, 104)
  104 FORMAT(' BASE 2 TIME?')
      READ(5, 101) TIME2
C MUST BE IN 24 HOUR FORMAT
      IF (TIME2 .GT. 24.0) GOTO 56
C TIME HAS NO DEC. PT.
      ATIME2=TIME2
```

```
C SAVE FOR LATER PRINT
      CALL HOURS(TIME2)
      WRITE(5,105)
  105 FORMAT(' BASE 2 READING?')
      READ(5,101) BASE2
      WRITE(5,106)
  106 FORMAT(' BASE 2 SHIFT?')
      READ(5,101) SHIFT2
C HAVE READ THE BASE STATION PARAMETERS
C
C NOW CALCULATE THE DRIFT PER HOUR
      DRIFT=((BASE2-SHIFT2)-(BASE1-SHIFT1))/(TIME2-TIME1)
C
    9 WRITE(5,98)
   98 FORMAT(' STN. NO.?')
      CALL INFREE(SNUM(K),IER,5)
      IF (SNUM(K) .EQ. 0) GOTO 20
C TERMINATE RUN BY 0 STSTION NUMBER
   10 WRITE(5,107)
  107 FORMAT(' STN TIME?')
      READ(5,101) STIME
      IF (STIME .GT. 24.0) GOTO 10
C TIME HAS NO DEC. PT.
      ATIME(K)=STIME
C STORE THE TIME FOR THE PRINT AT THE END
      CALL HOURS(STIME)
      WRITE(5,108)
  108 FORMAT(' STN VALUE?')
      READ(5,101) STNVAL(K)
C
C NOW CALCULATE DRIFT CORRECTED VALUE
      VALUE(K)=STNVAL(K)-(DRIFT*(STIME-TIME1))-(BASE1-SHIFT1)
      WRITE(7,109) VALUE(K)
  109 FORMAT(' VALUE',F9.1)
      K=K+1
C OK SO INCREMENT COUNTER BY ONE
      GOTO 9
   20 K=K-1
      IF (K .EQ. 0) GOTO 21
C NO STATIONS
      WRITE(6,200) SNUM1,ATIME1,BASE1,SHIFT1,SNUM2,ATIME2,
     . BASE2,SHIFT2
  200 FORMAT('   SNUM  TIME     BASE    DRIFT'/2(I6,F6.2,2F9.1/),
     . /'  SNUM     DIFF     TIME     OBS')
      WRITE(6,201) (SNUM(I),VALUE(I),ATIME(I),STNVAL(I),I=1,K)
  201 FORMAT(I6,F9.1,F9.2,F9.1)
      WRITE(6,202)
  202 FORMAT(////)
   21 CALL SYSCOM(I,'AS 0620,0720,0820*')
C RE-ASSIGN ALL LOGICAL UNITS TO THE INPUT DEVICE
      STOP
      END
      SUBROUTINE HOURS(TIME)
      REAL HR,MINS,TIME
      HR=IFIX(TIME+0.005)
C ENSURE TIME ROUNDS DOWN TO THE CORRECT HOUR
      MINS=TIME-HR
C MINUTES
      MINS=MINS*1.666667
C CONVERT MINUTES TO DECIMAL HOURS
      TIME=HR+MINS
      RETURN
      END
  HOURS   0024R   FUNC/SUB
  HOURS   008CR   FUNC VAR
  .Q      0000R   EXT FUNC
  .P      0000R   EXT FUNC
  TIME    002AR   FORM PAR
  HR      0094R   REAL VAR
  MINS    0098R   REAL VAR
  IFIX    0000R   EXT FUNC
  .W      0000R   EXT FUNC

0000 ERRORS
```

```
      SUBROUTINE INFREE(INUM,IER,ILUN)
C A FREE FORMAT INPUT ROUTINE FOR INTEGER*2 NUMBER FROM
C ILUN.
C INUM,ILUN ARE BOTH INTEGER*2
C IER IS A LOGICAL
C MAKES USE OF THE CORE BUFFER TO GO FROM FREE FORMAT TO
C FIXED FORMAT BY WRITING IN A FORM SUITABLE FOR RE-READING
C IN I6 FORMAT.
      INTEGER*2 INUM,ILUN,A(6),BLA,I,J,K,JUP,L
      LOGICAL IER
      DATA BLA/1H /
C USED TO DETERMINE NON-BLANK CHARACTERS
      IER=.TRUE.
      READ(5,100,ERR=101) A
  100 FORMAT(6A1)
C READ IN A CHARACTER FORMAT
      DO 10 I=1,6
      IF (A(I) .NE. BLA) GOTO 20
C LOOK FOR FIRST NON-BLANK CHARACTER
   10 CONTINUE
C IF FALL OUT OF LOOP ALL CHARACTERS ARE BLANK - HENCE INUM=0
      IER=.FALSE.
      INUM=0
  101 RETURN
C THE QUICK RETURN
C
C NOW COPY THE FREE FORMAT INTEGER RIGHT JUSTIFIED INTO THE SAME
C ARRAY
   20 CONTINUE
      DO 11 J=I,6
      IF (A(J) .EQ. BLA) GOTO 21
   11 CONTINUE
C FIND THE END OF THE NUMBER, EITHER BY FINDING A BLANK AT THE END
C OR BY REACHING THE SIXTH CHARACTER
      J=7
   21 J=J-1
      K=6
      JUP=J
C NOW COPY RIGHT JUSTIFIED AND FILL LEADING PART WITH BLANKS IF NEEDS BE
      DO 12 L=I,J
      A(K)=A(JUP)
      K=K-1
      JUP=JUP-1
   12 CONTINUE
      IF (K .EQ. 0) GOTO 15
      DO 14 L=1,K
      A(L)=BLA
   14 CONTINUE
C NOW FILLED WITH LEADING BLANKS AND RIGHT JUSTIFIED
C
C SO
C WRITE OUT TO CORE BUFFER AND READ BACK IN IN I6 FORMAT
```

```
   15 CONTINUE
      WRITE(8,100) A
      READ(8,111,ERR=101) INUM
  111 FORMAT(I6)
C NOW HAVE THE NUMBER
      IER=.FALSE.
      RETURN
      END
  INFREE 0024R  FUNC/SUB
  INFREE 0220R  FUNC VAR
  .Q     0000R  EXT FUNC
  .P     0000R  EXT FUNC
  INUM   002AR  FORM PAR
  IER    002CR  FORM PAR
  ILUN   002ER  FORM PAR
  A      0228R  INT2 VAR
  BLA    0034R  INT2 VAR
  I      0234R  INT2 VAR
  J      0236R  INT2 VAR
  K      0238R  INT2 VAR
  JUP    023AR  INT2 VAR
  L      023CR  INT2 VAR
  100    0066R  LABEL
  101    00BER  LABEL
  @H     0000R  EXT FUNC
  10     0094R  LABEL
  20     00C6R  LABEL
  11     00EAR  LABEL
  21     0104R  LABEL
  12     015CR  LABEL
  15     01ACR  LABEL
  14     019AR  LABEL
  111    01FCR  LABEL

0000 ERRORS
```