a

Computers & Geosciences Vol. 16, No. 8, pp. 1193–1207, 1990
Printed in Great Britain. All rights reserved

0098-3004/90 $3.00 + 0.00

TILTVEC: A FORTRAN-77 PROGRAM FOR THE TILT CORRECTION OF PALEOCURRENT DATA WITH RESOLUTION OF INCONGRUITIES

DILIP SAHA and TAPAN CHAKRABORTY
Geological Studies Unit, Indian Statistical Institute, 203 Barrackpore Trunk Road,
Calcutta-700 035, India

(*Received* 8 *December* 1989; *accepted* 9 *January* 1990)

Abstract—TILTVEC is a FORTRAN-77 program for tilt corrections of paleocurrent data, which may be measurements on planar crossbeds or trough crossbeds. Tilt correction is effected by rotation of the dipping formation to its assumed initial horizontal position around present formation strike. The program borrows partly from a rotation of poles algorithm based on the transformation of the axes of the reference coordinate system. But new program modules take care of fallacies arising out of the distinction between axial data (poles) and vector data (paleocurrent directions).

Key Words: Axial data, FORTRAN-77, Paleocurrent, Rotation of poles, Transformation of axes, Transformation matrix, Vector data.

INTRODUCTION

A FORTRAN-77 program SPIN8 was written originally for handling multiple rotation of poles around an axis of general orientation (Saha, 1987). The SPIN8 algorithm exploits a transformation of axes algebraic rules. A number of other computer programs have been published for rotation of poles or tilt corrections of paleocurrent data (Noble, 1964; Parks, 1970, 1974; Starkey, 1970). The Noble method involves calculation of a rotation matrix by considering the rotation of two independent vectors always to a fixed plane (*XZ* plane of the Cartesian coordinate frame). The method is not amenable to situations where rotation of poles around any axis of general orientation is required (Saha, 1987). Nor would the algorithm readily take paleocurrent data involving measurements on both trough and planar crossbeds. The algorithms used in the Starkey method or the Parks method are based on trigonometric solution of spherical triangles for each pole. Therefore, these are inefficient in situations involving single or multiple rotation of hundreds or thousands of poles. Although the Parks method can be modified to accept paleocurrent data collated through measurements on both trough and planar crossbeds, the program was written primarily to accept planar crossbed measurements only as input data.

Poles, which are by definition undirected lines, should be classified as axial data. The characteristics of axial orientation data are that any set of such data if imagined to pass through the center of a reference sphere, has the minimum trivial symmetry of $\bar{1}$ (center of symmetry, notations of point group symmetry). In other words axial data are but nondirected vectors (Mardia, 1972). Because paleocurrent data are *sensu stricto* vectors, rotation of such data cannot be effected correctly by a direct use of SPIN8. The Noble algorithm has the same drawback. For effective tilt corrections of paleocurrent data an algorithm is presented here. The algorithm borrows partly from SPIN8 and includes additional program modules to buttress the incongruities arising out of the difference between vector data and axial data.

There are many geologic situations where rotation of a raw paleocurrent data set is necessary before other analysis can be employed. Let us consider the following simplest situation. Apparent paleocurrent direction is obtained from measurements on foreset dips of crossbed cosets or trough axis azimuth (down current direction). Present overall attitude of the formation containing the crossbed cosets is different from horizontal, and hence assumed to have undergone post-depositional orientation changes. As for example, simple buckle rotation around a horizontal fold axis may impart dips to originally horizontal beds. Differential vertical block movements or fault block rotations may be responsible for the present dipping attitude of the beds. The tilt correction procedure as elaborated here assumes that a simple rotation around the present bed strike connects original and observed orientations. The basic tenet may not hold in situations such as folding of beds where buckling and bulk homogeneous strain proceed hand-in-hand; or, where fault drag occurs around an axis other than horizontal. Hence, before using the proposed tilt correction procedure one needs to make sure that the basic tenet holds good.

THE GEOMETRY OF SIMPLE TILT CORRECTION

Let us assume that cosets of planar crossbeds or trough crossbeds are contained by a representative bed whose orientation gives the overall attitude of the concerned rock unit (henceforth this general orientation will be referred to as *bed* dip and strike). A dipping *bed* (dip amount = Θ) is restored to its assumed initial horizontal position by a rotation around *bed* strike. A rotation through Θ or $(180 - \Theta)$ may effect this rotation. However, in situations where the *bed* is not overturned rotation through Θ (< 90) only is justified (Fig. 1). Because the transformation matrix in SPIN8 algorithm is obtained from solutions of simultaneous equations of degree two, two alternative solutions are available corresponding to clockwise and anticlockwise rotations of poles.

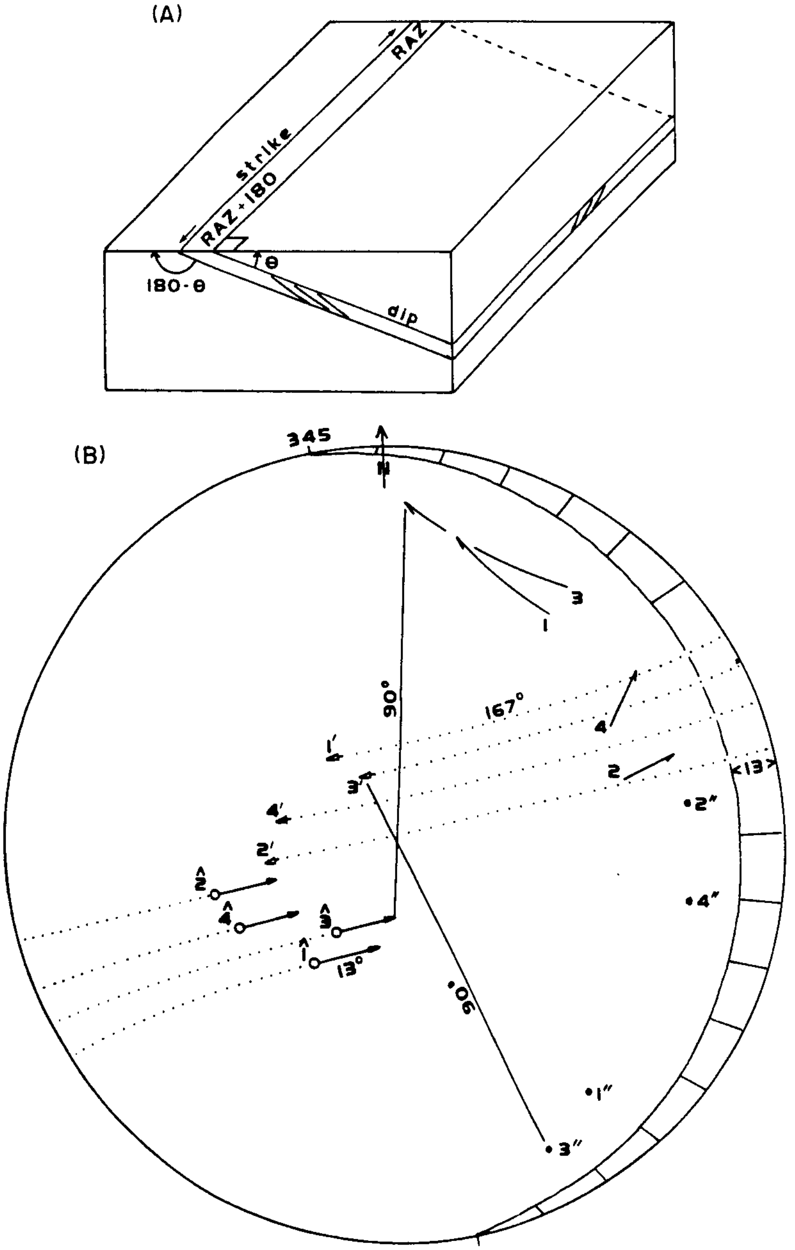

Figure 1. Restoration of dipping bed to assumed initial horizontal position. A—Anticlockwise rotation of bed around RAZ end of strike through angle Θ will effect restoration; clockwise rotation through $(180 - \Theta)$ around RAZ will do same, but crossbed relations preclude latter alternative. B—Lower hemisphere Schmidt net plot showing graphical restoration of dipping *bed* and concomitant reorientation of crossbed attitude. Poles of four coset crossbeds are marked. $\hat{1}, \hat{2}, \hat{3}, \hat{4}$ (1, 2, 3, 4, corresponding lines of foreset dip). After anticlockwise rotation through 13 (= *bed* dip amount) around *bed* strike 345 poles are at tips of solid arc trajectories. Ends of dotted trajectories (1′, 2′, 3′, 4′) correspond to clockwise rotation of crossbed poles through 167 (180 − 13 = 167). Dots marked 1″, 2″, 3″, 4″ are corresponding rotated dip lines.

THE GEOMETRY OF SIMPLE TILT CORRECTION

Let us assume that cosets of planar crossbeds or trough crossbeds are contained by a representative bed whose orientation gives the overall attitude of the concerned rock unit (henceforth this general orientation will be referred to as *bed* dip and strike). A dipping *bed* (dip amount = Θ) is restored to its assumed initial horizontal position by a rotation around *bed* strike. A rotation through Θ or $(180 - \Theta)$ may effect this rotation. However, in situations where the *bed* is not overturned rotation through Θ (< 90) only is justified (Fig. 1). Because the transformation matrix in SPIN8 algorithm is obtained from solutions of simultaneous equations of degree two, two alternative solutions are available corresponding to clockwise and anticlockwise rotations of poles.

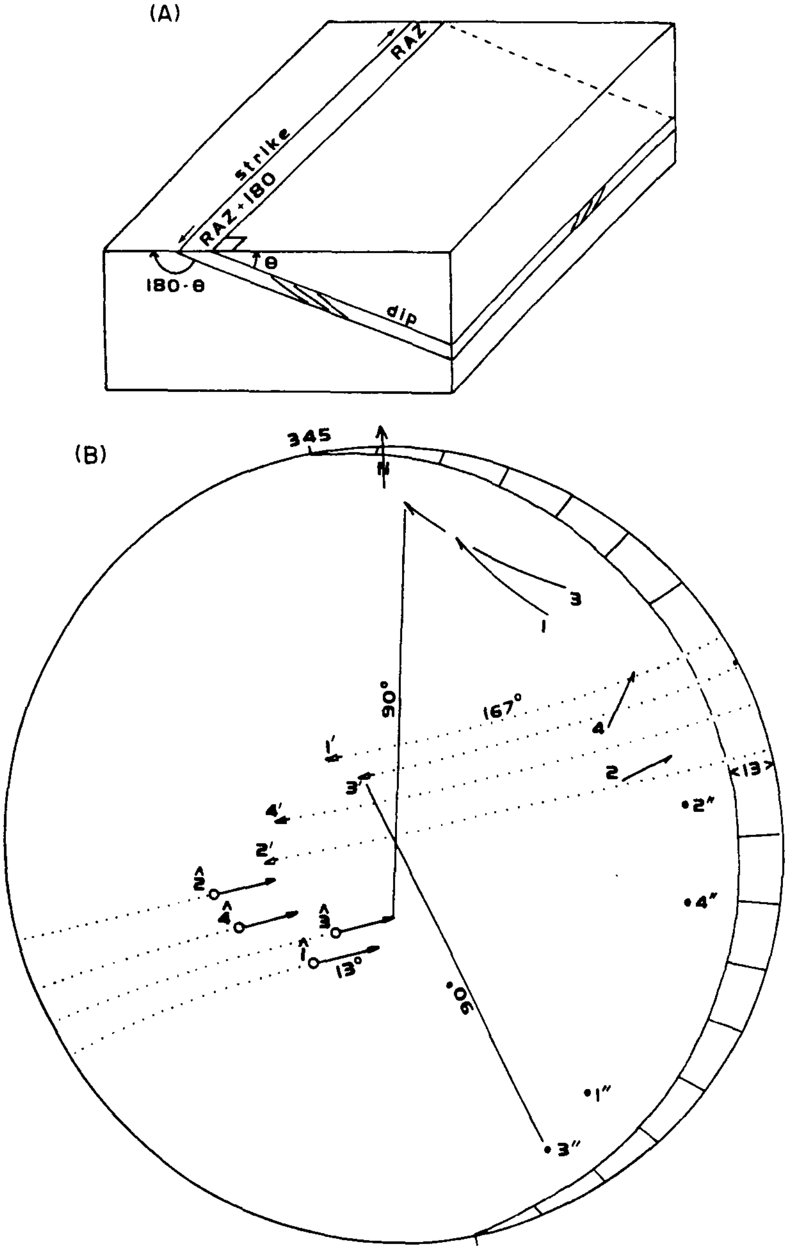

Figure 1. Restoration of dipping bed to assumed initial horizontal position. A—Anticlockwise rotation of bed around RAZ end of strike through angle Θ will effect restoration; clockwise rotation through $(180 - \Theta)$ around RAZ will do same, but crossbed relations preclude latter alternative. B—Lower hemisphere Schmidt net plot showing graphical restoration of dipping *bed* and concomitant reorientation of crossbed attitude. Poles of four coset crossbeds are marked, $\hat{1}, \hat{2}, \hat{3}, \hat{4}$ (1, 2, 3, 4, corresponding lines of foreset dip). After anticlockwise rotation through 13 (= *bed* dip amount) around *bed* strike 345 poles are at tips of solid arc trajectories. Ends of dotted trajectories (1', 2', 3', 4') correspond to clockwise rotation of crossbed poles through 167 (180 – 13 = 167). Dots marked 1″, 2″, 3″, 4″ are corresponding rotated dip lines.

In this instance one needs to select between the two ends of the *bed* strike to effect rotation in one consistent sense only. We stick to the alternative where anticlockwise rotation through Θ brings the dipping *bed* (not foreset) back to horizontal. Depending on the orientation of the planar crossbeds with respect to the overall attitude of the enclosing formation the restored foreset orientations will differ. The sense of rotation is an important factor in such restoration (Fig. 1B). The restoration of the crossbed orientations will follow different paths depending on the crossbed type.

Planar crossbeds

The downward facing direction of the crossbed normal is rotated through an angle Θ around that end of the *bed* strike, which effects anticlockwise rotation of poles. Usually the crossbed dip amount and directions are recorded. Therefore, orientation of the downward facing normal is obtained first knowing that the azimuth of the normal is 180° from the dip direction and plunge of the normal is 90° minus foreset dip amount. After the normal is restored to its initial orientation via tilt correction, one needs to do back-calculation to obtain foreset true dip direction and amount.

Trough crossbeds

Trough axis becomes horizontal through restoration of the *bed* to its assumed horizontal position. However, the nature of axial data precludes any distinction between two ends of a line (horizontal or otherwise). Therefore, unmodified SPIN8 may return restored trough axis azimuth diametrically opposite to that which it correctly should be. The adopted procedure rectifies such errant values as follows. Let us consider the lower hemisphere equal area projection of planes and lines (bed and lineation on bed). As the bed is restored to its *initial* horizontal position only one-half of the primitive circle corresponds to the lower hemisphere arc of the dipping bed (Fig. 2). All trough axes treated as lower hemisphere directed vectors should rightly lie on this specific one-half of the primitive. Situations where trough axis presently is directed towards upper hemisphere, are taken care of as follows.

The down-current direction is given by the direction of concavity of the traces of trough crossbed laminae on the bedding plane (Potter and Pettijohn, 1977, p. 95). However, through tectonic tilts, trough axis azimuth as measured may make an angle >90° with the *bed* dip direction. Because trough axis azimuth can take any value between 1 and 360°, the

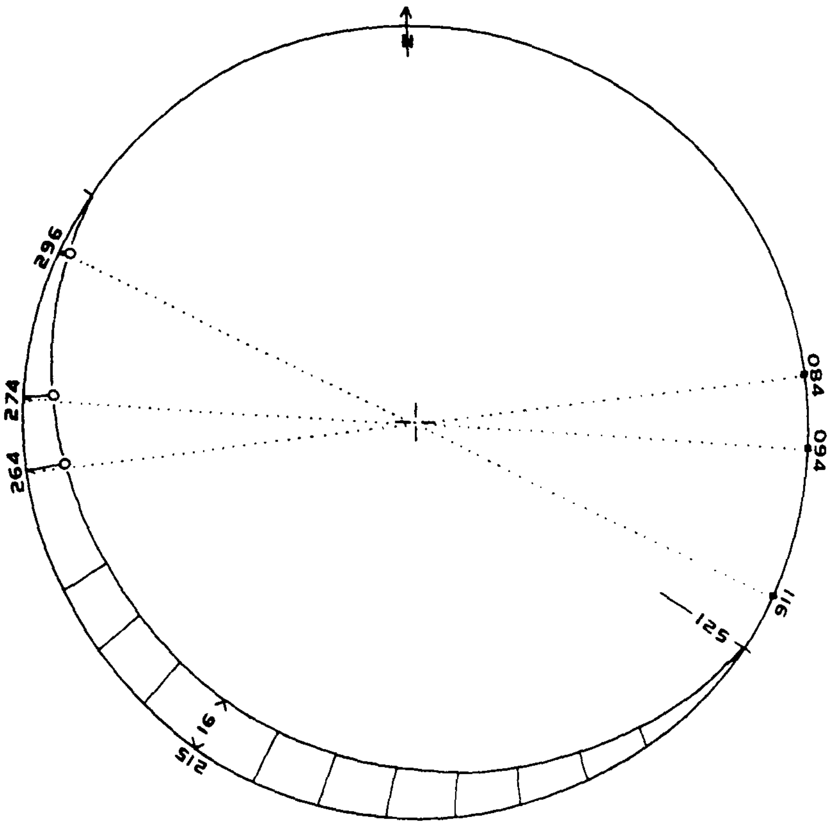

Figure 2. Restoration of trough axes lying on tilted bed dipping 16° towards 215. Lower hemisphere directed trough axes (open circles) are reorientated to positions on western one-half of primitive circle (arrowheads) as bed is rotated anticlockwise through 16° around bed strike 125. Had trough axes been upper hemisphere directed, corrected trough axes would lie on eastern one-half of primitive (solid squares).

absolute difference between trough axis azimuth and bed dip direction is the determining factor. For any trough axis directed towards upper hemisphere the above difference will be >90° but <270°.

Generally the raw data provides only trough axis azimuth. Input data forces a zero in the data field for vector plunge amount. The apparent dip of the *bed* along the trough axis provides the plunge amount. However, when the azimuth of the trough axis lies outside ±90° of the bed dip direction this calculation imparts a fallacy, which is transcended through use of subprogram module Q4S4.

TILTVEC PROCEDURE

Tilt corrections of paleovectors similar to paleocurrent direction is effected through borrowings from SPIN8, a general purpose routine for rotation of pole (Saha, 1987). As demonstrated earlier, in this situation all rotations are around a horizontal axis, that is one end of the *bed* strike. An abridged version of SPIN8, renamed as subroutine TILT, is used here as a program module. The MAIN unit within the TILTVEC procedure calls a number of subroutines (see flowchart in Fig. 3). The subroutines TRNSF1, TRNSF2, TRNSF3 are as they were in SPIN8. A new version of subroutine KDELTA is offered, where modifications are mainly in syntax. Additional subroutine modules which seem new in TILTVEC procedure are described next.

Subroutine VMEAN calculates the mean vector and consistency ratio from a set of paleocurrent orientation data (azimuth only). The formulae used for vector mean calculation are borrowed from Mardia (1972, p. 20). The mean direction x_0 of $\Theta_1, \Theta_2, \ldots, \Theta_n$ is given by

$$x_0 = \arctan(S/C),$$

where

$$C = 1/n \sum \cos\Theta_i, \quad i = 1, n,$$

$$S = 1/n \sum \sin\Theta_i, \quad i = 1, n. \qquad (1)$$

The consistency ratio (L), expressed in terms of percent is given by

$$L = 100(C^2 + S^2)^{1/2}/n. \qquad (2)$$

Notice that Equation (1) is different from that applicable to classified data (see for example, Sengupta and Rao, 1966).

Subroutine NORMAL returns the plunge and azimuth of the normal to a set of planar crossbeds. The orientation data, either input or output, are in degrees.

Subroutine Q4S4 is called to detect whether a paleocurrent direction (DIR) is "upper hemisphere directed" with respect to *bed* dip direction (see an earlier section for details). The subroutine attaches subsidiary tags to vectors with azimuth outside ±90° of the *bed* dip direction. In such situations DIR is returned as DIR + 180.

Subroutine Q4S returns the correct end of a trough axis azimuth where unmodified SPIN8 would offer an "errant" situation. This is because of the inherent difference between axial data and vector data.

Subroutine FIXRAZ returns that end of the azimuth of *bed* strike rotation around which ensures anticlockwise rotation of vectors.

Subroutine FLSDIP returns the apparent dip amount (Θ') of a bed along a direction α, where bed true dip amount (Θ) and direction (Ω) is provided.

$$\Theta' = \arctan(\tan\Theta \times \sin\beta), \qquad (3)$$

where $\beta = |\Omega + 90 - \alpha|$.

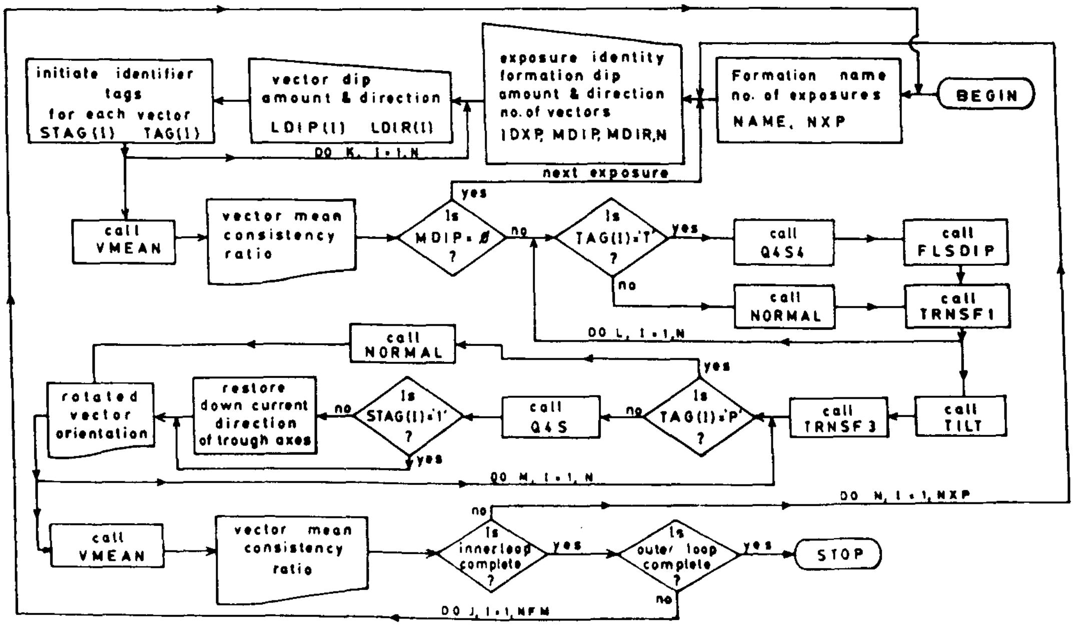

Figure 3. Flowchart for TILTVEC procedure.

A program listing is provided in the Appendix. In spite of the obvious risk of duplication, all the subroutine modules are listed for the sake of completeness, and slight but necessary changes in the argument list associated with some of the modules.

AN EXAMPLE

Comparison of paleocurrent data obtained from different facies, different formations and outcrops on a regional scale may aid interpretation significantly about depositional processes, paleogeography, and basin geometry (Cant and Walker, 1976; Ricci-Lucchi, 1975; Rust, 1981). Paleocurrent data on Proterozoic sandstones of Sullavai Group, Godavari Valley, South India include measurements from formations which are locally or regionally tilted. Table 1 shows part of this database in a format acceptable to TILTVEC. Paleocurrent orientation data recorded from isolated exposures usually can be assumed rightly to belong to the same formation on geological grounds. However, these can be compared meaningfully only if the effects of local tilts are nullified first.

The TILTVEC program has been developed and tested on an IBM compatible PC. The input data format is indicated in the program listing (see the Appendix). Table 2 shows mean vector orientation before and after tilt correction of data in Table 1. Note that the consistency ratio, a rough measure of dispersion around the mean vector, as obtained in this output of Table 2 is different for uncorrected and corrected data; this is only an artefact of disregarding the plunge of the vectors in subroutine VMEAN. The orientation of individual vectors after tilt correction

Table 1. Sample of paleocurrent data measured from Proterozoic Sullavai Group, Godavari Valley, South India

```
*VENKATPUR SST        004 exposures ... record 001
---------------------------------------------------
ldip(i) ldir(i)      ■dip ■dir   n
---------------------------------------------------
NTPC Rd               13 078    10  ... record 002
30 035
39 075
22 035
37 062
30 096
38 062
33 085
24 063
37 030
25 063

GODAVARI RIVER        14 242     5  ... record 013
09 130
09 144
10 161
29 237
19 152

MNCL N'LA             00 000     4  ... record 019
23 052
20 063
25 058
23 070

BELAMPALLI STN. QRY   17 050    20  ... record 024
57 035
37 080
35 044
45 055
30 046
33 065
42 060
40 082
47 040
37 040
39 042
33 045
44 040
38 092
38 053
36 055
42 051
45 052
 0 085
28 065                              ... record 044
```

Table 2. Comparison of mean vector directions before and after tilt correction. Tilt correction is unnecessary where formation dip is zero as in MNCL NALA exposure (see Table 1)

```
 *VENKATPUR SST.                         4  exposures

    Exposure Id.            Mean Vector      Consistency Ratio

1. NTPC Rd                      61                94  %
     TILT CORRECTED   **        50                85  %

2. GODAVARI RIVER              161                81  %
     TILT CORRECTED   **       112                67  %

3. MNCL NALA                    61                99  %

4. BELAMPALLI STN. QRY          56                96  %
     TILT CORRECTED   **        59                91  %
```

are written on to a disc file RDATA1.DAT and saved for future use. Table 3 shows part of the listing of RDATA1.DAT.

Acknowledgment—Two anonymous reviewers suggested some improvements on an earlier version of the manuscript. They are gratefully acknowledged.

Table 3. Tilt corrected paleocurrent data of Table 1. Note that MNCL NALA exposure data are missing (cf. Table 1) as corresponding formation dip is zero

```
 TILT CORRECTED *VENKATPUR SST          4 exposures

ldip(i) ldir(i)    mdip mdir    n             tilt correction

NTPC Rd             13   78    10   ** 13.DEGREES ROTATION AROUND 348.
22  13
26  74
15   1
25  55
18 108
26  55
20  89
12  47
30  13
13  49

GODAVARI RIVER      14 242      5   ** 14.DEGREES ROTATION AROUND 152.
19  88
18  93
16 101
15 233
23 117

BELAMPALLI STN. QRY 17  50     20   ** 17.DEGREES ROTATION AROUND 320.
41  31
24  99
18  39
28  58
13  41
17  79
25  66
27  99
30  35
20  33
22  37
16  40
27  35
27 113
21  55
19  59
25  52
28  53
 0  84
12  85
```

C

REFERENCES

Cant, D. J., and Walker, R., 1976, Development of braided fluvial facies model for Devonian Battery Point Sandstone, Quebec: Can. Jour. Earth Sci., v. 13, no. 1, p. 102–119.

Mardia, K. V., 1972, Statistics of directional data Academic Press, London, 357 p.

Noble, D. C., 1964, Mathematical rotation of orientation data: Geol. Soc. America Bull., v. 75, no. 3, p. 247–248.

Parks, J. M., 1970, Computerized trignometric method for rotation of structurally tilted sedimentary directional features: Geol. Soc. America Bull., v. 81, no. 2, p. 537–540.

Parks, J. M., 1974, Paleocurrent analysis of sedimentary crossbed data with graphic output using three integrated computer programs: Jour. Geology, v. 6, no. 4, p. 353–362.

Potter, P. E., and Pettijohn, F. J., 1977, Paleocurrent and basin analysis: Springer-Verlag, New York, 425 p.

Ricci-Lucchi, F., 1975, Sediment dispersal in turbidite basins: examples from Miocine of Northern Apennines: 9th Intern. Sedimentological Congress, Nice, theme 5(2), p. 347–352.

Rust, B. R., 1981, Alluvial deposits and tectonic style: Devonian and Carboniferous succession in Eastern Europe, *in* Miall, A. D., ed., Sedimentation and tectonics in alluvial basins: Geol. Soc. Canada Spec. Paper, v. 23, p. 49–76.

Saha, D., 1987, SPIN8: A FORTRAN-77 program for automated rotation of poles: Computers & Geosciences, v. 13, no. 3, p. 235–254.

Sengupta, S., and Rao, J. S., 1966, Statistical analysis of cross bending azimuths from Kamthi Formation around Bheemaram, Pranhita-Godavari Valley, Sankhya: Indian Jour. Statistics, Ser. B, v. 28, pt. 1–2, p. 165–174.

Starkey, J., 1970, A computer program to prepare orientation diagrams, *in* Experimental and natural rock deformation: Proc. Intern. Symposium, Darmstadt: Springer-Verlag, Berlin, p. 51–74.

APPENDIX

Program Listing

```
c        SUBROUTINE TILTVEC
c        written by D. SAHA, Indian Statistical Institute, 1989
c        computes exposurewise vector mean and consistency
c        ratio of paleocurrent data (cross bed and/or trough axis);
c        satelite program units do tilt correction prior to
c        vector computation, if necessary;
c        tilt correction routine borrows from SPIN8, a general purpose
c        routine for rotation of poles, written by D. SAHA, refer to
c        original publication in Computers & Geosci. v.13, p.235-254
c        of 1987 while using this routine.
c
c        Overall attitude of the formation is referred to as bed
c         strike and dip.
c         Tectonic tilts are assumed to be around present bed
c         strike and dip;
c         tilt corrections involve rotation around this strike
c         through an angle equal to bed dip;
c         both cross bed orientation data (dip amount and direction)
c         and trough axial orientaion are accepted;
c         in the latter case trough axis azimuth (down-current
c         direction) only is available as raw data;
c         enter zero in the data field for dip amount of trough axis.
c
c         DATA STRUCTURE AS FOLLOWS
c
c Record one: cols. 1-20 = formation name, type character,
c             cols. 21-23 = no. of exposures in this formation
c Record two: cols. 1-3 = exposure Id, col. 4 = blank, cols 5-6 =
c             bed dip amount, col. 7 = blank, cols. 8-10 = bed
c             dip direction, cols. 12-14 = no. of vectors in this
c             exposure (< = 60)
c Record   3: cols.1-2 = cross bed dip (zero if trough),
c             col. 3 = blank, cols. 5-7 = cross bed dip direction
c Next nvec-1 records as in record 3
c Next record as in record two
c Next nvec records as in record 3
c Repeat nxp times for all exposures in a single formation
c Repeat above nfm times to enter data for all formations
c
c          VARIABLES
c
c          no. of formations to be handled = NFM
c          formation name = NAME, exposure Identifier = IDXP (character)
c          no. of exposures in each formation = NXP
c          no. of vectors measured in one exposure = NVEC
c          bed dip amount = MDIP, direction = MDIR
c          vector dip (plunge) = LDIP, direction = NDIR
c          RAZEM = vector azimuth after tilt correction, GAMMA = 90-LDIP
c          LDIP,NDIR,RAZEM,GAMMA must be arrays of dimension NVEC
```

```
c       so are arrays LP,MP,NP, which give the direction cosines of
c       input vectors
c       array DC of dimension 3 x NVEC stores direction cosines of
c       vectors after tilt correction
c       TAG & STAG, arrays of dimension NVEC each, hold character
c       identifiers for each vector type, 'T' for trough; 'P' for planar;
c       '1' for trough axis within 90 degrees of MDIP, '0' otherwise;
c
        PARAMETER(NFM=1,NVEC=60)
c
c       change nfm parameter depending on the no. of formations
c       to be handled in one run
c       NVEC needs to be changed as well if there are more than 60
c       measurements in one exposure
c
        CHARACTER*20 NAME,IDXP
        CHARACTER*1 TAG(NVEC),STAG(NVEC)
        REAL LDIP(NVEC),NDIR(NVEC),RAZEM(NVEC),GAMMA(NVEC),LP(NVEC),
     +  MP(NVEC),NP(NVEC),DC(3,NVEC)
        EQUIVALENCE(LDIP,GAMMA)
c
        CONVRT=ATAN(1.)/45.
c
        OPEN(2,STATUS='OLD',FILE='A:DATA1.DAT')
c
c       raw data on a floppy disc accessible through logical A drive
c       file name "DATA1.DAT"
c       modifications of the open statement above may be necessary
c       for other than DOS operating system
c
        OPEN(7,STATUS='NEW',FILE='A:RDATA1.DAT')
c
c       vector orientation after tilt correction are written on to
c       file "RDATA1.DAT" residing on floppy accessed through A drive
cc
        DO 1000 K=1,NFM
        READ(2,'(A20,I3)') NAME,NXP
        WRITE(6,'(''1   '',A20,5X,I3,''  exposures'')') NAME,NXP
        WRITE(6,'(''   ----------------'',12X,''------------'')')
        WRITE(7,'(''  TILT CORRECTED'',A20,I3,'' exposures'')')NAME,NXP
        WRITE(6,'(//''      Exposure Id.    '',''    Mean Vector'',
     +  ''  Consistency Ratio'')')
        DO 500 J=1,NXP
        READ(2,'(A20,I3,I4,I5)')IDXP,MDIP,MDIR,N
c
         DO 200 I=1,N
         READ(2,'(I2,1X,I3)') L,M
         IF(L.EQ.0) TAG(I)='T'
         IF(L.NE.0) TAG(I)='P'
         LDIP(I)=FLOAT(L)
         NDIR(I)=FLOAT(M)
         STAG(I)='1'
  200    CONTINUE
        CALL VMEAN(NDIR,N,RPHI,RVEC)
        WRITE(6,'(//,I3,''.'',A20,f8.0,10X,F4.0,''   %'')')J,IDXP,
     +  RPHI,RVEC
        IF(MDIP.EQ.0) GO TO 500
        DIP=FLOAT(MDIP)
        TWORS=FLOAT(MDIR)
        DO 300 I=1,N
         IF(TAG(I).EQ.'T')THEN
         CALL Q4S4(TWORS,NDIR(I),STAG(I))
         CALL FLSDIP(DIP,TWORS,NDIR(I),LDIP(I))
         ELSE
         CALL NORMAL(LDIP(I),NDIR(I))
         END IF
         GAMMA(I)=(90.-LDIP(I))*CONVRT
         NDIR(I)=NDIR(I)*CONVRT
         CALL TRNSF1(NDIR.I),GAMMA(I),LP(I),MP(I),NP(I))
  300    CONTINUE

        CALL FIXRAZ(DIP,TWORS,RAZ,ROT)
        CALL TILT(RAZ,ROT,LP,MP,NP,DC,N)
        CALL TRNSF3(DC,CONVRT,N,RAZEM,GAMMA)
        WRITE(7,'(A20,I3,I4,I5,''  **'',F4.0,''DEGREES ROTATION AROUND''
     +  ,F5.0)') IDXP,MDIP,MDIR,N,ROT,RAZ
        NRAZ=NINT(RAZ)
        DO 400 I=1,N
```

```
      IF(TAG(I).EQ.'P')THEN
      CALL NORMAL(LDIP(I),RAZEM(I))
      LAZEM=NINT(RAZEM(I))
      LUNGE=NINT(LDIP(I))
      ELSE
      LAZEM=NINT(RAZEM(I))
      NAZEM=NINT(NDIR(I)/CONVRT)
      LUNGE=NINT(LDIP(I))
      CALL Q4S(LAZEM,NAZEM,NRAZ,TWORS)
      IF(STAG(I).EQ.'1')THEN
      RAZEM(I)=FLOAT(LAZEM)
      ELSE
      RAZEM(I)=FLOAT(LAZEM)+180.
      IF(RAZEM(I).GT.360.) RAZEM(I)=RAZEM(I)-360.
      LAZEM=NINT(RAZEM(I))
      END IF
      END IF
      WRITE(7,'(I2,I4)') LUNGE,LAZEM
  400 CONTINUE
      CALL VMEAN(RAZEM,N,RPHI,RVEC)
      WRITE(6,'(''      TILT CORRECTED ** '',F9.0,10X,F5.0,''  %'')')
     + RPHI,RVEC
  500 CONTINUE
 1000 CONTINUE
      CLOSE(7)
      STOP
      END
c
      SUBROUTINE NORMAL(LD,ND)
c
c     returns perpendicular to a line with plunge = LD, in direction
c     ND, in degrees
c
      REAL LD,ND
      LD=90.-LD
      ND=ND+180.
      IF(ND.GT.360.)ND=ND-360.
      RETURN
      END
c
      SUBROUTINE TILT(AZEM,ROT,LP,MP,NP,DC,NVEC)
c     rotates poles around a horizontal axis
c     refer to original version in Computers & Geosciences v.13 p. 235-254
c
      REAL L,M,N,AZEM,ROT,A(3,3),LP(NVEC),MP(NVEC),NP(NVEC),
     + DC(3,NVEC)
c
c     this version takes direction cosines as input data
c
      CONVRT=ATAN(1.)/45.
c
c     compute direction cosines of poles
c
      AZ=AZEM*CONVRT
      L=COS(AZ)
      M=SIN(AZ)
      N=0.
      COSROT=COS(ROT*CONVRT)
      COSR1=COS((ROT-90.)*CONVRT)
      COSR2=COS((ROT+90.)*CONVRT)
c
c     following block is for a special rotation axis when
c     it coincides with the horizontal north; azimuth=0 gama=90
c
      IF(AZEM.EQ.0..OR.AZEM.EQ.360.)THEN
c

      DO 200 I=1,NVEC
      A1=LP(I)
      A2=MP(I)
      A3=NP(I)
      IF(ROT.EQ.90.)THEN
      DC(1,I)=A1
      DC(2,I)=-A3
      DC(3,I)=A2
      ELSE
      DC(1,I)=A1
      DC(2,I)=COSROT*A2+COSR1*A3
```

```
      DC(3,I)=COSR2*A2+COSROT*A3
      END IF
200   CONTINUE
c
c     following block is for when rotation axis coincides with
c     horizontal south azimuth=180 gama=90
c
      ELSE IF(AZEM.EQ.180.)THEN
c

      DO 250 I=1,NVEC
      A1=LP(I)
      A2=MP(I)
      A3=NP(I)
      IF(ROT.EQ.90.)THEN
      DC(1,I)=A1
      DC(2,I)=A3
      DC(3,I)=-A2
      ELSE
      DC(1,I)=A1
      DC(2,I)=COSROT*A2+COSR2*A3
      DC(3,I)=COSR1*A2+COSROT*A3
      END IF
250   CONTINUE
c
c     following block is for when rotation axis coincides with
c     the horizontal east( y-axis); azimuth=90 gama=90
c
      ELSE IF(AZEM.EQ.90.)THEN
c

      DO 300 I=1,NVEC
      A1=LP(I)
      A2=MP(I)
      A3=NP(I)
      IF(ROT.EQ.90.)THEN
      DC(1,I)=A3
      DC(2,I)=A2
      DC(3,I)=-A1
      ELSE
      DC(1,I)=COSROT*A1+COSR2*A3
      DC(2,I)=A2
      DC(3,I)=COSR1*A1+COSROT*A3
      END IF
300   CONTINUE
c
c     following block is for when rotation axis coincides with
c     horizontal west; azimuth=270 gama=90
c
      ELSE IF(AZEM.EQ.270.)THEN
c

      DO 350 I=1,NVEC
      A1=LP(I)
      A2=MP(I)
      A3=NP(I)
      IF(ROT.EQ.90.)THEN
      DC(1,I)=-A3
      DC(2,I)=A2
      DC(3,I)=A1
      ELSE
      DC(1,I)=COSROT*A1+COSR1*A3
      DC(2,I)=A2
      DC(3,I)=COSR2*A1+COSROT*A3
      END IF
350   CONTINUE
c
c     following block is for when the rotation axis is on the
c     primitive but the azimuth is not 0-180 nor 90-270
c
      ELSE
c
      COSTA=1.-COSROT
c
c     reorient x-axis
c
      BCOS=1.-(1.-L*L)*COSTA
      XLF=BCOS
```

d

```
      XMF=(L/M)*(1.-XLF)
      XNF=1.-XLF*XLF-XMF*XMF
      XNF=SQRT(XNF)
      X1NF=-XNF
      X1MF=XMF
      X1LF=XLF
c
c     reorient y-axis
c
      YMF=1.-(1.-M*M)*COSTA
      YLF=(M/L)*(1.-YMF)
      YNF=1.-YLF*YLF-YMF*YMF
      YNF=SQRT(YNF)
      Y1NF=-YNF
      Y1MF=YMF
      Y1LF=YLF
c
c     reorient z-axis
c
      ZNF=COSROT
      SINT=SIN(ROT*CONVRT)
      ZMF=SINT*L
      Z1MF=-ZMF
      Z1NF=ZNF
      ZLF=-SINT*M
      Z1LF=-ZLF
      NRAZ=NINT(AZEM)
c
      CALL KDELTA(XLF,X1LF,XMF,X1MF,XNF,X1NF,YLF,Y1LF,YMF,Y1MF,YNF,
     + Y1NF,ZLF,Z1LF,ZMF,Z1MF,ZNF,Z1NF,A,NRAZ)
c
      CALL TRNSF2(LP,MP,NP,A,DC,NVEC)
c
      END IF
c
c     convert upper hemisphere pole positions to lower hemisphere
c
      DO 800 I=1,NVEC
      IF(DC(3,I).GE.0.) GO TO 800
      DC(1,I)=-DC(1,I)
      DC(2,I)=-DC(2,I)
      DC(3,I)=-DC(3,I)
 800  CONTINUE
c
      RETURN
      END
c

      SUBROUTINE KDELTA(X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,
     + Z4,Z5,Z6,A,NRAZ)
c
c     A NEW VERSION OF KDELTA
c     Choose that set of (A(I,J),I=1,3).J=1,3) that
c     satisfies Kronecker delta
c     use only when the rotation axis is horizontal
c
      REAL X(6),Y(6),Z(6),AF(3,3),AAF(3,3),A(3,3),SUMNOW,SUMAGO
c
      NSET=0
      X(1)=X1
      X(2)=X2
      X(3)=X3
      X(4)=X4
      X(5)=X5
      X(6)=X6
      Y(1)=Y1
      Y(2)=Y2
      Y(3)=Y3
      Y(4)=Y4
      Y(5)=Y5
      Y(6)=Y6
      Z(1)=Z1
      Z(2)=Z2
      Z(3)=Z3
      Z(4)=Z4
      Z(5)=Z5
      Z(6)=Z6
c
```

```
      SUMAGO=0.0001
c
      DO 1000 JX=1,2
        A1=X(JX)
        B1=X(JX+2)
        C1=X(JX+4)
        DO 900 JY=1,2
          A2=Y(JY)
          B2=Y(JY+2)
          C2=Y(JY+4)
C
          C3=Z(6)
C
          DO 800 JZ=1,4
            IF(JZ.EQ.1)THEN
            B3=Z(3)
            A3=Z(1)
            ELSE IF(JZ.EQ.2)THEN
            B3=Z(4)
            A3=Z(1)
            ELSE IF(JZ.EQ.3)THEN
            B3=Z(3)
            A3=Z(2)
            ELSE
            B3=Z(4)
            A3=Z(2)
            END IF
c
                      S1=ABS(A1*B1+A2*B2+A3*B3)
                      S2=ABS(A1*C1+A2*C2+A3*C3)
                      S3=ABS(B1*C1+B2*C2+B3*C3)
                      S4=ABS(A1*A2+B1*B2+C1*C2)
                      S5=ABS(A1*A3+B1*B3+C1*C3)
                      S6=ABS(A2*A3+B2*B3+C2*C3)
c
                      SUMNOW=S1+S2+S3+S4+S5+S6
                      IF(SUMNOW.GE.SUMAGO) GO TO 800
                      NSET=NSET+1
                      IF(NSET.EQ.1)THEN
                      AF(1,1)=A1
                      AF(2,1)=B1
                      AF(3,1)=C1
                      AF(1,2)=A2
                      AF(2,2)=B2
                      AF(3,2)=C2
                      AF(1,3)=A3
                      AF(2,3)=B3
                      AF(3,3)=C3
c       the above set entails anticlockwise rotation of poles if
c       rotation axis azimuth (NRAZ) is .LE.180
c
                      ELSE
                      AAF(1,1)=A1
                      AAF(2,1)=B1
                      AAF(3,1)=C1
                      AAF(1,2)=A2
                      AAF(2,2)=B2
                      AAF(3,2)=C2
                      AAF(1,3)=A3
                      AAF(2,3)=B3
                      AAF(3,3)=C3
c       above alternative set entails anticlockwise rotation
c       of poles if rotation axis azimuth (NRAZ) is .GT.180
c
                      END IF
  800            CONTINUE
  900          CONTINUE
 1000        CONTINUE
      IF(NRAZ.GT.180)THEN
      A(1,1)=AF(1,1)
      A(2,1)=AF(2,1)
      A(3,1)=AF(3,1)
      A(1,2)=AF(1,2)
      A(2,2)=AF(2,2)
      A(3,2)=AF(3,2)
      A(1,3)=AF(1,3)
      A(2,3)=AF(2,3)
      A(3,3)=AF(3,3)
```

e

```
      ELSE
      A(1,1)=AAF(1,1)
      A(2,1)=AAF(2,1)
      A(3,1)=AAF(3,1)
      A(1,2)=AAF(1,2)
      A(2,2)=AAF(2,2)
      A(3,2)=AAF(3,2)
      A(1,3)=AAF(1,3)
      A(2,3)=AAF(2,3)
      A(3,3)=AAF(3,3)
      END IF
      RETURN
      END
c
      SUBROUTINE TRNSF1(AZEM,GAMA,A1,A2,A3)
c
c     converts spherical polar coordinates(1,gama,azem) to
c     rectangular cartesian coordinates(a1,a2,a3/x,y,z)
c     azimuth is measured clockwise positive from horizontal
c     north; north is +x, east is +y, downwards vertical +z
c     AZEM, GAMA in radians; GAMA = PI/2 - PLUNGE
c     direction cosines are equal to (x,y,z) coordinates
c     for unit vectors
c
      P1=SIN(GAMA)
      P2=COS(GAMA)
      P3=SIN(AZEM)
      P4=COS(AZEM)
      A1=P1*P4
      A2=P1*P3
      A3=P2
      RETURN
      END
c
      SUBROUTINE TRNSF2(LP,MP,NP,A,DC,NPTS)
c
c     transforms coordinates of a set of poles knowing the initial
c     and the final orientations of the coordinate frame; a finite
c     rotation around any line is the greometric operation
c     transformation matrix is in array A(3,3)
c     DC(3,NPTS) contains the direction cosines of the rotated poles
c     AZ(NPTS) and G(NPTS) contain azimuth and gama values for
c     the initial orientation of poles; AZ and G in radians
c
      REAL LP(NPTS),MP(NPTS),NP(NPTS),A(3,3),DC(3,NPTS)
      DO 100 I=1,NPTS
      X=LP(I)
      Y=MP(I)
      Z=NP(I)
c
      DC(1,I)=A(1,1)*X+A(2,1)*Y+A(3,1)*Z
      DC(2,I)=A(1,2)*X+A(2,2)*Y+A(3,2)*Z
      DC(3,I)=A(1,3)*X+A(2,3)*Y+A(3,3)*Z
  100 CONTINUE
      RETURN
      END
c
      SUBROUTINE TRNSF3(DC,CONVRT,NPTS,AZEM,PLUNGE)
c
c     converts direction cosines to azimuth and plunge
c     GAMA = PI/2 - PLUNGE
c
      REAL DC(3,NPTS),AZEM(NPTS),PLUNGE(NPTS)
c
      DO 300 I=1,NPTS
      A=DC(1,I)
      B=DC(2,I)
      C=DC(3,I)
      IF(C.GE.0.) GO TO 100
      A=-A
      B=-B
      C=-C
  100 CONTINUE
      GAMA=ACOS(C)
      IF(GAMA.GT..1E-03.AND.GAMA.LT.(180.*CONVRT-.1E-03))THEN
      BADDY=A/SIN(GAMA)
c     WRITE(6,'(////'' WHY STOPPING'',5X, E10.4)') BADDY
```

```
      DX=BADDY-1.
      IF(DX.GT..1E-09) STOP
      IF(DX.GT.0..AND.DX.LT..1E-09) AZEM(I)=0.
      IF(DX.LE.0.) AZEM(I)=ACOS(BADDY)
      IF(B.LT.0.)AZEM(I)=360.*CONVRT-AZEM(I)
      ELSE
      AZEM(I)=0.
      END IF
      AZEM(I)=AZEM(I)/CONVRT
      PLUNGE(I)=90.-(GAMA/CONVRT)
  300 CONTINUE
      RETURN
      END
c
      SUBROUTINE FIXRAZ(FMDIP,DP2RS,RAZ,ROT)
c
c     returns rotation axis azimuth (RAZ) as that end of bed strike
c     whose choice ensures anticlockwise rotation through
c     an angle (ROT) equal to formation dip amount (< = 90 degrees)
c     the dip amount and directions are (FMDIP) and (DP2RS)
c
      IF(DP2RS.EQ.360..OR.DP2RS.EQ.0.)THEN
      RAZ=270.
      ELSE IF(DP2RS.EQ.180.)THEN
      RAZ=90.
      ELSE IF(DP2RS.LE.90.)THEN
      RAZ=DP2RS+270.
      ELSE
      RAZ=DP2RS-90.
      END IF
      ROT=FMDIP
      RETURN
      END
c
      SUBROUTINE Q4S(AZM,NAZ,RAZ,DP2RS)
c
c     forces rotated pole azimuth to a sector which is
c     within plus or minus 90 degrees of formation dip
c     direction; this is necessary for SPIN8 was written
c     for handling axial data only
c
c     also takes care of situtations where the original
c     vector coincides with the rotation axis
c
      LOGICAL DECIDE,TEST
      INTEGER AZM,RAZ
      TEST=.TRUE.
       MAZ=ABS(NAZ-RAZ)
       DECIDE=NAZ.EQ.RAZ.OR.MAZ.EQ.360.OR.MAZ.EQ.180
       IF(DECIDE.EQV.TEST)THEN
       AZM=NAZ
       ELSE
       NPASS=ABS(AZM-NINT(DP2RS))
       IF(NPASS.GT.90.AND.NPASS.LT.270) AZM=AZM+180
       END IF
      IF(AZM.GT.360)AZM=AZM-360
      RETURN
      END
c
      SUBROUTINE FLSDIP(TRUDP,DIPDR,ADIR,FLSDP)
c
c    returns apparent dip amount (FLSDP) along the
c    direction (ADIR) on a bed whose true dip direction (DIPDR)
c    and amount (TRUDP) are known
c
      BDSTRK=DIPDR+90.
      IF(BDSTRK.GT.180.) BDSTRK=BDSTRK-180.
      ALPHA=BDSTRK-ADIR
      IF(ALPHA.GE.0.) GO TO 10
      PALPHA=ABS(ALPHA)
      IF(PALPHA.LE.180.) ALPHA=PALPHA
      IF(PALPHA.GT.180.) ALPHA=ALPHA+360.
   10 CONVRT=ATAN(1.)/45.
      PHITAN=TAN(TRUDP*CONVRT)*SIN(ALPHA*CONVRT)
      FLSDP=ATAN(PHITAN)/CONVRT
      FLSDP=FLOAT(NINT(FLSDP))
      RETURN
      END
```

```
c
      SUBROUTINE VMEAN(AZEM,NVEC,DIR,RL)
c
c     returns vector mean for 2-dimensional case
c     uses the formulae       xsum = summation<cosine(azem)>
c                             ysum = summation<sine(azem)>
c     vector mean direction  dir = arctan(ysum/xsum)
c     vector mean magnitude is normalized to obtain
c     consistency ratio        rl = sqrt(xsum**2 + ysum**2)/nvec
c                             nvec = n. of vectors
c
c     vector azimuth (azem) 1-360 measured from north (360 or 0)
c                                           in a clockwise sense
c     +X axis coincides with azem = 360 or 0
c     +Y axis coincides with azem =  90
c     program will fail if   xsum = 0
c
      REAL AZEM(NVEC),DIR,RL
      CONVRT=ATAN(1.)/45.
      XSUM=0.
      YSUM=0.
c
      DO 10 I=1,NVEC
      AZ=AZEM(I)*CONVRT
      CAZ=COS(AZ)
      SAZ=SIN(AZ)
      XSUM=XSUM+CAZ
      YSUM=YSUM+SAZ
   10 CONTINUE
c
c     if blocks to take care of the limitations of the intrinsic
c     function y = arctan(x)    -90 < Y < +90, Y = y/convrt
c     also of situations where xsum.eq.0.and.ysum.ne.0
c
      IF(XSUM.EQ.0.)THEN
      DIR=90.
      IF(YSUM.LT.0.) DIR=270.
      ELSE IF(XSUM.GT.0.)THEN
      TANDIR=ABS(YSUM)/XSUM
      DIR=ATAN(TANDIR)/CONVRT
      IF(YSUM.LT.0.) DIR=360.-DIR
      ELSE IF(YSUM.EQ.0.AND.XSUM.GT.0.)THEN
      DIR=0.
      ELSE
      TANDIR=ABS(YSUM)/ABS(XSUM)
      DIR1=ATAN(TANDIR)/CONVRT
      DIR=180.-DIR1
      IF(YSUM.LT.0.) DIR=180.+DIR1
      END IF
c
      RL=SQRT(XSUM*XSUM+YSUM*YSUM)/FLOAT(NVEC)
      RL=RL*100.
      RETURN
      END
c
      SUBROUTINE Q4S4(DP2RS,DIR,TAGS)
c
      CHARACTER*1 TAGS
      REAL DP2RS,DIR
      DIFF=ABS(DP2RS-DIR)
      IF(DIFF.LT.90..OR.DIFF.GE.270.) RETURN
      DIR=DIR+180.
      IF(DIR.GT.360.)DIR=DIR-360.
      TAGS='0'
      RETURN
      END
```