Designation: E 1489 – 98

AMERICAN SOCIETY FOR TESTING AND MATERIALS
100 Barr Harbor Dr., West Conshohocken, PA 19428
Reprinted from the Annual Book of ASTM Standards. Copyright ASTM

Standard Practice for Computing Ride Number of Roads from Longitudinal Profile Measurements Made by an Inertial Profile Measuring Device[1]

This standard is issued under the fixed designation E 1489; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ϵ) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This practice covers the mathematical processing of longitudinal profile measurements to produce an estimate of subjective ride quality, termed Ride Number (RN).

1.2 The intent of this practice is to provide the highway community a standard practice for the computing and reporting of an estimate of subjective ride quality for highway pavements.

1.3 This practice is based on an algorithm developed in National Cooperative Highway Research Project (NCHRP) 1–23 **(1** and **2)**,[2] two Ohio Department of Transportation ride quality research projects (**3** and **4**), and work presented in Refs. **(5** and **6)**.

1.4 The computed estimate of subjective ride quality produced by this practice was named Ride Number RN in NCHRP Research Project 1–23 (**1** and **2**) to differentiate it from other measures of ride quality computed from longitudinal profile. Eq 1 of Section 8.2 represents the mathematical definition of Ride Number.

1.5 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.*

2. Referenced Documents

2.1 *ASTM Standards:*

E 177 Practice for Use of the Terms Precision and Bias in ASTM Test Methods[3]

E 867 Terminology Related to Traveled Surface Characteristics[4]

E 950 Test Method for Measuring the Longitudinal Profile of Traveled Surfaces With an Accelerometer Established Inertial Profiling Reference[4]

E 1170 Standard Practice for Simulating Vehicular Response to Longitudinal Profiles of a Vehicular Traveled Surface.

E 1364 Standard Test Method for Measuring Road Roughness by Static Rod and Level Method.

E 1500 Practice for Computing Mean Square Numerics from Road Surface Elevation Profile Records[4]

E 1656 Guide for Classification of Automated Pavement Condition Survey Equipment[4]

E 1927 Standard Guide for Conducting Subjective Pavement Ride Quality Ratings.

3. Terminology

3.1 *Terminology* used in this standard conforms to the definitions included in Terminology E 867.

3.2 *Definitions of Terms:*

3.2.1 *Rideability Index (RI)*—an index derived from controlled measurements of longitudinal profile in the wheel tracks and correlated with panel ratings of rideability.

3.2.2 *Ride Number (RN)*—rideability index of a pavement using a scale of 0 to 5, with 5 being perfect and 0 being impassable.

4. Summary of Practice

4.1 The practice presented here was developed specifically for estimating subjective ride quality from longitudinal profile measurements.

4.2 This practice uses longitudinal profile measurements for two wheel tracks as an input to a mathematical computation of estimated subjective ride quality (RN). The profile must be represented as a series of elevation values take at constant intervals along the wheel tracks.

4.3 The range of the computed subjective ride quality estimate is 0 to 5.0 Ride Number (RN) units where an RN of 5.0 is considered to be a perfect ride quality road. The 0 to 5.0 Ride Number scale is defined in Refs **(1, 2, 3, 4**, and **5)**.

4.3.1 In the 0 to 5.0 Ride Number rating scale, the end points and some of the intermediate points have the following descriptions:

Ride Number Rating Scale

Description	Ride Number
Perfect	5.0
Very good	4.5

[1] This practice is under the jurisdiction of ASTM Committee E-17 on Vehicle-Pavement Systems and is the direct responsibility of Subcommittee E17.33 on Methodology for Analyzing Pavement Roughness.

Current edition approved January 10, 1998. Published March 1999. Originally published as E 1489–96. Last previous edition E 1489–96.

[2] The boldface numbers in parentheses refer to the list of references at the end of this standard.

[3] *Annual Book of ASTM Standards*, Vol 14.02.

[4] *Annual Book of ASTM Standards*, Vol 04.03.

	4.0
Good	3.5
	3.0
Fair	2.5
	2.0
Poor	1.5
	1.0
Very poor	0.5
Impassable	0.0

4.3.2 The end points are further defined by the following descriptions **(1)**:

Perfect: A road which is so smooth that at the speed you are traveling you would hardly know the road was there. You doubt that if someone made the surface smoother that the ride would be detectably nicer.

Impassable: A road which is so bad that you doubt that you or the car will make it to the end at the speed you are traveling—like driving down railroad tracks along the ties.

4.4 The quality of the computed subjective ride quality estimate (RN) produced by this practice is based on the processing of the longitudinal profile as measured with a road profile measuring device that meets the Class 1 requirements of ASTM Standard E 950.

NOTE 1—Less accurate Ride Number values will result from Road Profile Data obtained from Profile Measuring Devices that are less accurate than class I (E 950).

5. Significance and Use

5.1 This practice provides a means for obtaining a quantitative estimate of a pavement property defined as ride quality or rideability using longitudinal profile measuring equipment.

5.1.1 The Ride Number (RN) is portable because it can be obtained from longitudinal profiles obtained with a variety of instruments.

5.1.2 The RN is stable with time because true RN is based on the concept of a true longitudinal profile, rather than the physical properties of particular type of instrument.

5.2 Ride quality information is a useful input to the pavement manage systems (PMS) maintained by transportation agencies.

5.2.1 The subjective ride quality estimate produced by this practice has been determined **(6)** to be highly correlated (r = 0.92) with measured subjective ride quality and to produce a low standard estimate of error (0.29 RN units) for the ride quality estimate.

5.2.2 The subjective ride quality estimates produced by this practice were found to be not significantly different with respect to pavement type, road class, vehicle size, vehicle speed (within posted speed limits), and regionality over the range of variables included in the experiment **(1, 2, 3**, and **4)**.

5.2.3 The subjective ride quality estimates produced by this practice have been found to be good predictors of the need of non-routine road maintenance for the various road classifications **(3)**.

5.3 The use of this practice to produce subjective ride quality estimates from measured longitudinal profile eliminates the need for expensive ride panel studies to obtain the same ride quality information.

6. Longitudinal Profile Measurement

6.1 The elevation profile data used in this practice must have sufficient accuracy to measure the longitudinal profile attributes that are essential to the computation of estimated subjective ride quality.

6.1.1 The quality of the Ride Number estimates cited in this practice are based on the use of elevation profile measurements made with a Class 1 road profile measuring device as defined in Test Method E 950.

6.1.2 *Wave Length Content*—The measured longitudinal profile used as input to this practice must have the wavelength content required for the application.

6.1.2.1 As a guide to the wavelength requirement, a repeating sine wave of the following wavelengths and peak-to-peak amplitudes in the absence of any other roughness will produce the following RN values:

Amplitude mm	Wavelength m	RN
25.4	91.4	4.95

Amplitude inch	Wavelength feet	RN
1.00	300	4.95

6.1.2.2 The quality of Ride Number estimates cited in this practice are based on measured longitudinal profile with wavelength content up to 91.4 m (300 feet).

7. Precision and Bias

7.1 *Precision and Bias*—The accuracy of the computed subjective ride quality estimate produced by this practice will be a function of the accuracy of the longitudinal profile measurements.

7.1.1 *Correlation*—The ride quality estimates (RN) computed by this practice have been determined to have a correlation coefficient of .92 (r) with actual measured subjective ride quality **(3, 4, 5,** and **6)**.

7.2 *Standard Error of Estimate*—The ride quality estimates (RN) computed by this practice have been determined to have a Standard Error of Estimate of .29 RN units when compared to actual measured subjective ride quality **(3, 4, 5,** and **6)**.

7.3 The Correlation Coefficient and Standard Error of Estimate values cited in Sections 7.1 and 7.2 are based on longitudinal profile measurements made with a road profile measuring device that meets the requirements of a Class 1 measuring device as defined by ASTM Standard E 950 and wavelength content up to 100 m (300 feet).

7.4 It is not known how road profile measuring equipment with lesser resolution and precision, and greater bias would affect the accuracy of computed ride numbers.

8. Ride Number Program

8.1 This practice consists of the computation of Ride Number (RN) from an algorithm developed in National Cooperative Highway Research Project (NCHRP) 1–23 (**1** and **2**), two Ohio DOT ride quality research projects (**3** and **4**), and the work presented in Refs **(5)** and **(6)**.

8.2 Ride Number is defined in this practice by the equation:

$$RN = 5e^{-160(PI)} \tag{1}$$

where:

$$PI = \sqrt{\frac{PI_L^2 + PI_R^2}{2}} \quad (2)$$

and where:

PI_L and PI_R are Profile Indexes for the left and right wheel paths, respectively, and are the computed root mean square (RMS) of the filtered slopes of the measured elevation profiles of the individual right and left wheel paths **(6)**. The wave length components of the profile slopes are modified by the filter shown in Fig. 7 **(6)**.

8.3 A FORTRAN computer version of this algorithm has been implemented as described in Ref **(6)**.

8.3.1 This practice presents a sample computer program" RNSMP" for the computation of the Ride Number equation from the recorded longitudinal profile measurement.

8.3.1.1 The computer program RNSMP is a general computer program that accepts the elevation profile data set as input, and then calculates the Ride Number using the equation presented in 8.2.

8.3.1.2 A listing of the RNSMP computer program for the computation of the Ride Number transform equation is included in this practice as Appendix X1.

8.3.1.3 A provision has been made in the computer program listing (Appendix X1) for the computation of the Ride Number transform equation from recorded longitudinal profile measurements in both SI and inch-pound units.

8.3.2 The input to the sample Ride Number computer program is an ASCII profile data set stored in a 1X, F8.3, 1X, F8.3 Fortran format. In this format, the profile data appears as a multi-row, two-column array with the left wheel path profile data points in column 1 and the right wheel path points in column 2. The profile data point interval is discretionary. However, the quality of the Ride Number estimates cited in this practice, are based on a data point interval of 150 mm (6 in.) (see Section 5).

8.3.2.1 If the input to the Ride Number computer program is in SI units, the elevation profile data points are scaled in millimetres with the least significant digit being equal to 0.001 mm.

8.3.2.2 If the input to the Ride Number computer program is in inch-pound units, the elevation profile data points are scaled in inches with the least significant digit being equal to 0.001 inch.

8.4 The distance interval over which the Ride Number is computed is discretionary, but shall be reported along with the Ride Number results.

8.5 Validation of the Ride Number program is required when it is installed. Provisions for the RN program installation validation have been provided in this practice.

8.5.1 The sample profile data set TRIPULSE.ASC has been provided in SI units in Appendix X2 for validation of the computer program installation.

8.5.2 Using the sample profile data set TRIPULSE.ASC (Appendix X2) as input to the Ride Number computer program (Appendix X1), a Ride Number of 3.66 Ride Number units was computed as shown in Appendix X3 for a profile data point interval of .15 m (.5 feet) and a distance interval equal to 15 meters of the profile data set.

9. Report

9.1 The report for this practice shall contain the following information:

9.1.1 *Profile Measuring Device*—The report data shall include the ASTM Standard E 950 classification of the device used to make the measurements, the date of the last successful device calibration and the highpass filter wavelength used in the profile measurement.

9.1.2 *Longitudinal Profile Measurements*—Report data from the profile measuring process shall include the date and time of day of the measurement, the location of the measurement, the lane measured, the direction of the measurement, length of measurement, and the descriptions of the beginning and ending points of the measurement.

9.1.3 *Ride Number Results*—The Ride Number results should be reported to two decimal places along with the distance interval over which the RN was computed.

10. Keywords

10.1 longitudinal profile; mean panel ratings (MPR); panel rating; pavement ride quality; profile index; rideability; ride number (RN); subjective ride quality; subjective ride quality estimate

APPENDIXES

(Nonmandatory Information)

X1. RIDE NUMBER COMPUTER PROGRAM

X1.1 Included as Fig. X1.1 is a sample Fortran computer program using user-selected SI or inch-pound units that can be used to compute the Ride Number specified by this practice. If the SI option is selected, the program assumes the input road profile amplitudes are stored in millimetre units, if inch-pound, inches. In the sample program, the profile data sample interval is assumed to be either .15 m or .5 feet depending on the option selected. For the sample program, the maximum road section length that can be processed is 160.9 meters (528 feet).

X1.2 The elevations in the sample data file shown in Appendix X2 are in SI units (mm) and contain one thousand profile data points. If the inch-pound option is selected, the user must convert the profile data set from SI units to inch-pound units to get the inch-pound output shown in Appendix X3.

```
c================================================

c      1.lnk:
c       rnsmp+               ('MAIN' program)
c         ;
c
c    Sample RN Fortran Computer Program
c========================================
c- Sample program to read a data file containing two tracks of
c- road profile elevation data into a "DATA" array, call SUBR RN
c- and print a final report of the Ride Number index (RN).
c- If the input profile data is in English units, the elevation values
c- are converted from inch to mm units and the sampling interval, from
c- feet to meters. RN ranges from 0.0 to 5.0 with 5.0 being 'perfect'.
c-
c- (SUBROUTINE GETPI is called to perform the RN computation as
c- prescribed by this practice.)
c========================================
c
      PROGRAM RNSMP
c
c---------------------------------
c
c      DELT ---------------> DX
c      DATA(1, 1058) ------> PROF(NPTS), left track
c      DATA(2, 1058) ------> PROF(NPTS), right track
c      RNL ----------------> RN, left track
c      RNR ----------------> RN, right track
c      RN -----------------> == SQRT(RNL**2 + RNR**2)/2.
c
c========================================

      REAL     DATA(2, 1058), DELT, ANSWER
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index.

```
      REAL      DINTVL, SECLEN
      REAL      BASE, PROF(1058)
      REAL      PIL, PIR, PI, RNL, RNR, RN
c
      BYTE      UNITS
      CHARACTER   KNAME*12
c
      INTEGER        NREC, I, J, NPTS
c-----------------------------------
c
      NREC = 0
      BASE = 0.
c
      WRITE (*, 1000)
1000  FORMAT (/'1Enter Data File Name (in single quotes)'/
     1 '       (''TRIPULSE.ASC'' in example):    '$)
c
      READ(*,*)KNAME(1:12)
c
      WRITE(*,1010)
1010  FORMAT(/' Is road profile data in SI or inch-pound units?'/
     1 '        Type ''S'' or ''I'', (cr)    :    '$)
c
      READ(*,1020)UNITS
1020  FORMAT(A1)
c
      IF(UNITS.EQ.'I'.OR.UNITS.EQ.'i') WRITE(*,1030)
1030  FORMAT(/' Enter length of section, feet '/
     1 '        (49 ft in example)          :    '$)
c
      IF(UNITS.EQ.'S'.OR.UNITS.EQ.'s') WRITE(*,1040)
1040  FORMAT(/' Enter length of section, meters '/
     1 '        (15 m in example)           :    '$)
c
      READ(*,1050)SECLEN
1050  FORMAT(F10.0)
c
      IF(UNITS.EQ.'I'.OR.UNITS.EQ.'i') DELT = .5    ! ft
      IF(UNITS.EQ.'S'.OR.UNITS.EQ.'s') DELT =.15    ! m
c
      IF(UNITS.EQ.'I'.OR.UNITS.EQ.'i')SECLEN=SECLEN/3.281  !ft --> m
c
      NPTS=SECLEN/DELT      !number of elevation pairs in DATA array.
c
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
c-----------------------------
      WRITE(*,1075)
1075  FORMAT(/' Is Input Profile Pre-Smoothed (Y or N) ? '$)
      READ(*,1020)ANSWER
      BASE=.250                      ! meters
      IF(ANSWER .NE. 'N' .AND. ANSWER .NE. 'n')BASE = 0.0
c
c-----------------------------
c- Open input file and read profile elevations
c  into 'DATA' array:
c
      OPEN(UNIT=2,FILE=KNAME(1:12),FORM='FORMATTED')
c
c-----------------------------
c
      DO 20 I = 1,NPTS
c
      READ(2,1080)(DATA(J, I), J = 1, 2)    ! read 'NPTS' data pairs
c                                       into DATA array
1080  FORMAT(2(1X, F8.3))
c
20    CONTINUE
c-----------------------------------
c- Call subroutine GETPI to compute Ride Number index:
c
c  Copy left wheelpath profile to 'PROF' array:
c
      DO 30 I = 1, NPTS
      IF(UNITS .EQ. 'S' .OR. UNITS .EQ. 's')
     & PROF(I) = DATA(1, I)/1000.        ! mm --> m
      IF(UNITS .EQ. 'I' .OR. UNITS .EQ. 'i')
     & PROF(I) = DATA(1,I)*.0254         ! in --> m
30    CONTINUE

c
c-------------------------------------------------------
      CALL GETPI (PROF, NPTS, DELT, BASE, PIL)
c-------------------------------------------------------

c
c  Copy right wheelpath profile to PROF array:
c
c
      NPTS = SECLEN/DELT       !number of elevation pairs in DATA array.
c
      DO 40 I = 1, NPTS
      IF(UNITS .EQ. 'S'.OR. UNITS .EQ. 's')
     & PROF(I) = DATA(2, I)/1000.         ! mm --> m
      IF(UNITS .EQ. 'I'.OR. UNITS .EQ. 'i')
     & PROF(I) = DATA(2, I)*.0254         ! in --> m
c
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
40    CONTINUE
c
c-----------------------------------------------------
      CALL GETPI (PROF, NPTS, DELT, BASE, PIR)
c-----------------------------------------------------
c
c  Transform PIs to Ride Numbers:
c
      RNL = 5.*EXP(-160.*PIL)
      RNR = 5.*EXP(-160.*PIR)
c
      PI = SQRT((PIL**2 + PIR**2)/2.)
      RN = 5.*EXP(-160.*PI)
c
      DINTVL = NPTS*DELT            ! length of profile section
c
c-----------------------------------
c- Output computed ASTM Ride Number:
c
      IF(UNITS .EQ. 'S'.OR. UNITS .EQ. 's')GOTO 200
c
      WRITE(*, 2010)
     1 RNL, RNR, RN, DINTVL*3.281
2010  FORMAT(//////
     1              6X' RN, left                        = ',F10.2//
     2              6X' RN, right                       = ',F10.2////
     3              6X' ASTM Ride Number (RN)           = ',F10.2//
     4            6X' Distance    =  ',F6.1,' feet'/)
c
      GOTO 210
c--------------------------------------
c
200   CONTINUE
c
      WRITE(*, 2020)
     1 RNL, RNR, RN, DINTVL
2020  FORMAT(//////
     1              6X' RN, left                        = ',F10.2 //
     2              6X' RN, right                       = ',F10.2////
     3              6X' Ride Number (RN)                = ',F10.2//
     4            6X' Distance    =  ',F6.1,' meters'/)
c
c-----------------------------------
210   CONTINUE
c
      END    ! of program RNSMP
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
c        Listing 1. Fortran code to calculate RN from profile.
C=============================================================
      SUBROUTINE GETPI (PROF, NSAMP, DX, BASE, PI)

Fig. X1.1. Sample Fortran Program Using Subroutine GETPI
           to Compute Ride Number Index (con.).
   C  Filter a longitudinal road profile and calculate RN.

   C  <-> PROF      REAL   On input, an array of profile height values (m).
   C                          On output, an array of filtered profile
   C                          values.
   C  <-> NSAMP    INTEGER   Number of data values in array PROF. (The
   C                      filtered profile always has fewer points
   C                      than the original).
   C  --> DX       REAL       Distance step between profile points (m).
   C  --> BASE     REAL       Distance covered by moving average (m).
   C                          Use .250 for unfiltered profile input,
   C                          and 0.0 for pre-smoothed profiies
   C                          (e.g. K.J. Law data).
   C  <-- PI       REAL       The Profile Index for the entire profile.
   C  <-- K1, K2, C, MU       Filter coefficients.
   C
   C-------------------------------------------
         INTEGER      I, Ill, IBASE, NSAMP
         REAL    AMAT, BMAT, BASE, CMAT, DX, PI, PR
         REAL    PROF, SFPI, ST, XIN
         DIMENSION  AMAT(4, 4), BMAT(4), CMAT(4), PR(4),
        &                   PROF(NSAMP+2), ST(4, 4), XIN(4)
   C-------------------------------------------
   C  Set parameters and arrays.
   c- RN coefs:
         CALL SETABC(5120., 390., 17., .036, AMAT, BMAT, CMAT)
         CALL SETSTM(DX/(80./3.6), AMAT, BMAT, ST, PR)
   c
   C  Compute PI from filtered profile.
   c
         IBASE = MAX(NINT(BASE/DX), 1)
         SFPI = 1.0/(DX*IBASE)
   C-------------------------------------------
   C  Initialize simulation variables based on profile start.

         Ill     = MIN(NINT(19./DX), NSAMP)
         XIN(1) = (PROF(Ill) - PROF(1))/(DX*Ill)
         XIN(2) = 0.0
         XIN(3) = XIN(1)
         XIN(4) = 0.0

   C  Convert to averaged slope profile.
         NSAMP = NSAMP - IBASE
         DO 10 I = 1, NSAMP
      10   PROF(I) = SFPI*(PROF(I + IBASE) - PROF(I))

   C  Filter profile.
   c
         CALL STFILT.PROF, NSAMP, ST, PR, CMAT, XIN)

   c
   C  Compute PI from filtered profile.
         PI = 0.0
         DO 20 I = 1, NSAMP
    20    PI = PI + PROF(I)**2
         PI = SQRT(PI/NSAMP)
   c
         RETURN
         END        ! of SUBR GETPI
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
c            Listing 2. Code to set model matrices.
C===============================================================
      SUBROUTINE SETABC(K1, K2, C, MU, AMAT, BMAT, CMAT)
C===============================================================
C  Set the A, B and C matrices filter.

C  -->  K1   REAL   Filter coefficients
C  -->  K2   REAL     "        "
C  -->  C    REAL     "        "
C  -->  MU   REAL     "        "
C  <--  AMAT REAL   The 4x4 A matrix.
C  <--  BMAT REAL   The 4x1 B matrix.
C  <--  CMAT REAL   The 4x1 C matrix.

C---------------------------------------------
      INTEGER         I, J
      REAL            AMAT, BMAT, CMAT, K1, K2, C, MU
      DIMENSION       AMAT(4, 4), BMAT(4), CMAT(4)

C---------------------------------------------
C  Set default for all matrix elements to zero.
      DO 10  J = 1, 4
        BMAT(J) = 0
        CMAT(J) = 0
        DO 10  I = 1, 4
10        AMAT(I, J) = 0
C  Put filter coefficients into the A Matrix.
      AMAT(1, 2) = 1.
      AMAT(3, 4) = 1.
      AMAT(2, 1) = -K2
      AMAT(2, 2) = -C
      AMAT(2, 3) = K2
      AMAT(2, 4) = C
      AMAT(4, 1) = K2/MU
      AMAT(4, 2) = C/MU
      AMAT(4, 3) = -(K1 + K2)/MU
      AMAT(4, 4) - -C/MU

C  Set the B matrix.
      BMAT(4) = K1/MU

C  Set the C matrix to use filtered profile slope as output.
      CMAT(1) = -1
      CMAT(3) = 1

      RETURN
      END
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
c          Listing 3. Code to set state transition matrix.
C=============================================================
      SUBROUTINE SETSTM(DT, A, B, ST, PR)
C=============================================================

C  Compute ST and PR arrays.  This requires INVERT for matrix inversion.

C  -->  DT   REAL  Time step (sec)
C  -->       A    REAL    The 4x4 A matrix.
C  -->       B    REAL    The 4x4 B matrix.
C  <--  ST   REAL         4x4 state transition matrix.
C  <--  PR   REAL         4x1 partial response vector.

C-----------------------------------
      INTEGER       I, ITER, J, K
      LOGICAL       MORE
      REAL          A, A1, A2, B, DT, PR, ST, TEMP
      DIMENSION     A(4, 4), A1(4, 4), A2(4, 4), B(4)
      DIMENSION     PR(4),ST(4, 4), TEMP(4, 4)
C------------------------------
      DO 20 J = 1,4
        DO 10 I = 1,4
          A1(I, J) = 0
10        ST(I, J) = 0
        A1(J, J) = 1.
20      ST(J, J) = 1.

C  Calculate the state transition matrix ST = exp(dt*A) with a Taylor
C  series. A1 is the previous term, A2 is the next one.
      ITER = 0
   30 ITER = ITER + 1
      MORE = .FALSE.
      DO 40 J = 1,4
        DO 40 I = 1,4
          A2(I, J) = 0
          DO 40 K = 1,4
   40       A2(I, J) = A2(I, J) + A1(I, K) * A(K, J)
      DO 50 J = 1,4
        DO 50 I = 1,4
          A1(I, J) = A2(I, J)*DT/ITER
          IF (ST(I, J) + A1(I, J) .NE. ST(I, J)) MORE = .TRUE.
   50     ST(I, J)= ST(I, J) + A1(I, J)
       IF (MORE) GO TO 30

C  Calculate particular response matrix: PR = A**-1*(ST-I)*B
      CALL INVERT(A, 4)
      DO 60 I = 1,4
        PR(I) = 0.0
        DO 60 K = 1,4
   60     PR(I) = PR(I) - A(I, K)*B(K)
      DO 90 J = 1,4
        DO 70 I = 1, 4
          TEMP(J, I) = 0.0
          DO 70 K = 1,4
   70       TEMP(J, I) = TEMP(J, I) + A(J, K)*ST(K, I)
        DO 80 K = 1, 4
   80     PR(J) = PR(J) + TEMP(J, K)*B(K)
   90   CONTINUE
      RETURN
      END
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
c             Listing 4. Code to filter profile.
C=============================================================
      SUBROUTINE STFILT(PROF, NSAMP, ST, PR, C, XIN)
C=============================================================
C  Filter profile using matrices ST, PR, and C.
C
C  <->       PROF  REAL        Input profile. Replaced by the output.
C  -->       NSAMP INTEGER     Number of data values in array PROF.
C  -->       ST    REAL        4x4 state transition matrix.
C  -->       PR    REAL        4x1 partial response vector.
C  -->       C     REAL        4x1 output definition vector.
C  -->       XIN   real        Initial values of filter variables.

C--------------------------------------------
      INTEGER      I, J, K, NSAMP
      REAL       C, PR, PROF, ST, X, XIN, XN
      DIMENSION  C(4), PR(4), PROF(NSAMP+2), ST(4, 4)
      DIMENSION  X(4), XIN(4), XN(4)
c
C  Initialize simulation variables.
      DO 10 I = 1,4
   10    X(I) = XIN(I)

C  Filter profile using the state transition algorithm.
      DO 40 I = 1, NSAMP
      DO 20 J = 1, 4
        XN(J) = PR(J)*PROF(I)
        DO 20 K = 1, 4
   20     XN(J) = XN(J) + X(K)*ST(J, K)

      X(1) = XN(1)
      X(2) = XN(2)
      X(3) = XN(3)
      X(4) = XN(4)

c
      PROF(I) = X(1)*C(1) + X(2)*C(2) + X(3)*C(3) + X(4)*C(4)

   40 CONTINUE
c
      RETURN
      END       ! of SUBR STFILT
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
c           Listing 5. Code to invert matrix.
C===============================================================
      SUBROUTINE INVERT(Y1, N)
C===============================================================
C  This routine will store the inverse of NxN matrix Y1 in matrix YINV.
C  It was copied from "Numerical Recipes."

C  Y1   --> Real     The matrix to be inverted.
C  YINV --> Real     The inverse of matrix Y1.
      INTEGER       N, INDX, I, J
      REAL*4        Y1, YINV, D, A
      DIMENSION     Y1(N, N), YINV(4, 4), INDX(4), A(4, 4)
      DO 8 I = 1, N
        DO 9 J = 1, N
    9     A(I, J) = Y1(I, J)
    8   CONTINUE
      DO 10 I = 1, N
         DO 20 J = 1, N
   20     YINV(I, J) = 0.0
        YINV(I, I) = 1.0
   10   CONTINUE

      CALL LUDCMP(A, INDX, D)
      DO 30 J = 1, N
   30   CALL LUBKSB(A, INDX, YINV(1, J))
      DO 40 I = 1, N
        DO 50 J = 1, N
   50     Y1(I ,J) = YINV(I, J)
   40   CONTINUE
      RETURN
      END

C===============================================================
      SUBROUTINE LUDCMP(A, INDX, D)
C===============================================================
C  This routine was copied from "Numerical Recipes" for matrix
C  inversion.

      INTEGER          N, INDX, NMAX, I, J, IMAX, K
      REAL*4        A, TINY, VV, D, AAMAX, SUM, DUM
      PARAMETER     (NMAX = 100, TINY = 1.0E-20, N = 4)
      DIMENSION     A(N, N), INDX(N), VV(NMAX)
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
      D = 1.0
      DO 10 I = 1, N
        AAMAX = 0.0
        DO 20 J = 1, N
20        IF(ABS(A(I,J)).GT.AAMAX) AAMAX=ABS(A(I,J))
        IF(AAMAX.EQ.0.0) PAUSE 'Singular matrix'
        VV(I) = 1.0/AAMAX
10    CONTINUE
      DO 30 J = 1, N
        DO 40 I = 1, J-1
        SUM = A(I, J)
          DO 50 K = 1, I-1
50          SUM = SUM - A(I, K)*A(K, J)
          A(I, J) = SUM
40        CONTINUE
        AAMAX = 0.0
        DO 60 I = J, N
          SUM = A(I, J)
          DO 70 K = 1, J-1
70          SUM = SUM - A(I, K)*A(K, J)
          A(I, J) = SUM
          DUM = VV(I)*ABS(SUM)
          IF(DUM.GE.AAMAX)THEN
            IMAX = I
            AAMAX = DUM
        ENDIF
60      CONTINUE
        IF(J.NE.IMAX)THEN
          DO 80 K = 1, N
            DUM = A(IMAX, K)
            A(IMAX, K) = A(J, K)
            A(J, K) = DUM
80          CONTINUE.
        D = -D
          VV(IMAX) = VV(J)
        ENDIF
        INDX(J) = IMAX
        IF(A(J, J).EQ.0.0) A(J, J) = TINY
        IF(J.NE.N)THEN
          DUM = 1.0/A(J, J)
          DO 90 I = J+1, N
90          A(I, J) = A(I, J)*DUM
        ENDIF
30      CONTINUE
      RETURN
      END
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

```
C=================================================================
      SUBROUTINE LUBKSB(A, INDX, B)
C=================================================================
C  This routine was copied from "Numerical Recipes" for matrix
C  inversion.

Fig. X1.1. Sample Fortran Program Using Subroutine GETPI
           to Compute Ride Number Index (con.).
```

```
      INTEGER        N, INDX, I, II, LL, J
      REAL*4      A, B, SUM
      PARAMETER   (N = 4)
      DIMENSION   A(N, N), INDX(N), B(N)
      II = 0
      DO 10 I = 1, N
        LL = INDX(I)
        SUM = B(LL)
        B(LL) = B(I)
        IF(II.NE.0)THEN
          DO 20 J = II, I-1
20          SUM = SUM - A(I, J)*B(J)
        ELSEIF(SUM.NE.0)THEN
          II = I
        ENDIF
        B(I) = SUM
10    CONTINUE
      DO 30 I = N, 1, -1
        SUM = B(I)
      IF(I.LT.N)THEN
          DO 40 J = I+1, N
40          SUM = SUM - A(I, J)*B(J)
        ENDIF
        B(I) = SUM/A(I, I)
30    CONTINUE
      RETURN
      END
```

FIG. X1.1 Sample Fortran Program Using Subroutine GETPI to Compute Ride Number Index *(continued)*

X2. SAMPLE ROAD ELEVATION PROFILE DATA SET

X2.1 Included in Appendix X2 is a sample road profile data set in SI units (mm) (Fig. X2.1) that can be used as an input profile data set to the RN computer program shown in Figs. X1.1-X1.10. The sample road profile elevations can be converted to inch-pound units by dividing each value by 25.4.

0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
2.500	2.500
5.000	5.000
7.500	7.500
10.000	10.000
12.500	12.500
15.000	15.000
17.500	17.500
20.000	20.000
17.500	17.500
15.000	15.000
12.500	12.500
10.000	10.000
7.500	7.500
5.000	5.000
2.500	2.500
0.000	0.000
0.000	0.000
0.000	0.000

... (pad with zeros to make a total of 101 numerical data)

0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000

NOTE 1—Elevations are in SI units (mm). The profile consists of identical Fig. tracks, each consisting of zero elevations everywhere except the triangle from 1.0 to 5.0 m peaking at 20.0 mm. The interval between elevation per m. This data set may be used as a test of the user's implementation standard computation.

FIG. X2.1 Sample Road Profile Input Data Set, TRIPULSE.ASC.

X3. OUTPUT FOR THE RIDE NUMBER COMPUTER PROGRAM

X3.1 Included in this appendix is the output of the RN computer program shown in Figs. X1.1-X1.10 for the sample profile data set shown in Fig. X2.1.

X3.1.1 The output of the RN shown here is for a profile data point interval of 150 mm for metric data or 6 in. for inch-pound data.

X3.1.2 The outputs of the RN program for the sample profile data set given in Fig. X2.1 are shown in this section.

X3.1.2.1 The inputs and outputs of the RN test program using Fig. X2.1 sample profile data set (SI units) are shown in Fig. X3.1.

```
Enter Data File Name (in single quotes)
('TRIPULSE.ASC' in example)       :   'TRIPULSE.ASC'

Is road profile data in SI or inch-pound units?

Type 'S' or 'I'                   :    S

Enter length of section, m (ft)
(15 meters in example)            :    15

Enter sample interval, meters     :   .15

Is Input Profile presmoothed?     :    Y



RN, left wheel path          =    1.54

RN, right wheel path         =    1.54

Ride Number                  =    1.54


Distance                     =    14.7 meters
```

FIG. X3.1 Input/output for RNSMP sample program using data input file 'TRIPULSE.ASC'.

REFERENCES

(**1**) Janoff, M. S., Nick, J. B., Davit, P. S., and Hayhoe, G. F., "Pavement Roughness and Rideability," *NCHRP Report 275*, Transportation Research Board, Washington, DC, 1985.

(**2**) Janoff, M. S., "Pavement Roughness and Rideability Field Evaluation," *NCHRP Report 308*, Transportation Research Board, Washington, DC, 1988.

(**3**) Spangler, E. B., and Kelly, W. J., "Use of the Inertial Profilometer in the Ohio DOT Pavement Management System," *FHWA Report FHWA/OH-87/005*, 1987.

(**4**) Spangler, E. B., Kelly, W. J., Janoff, M. S., and Hayhoe, G. F., "Long-Term Time Stability of Pavement Ride Quality Data," *FHWA Report FHWA/OH-91/001*, 1991.

(**5**) Spangler, E.B., Kelly, W.J., "Development and Evaluation of the Ride Number Concept", *ASTM STP 1225*, Vehicle-Road Interaction II, Engineering Foundation Conference, Santa Barbara, California, May 1992.

(**6**) Sayers, M.W. and Karamihas, S. M., "Interpretation of Road Roughness Profile Data", *University of Michigan Transportation Research Institute Report UMTRI 96-19*, June 1996.