SMOOTHING AND DIFFERENTIATION OF DISPLACEMENT-TIME DATA: AN APPLICATION OF SPLINES AND DIGITAL FILTERING

C.L. VAUGHAN

Department of Biomedical Engineering, University of Cape Town and Groote Schuur Hospital, Observatory 7925, Cape (South Africa)

(Received 31 January, 1982)

The advent of the digital computer has allowed workers in the field of biomechanics to perform data smoothing and time differentiation numerically. The methods which are most commonly used are spline functions and digital filtering. This paper provides raw displacement-time data for an object falling in the earth's gravitational field and demonstrates the results obtained using splines and digital filtering to calculate acceleration. It is recommended that the quintic spline is a good method to use provided one has sufficient core available, and the digital filter, with some form of augmentation and/or velocity smoothing, can yield satisfactory results for the user who has a mini- or micro-computer. Listings of the digital filter and finite difference subroutines are provided.

Introduction

Before the advent of the digital computer, workers in the field of biomechanics were obliged to perform their data smoothing and time differentiation using graphical techniques. Despite the obvious tedium involved, many of these early workers were quite prolific in their output and their findings agree quite well with the results of today. For many years applied mathematicians have been developing algorithms for the numerical smoothing and differentiation of displacement data, but it is only recently that these algorithms have become widely used.

The methods which appear to be most commonly used in biomechanics research at present involve use of spline functions and digital filtering (Miller, 1979). The spline is a piecewise differentiable polynominal function satisfying certain continuity conditions on the derivatives at the data points. It was first developed and then expanded by a German mathematician, Reinsch (1967, 1971) (cf. also Herriot and Reinsch, 1976). The International Mathematical and Statistical Libraries (IMSL) (1975) made available *FORTRAN* subroutines which incorporated the *cubic* spline algorithm, and these subroutines have been used extensively by biomechanics researchers. Zernicke *et al.* (1976) compared various smoothing and differentiating techniques and concluded that the cubic spline of the IMSL subroutine gave very acceptable results except at the endpoints where acceleration was known to be non-

Int. J. Bio-Medical Computing (**13**) (1982) 375–386
Elsevier Scientific Publishers Ireland Ltd.

zero. More recently, two workers from the University of Western Australia, Wood and Jennings (1978, 1979), have shown that the *quintic* spline does not suffer from endpoint problems of the cubic spline. Furthermore, since the acceleration derivative ('jerk') was continuous, the authors concluded that the kinetic data generated were more likely to approximate true human movement.

Workers at the University of Waterloo in Canada, led by Winter (1974, 1979), have used a second order Butterworth digital filter to smooth the raw data, and first order finite differences for differentiation. Pezzack *et al.* (1977) compared the acceleration curves generated from film data with analogue curves from an accelerometer, for the planar movement of a single segment. They actually provided the raw angular displacement data so that anyone wishing to replicate their study, or to attempt an alternative smoothing and differentiation procedure, could do so. This was done by Wood and Jennings (1978, 1979) who showed that use of the cubic and quintic spline functions gave excellent acceleration data. Lesh *et al.* (1979), using a non-recursive digital filter described by Rabiner *et al.* (1975), also obtained satisfactory acceleration values for the displacement data of Pezzack *et al.* (1977). Others to have used these data were Phillips *et al.* (1980), who showed that use of an augmented cubic spline (i.e. data points were added at the beginning and end) gave superior results to those obtained when Winter's digital filter was used. This occurred for the special case where the displacement data were terminated at the point where the acceleration was known to be of greatest magnitude.

Various authors have presented different methods for smoothing and differentiating raw displacement data which appear to give satisfactory results for their particular application. However, it is not always possible to check the validity of a particular method nor is it easy to obtain a program listing which can be readily adapted to any machine. The purposes of the present paper are therefore:

(1) to provide raw displacement data for which the associated acceleration is known exactly;
(2) to demonstrate the differentiation results obtained using splines and digital filtering;
(3) to suggest an appropriate method to be used by biomechanics researchers; and
(4) to provide a listing of the short digital filter program for smoothing (Radar *et al.*, 1967; Winter *et al.*, 1974) and a first order finite difference program for differentiating (Miller *et al.*, 1973).

Spline theory

The spline is a piecewise differentiable polynomial function $s(t)$ satisfying certain continuity conditions on the derivatives at the data points. The problem is to find a spline function $s(t)$ of degree $2m - 1$ (where m refers to the mth derivative) which minimises the integral

$$\int_{t_1}^{t_n} [s^m(t)]^2 dt \tag{1}$$

The smoothing is controlled by the boundary condition

$$\sum_{i=1}^{n} \left[\frac{s(t_i) - x_i}{\sigma_i} \right]^2 \leq S \tag{2}$$

where σ_i are the standard errors in the displacement data, and where S is a parameter that specifies the extent of smoothing. If $m = 2$, the norm of the second derivative is minimised, and $s(t)$ is a cubic spline; if $m = 3$, $s(t)$ is a quintic spline. Further details may be found in Reinsch (1967, 1971) or Wood and Jennings (1978, 1979). Suppose that the cubic spline $s(t)$ has been determined for a particular time interval. The first and second derivatives then follow quite naturally:

$$s(t) = C_3t^3 + C_2t^2 + C_1t + C_0 \tag{3}$$

$$\dot{s}(t) = 3C_3t^2 + 2C_2t + C_1 \tag{4}$$

$$\ddot{s}(t) = 6C_3t + 2C_2 \tag{5}$$

The IMSL cubic spline subroutine ICSSCU was used in the present study, as was an adaptation of the quintic spline subroutines of Jennings (1979). Both sets of programs were used on an IBM 370 machine.

Digital filter theory

In human movement activities the frequency of the displacement signal is almost always less than the frequency of the noise. The purpose of a digital filter, therefore, is to filter out the high-frequency noise while allowing the low-frequency displacement signal to pass through untouched. The format of a low-pass digital filter is as follows:

$$x_i' = a_0x_i + a_1x_{i-1} + a_2x_{i-2} + b_1x_{i-1}' + b_2x_{i-2}' \tag{6}$$

where x' refers to filtered output coordinates, x refers to unfiltered coordinate data, i refers to the ith sample frame, and $a_0 \cdots b_2$ are the filter coefficients. These filter coefficients are constants that depend on the type and order of the filter, the sampling frequency (i.e. frame rate), and the cutoff frequency (i.e. how much noise should be attenuated). As can be seen from Eqn. (6), the filtered output x'_i is a weighted version of the immediate and past raw data, plus a weighted contribution of past filtered output.

For the present study, the second-order low-pass Butterworth filter was used. Further details may be obtained in Rabiner *et al.* (1975) or Winter (1979). A listing of the subroutine DIGFIL is included at the end of this paper as an appendix.

Finite difference theory

Finite difference methods have been based on Taylor series expansions (Miller *et al.*, 1973) and have provided formulae for calculating first and second derivatives of displacement-time data. The first and second derivatives (i.e. velocity and acceleration) are expressible as

$$\dot{x}_i = \frac{x_{i+1} - x_{i-1}}{2\Delta t} \tag{7}$$

$$\ddot{x}_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2} \tag{8}$$

where x is an imput data point, i refers to the ith point, and Δt is the time between adjacent points. Equations (7) and (8) are known as central difference formulae. Forward and backward difference formulae were used for derivatives of displacement data at the beginning and end of the data set (Miller *et al.*, 1973). All these formulae are approximations since the time interval Δt is not infinitely small. Therefore, any noise in the input signal has a large influence on the accuracy of the derivative values (Lesh *et al.*, 1979). A listing of the subroutine FIDIFF is included at the end of this paper as an appendix.

Collection of data

A golf ball, falling in the earth's gravitational field, was filmed with the aid of a Locam high-speed motion camera. The frame rate was set at approximately 100 frames/s. Accurate temporal calibration was done by means of a large clock with a 1-s sweep placed in the camera's field of view, while the linear calibration was achieved by placing a scale of known length in the ball's plane of movement. The film was digitised on a Vanguard Motion Analyser which had an image size of 14 cm × 19 cm and a resolution of 0.025 mm. On average, objects viewed on the Vanguard image were 1/25 their real-life size.

If air resistance on the golf ball is ignored (and this is a reasonable assumption when all factors are considered), the upward acceleration has the form:

$$\ddot{y} = -g \tag{9}$$

where the double dots indicate double differentiation with respect to time and g is the acceleration due to gravity (a constant, approximately equal to 9.8 m/s^2). Integration yields the expression for the vertical displacement

$$y = y_0 + \dot{y}_0 t - \tfrac{1}{2}gt^2 \tag{10}$$

where y_0 and $\dot{y}_0$ are the initial displacement and velocity values respectively. It was possible, using a least-squares minimisation technique, to fit a second order polynomial to the displacement-time data, and see whether the value for g could be accurately predicted. This was done and the results are presented in the next section. Each of the smoothing and differentiating methods (splines and digital filtering) was used to obtain acceleration and these results too are presented in the next section.

Results

The raw displacement data for the falling golf ball are presented in Table 1. The purpose of including these data is to provide an opportunity for replicating the study or attempting an alternative smoothing and differentiating procedure. It should be pointed out that the first point digitised did not coincide with the instant of ball release since that point was difficult to observe and the camera had not reached a constant frame rate. The initial velocity of the ball was therefore not equal to zero.

TABLE 1

RAW DISPLACEMENT DATA FOR FALLING GOLF BALL

Measured in m above ground level; time between consecutive data points is 0.00985 s.

1.770	1.757	1.748	1.740	1.726
1.715	1.698	1.683	1.667	1.651
1.632	1.612	1.593	1.572	1.551
1.530	1.507	1.483	1.445	1.428
1.401	1.371	1.343	1.311	1.279
1.245	1.212	1.175	1.143	1.105
1.063	1.029	0.991	0.953	0.910
0.869	0.823	0.779	0.732	0.691
0.644	0.595	0.548	0.501	0.447
0.395	0.350	0.294	0.243	0.185

A parabola was fitted to these data by the least-squares method. A Hewlett Packard HP-9815A desk top calculator, which used a packaged program, was employed for this purpose. The following result was obtained:

$$y = 1.771 - 0.927t - 4.896t^2 \tag{11}$$

The correlation coefficient between measured and predicted displacements was 1.000 and the standard error of the estimate was 0.003 m. Reference to Eqn. (10) shows that the initial displacement was 1.771 m, the initial velocity was −0.927 m/s, and the value for the acceleration (in the absence of velocity dependent retardation forces) due to gravity was ($-4.896 \times 2 =$) -9.792 m/s^2. This value for g compares very favourably with the Iowa Geological Survey's value of −9.802418 m/s^2 for the Iowa City Municipal Airport (Hase *et al.*, 1969). (*Note:* parts of this study were performed while the author was a doctoral student at the University of Iowa.)

Five separate methods were used to smooth and differentiate the raw data of Table 1:

(a) a cubic spline (IMSL, 1975) with $\sigma_i = 1.0$ and $s = 0.0005$ (cf. Eqn. (2));
(b) a quintic spline (Jennings, 1979) with $\sigma_i = 1.0$ and $s = 0.0005$;
(c) a digital filter (Winter *et al.*, 1974) with a cutoff frequency of 6.0 followed by first order finite differences (Miller *et al.*, 1973);
(d) a digital filter with a cutoff frequency of 6.0 and the raw data augmented by 10 points at the beginning and end, followed by first order finite differences; and
(e) a digital filter with cutoff of 6.0 and augmented by 20 points where the filter is applied a second time (cutoff = 4.0) to smooth the velocity-time data.

The results of each of these procedures are presented in Figs. 1–5. Figures 1–3 were drawn to the same scale while Figs. 4 and 5, which were done at a later date, are slightly different (Missing points in Figs. 3 and 4 were out of the −15 to 5 m/s^2 scale.)

The curve generated by the cubic spline in Fig. 1 illustrates the requirement of zero accelerations at the end-points. Figure 1 is in fact very similar to the curve of McLaughlin *et al.* (1977) who also used a cubic spline to generate the acceleration of a falling weight. The acceleration obtained by the quintic spline (Fig. 2) is an extremely good approximation to the gravitational constant of −9.802 m/s^2. The results of the ball drop experiment suggest therefore that the quintic spline is superior to the digital filter and the cubic spline.

The end-point problems of the digital filter (Fig. 3) which are independent of the cutoff frequency used, are in agreement with the findings of Phillips *et al.* (1980).

Despite the obvious success of the quintic spline in predicting acceleration for the raw data of Table 1, a serious drawback is the length of the program especially if the

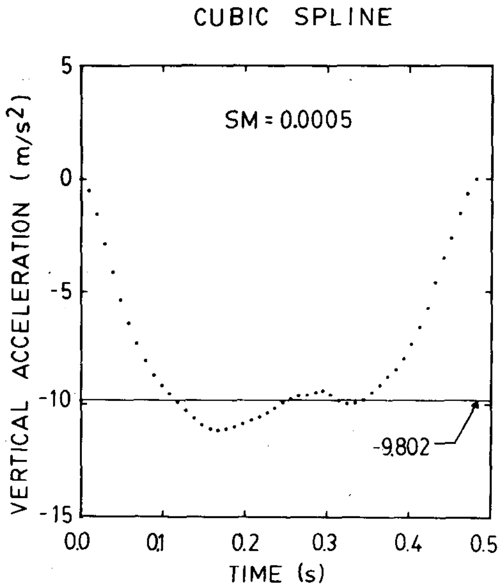

Fig. 1. Vertical acceleration for a ball-drop experiment (cubic spline).

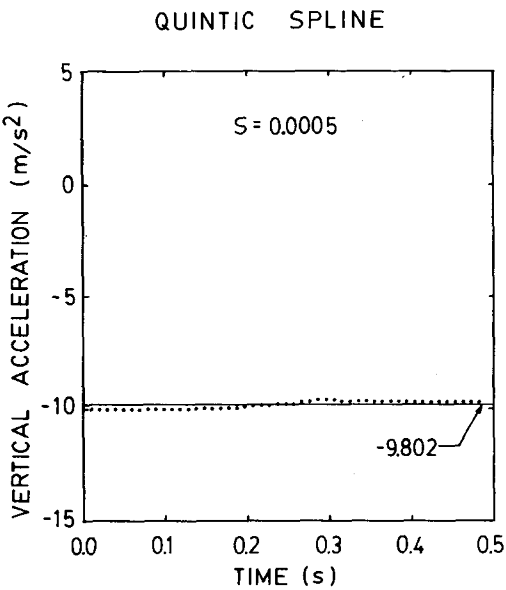

Fig. 2. Vertical acceleration for ball-drop experiment (quintic spline).

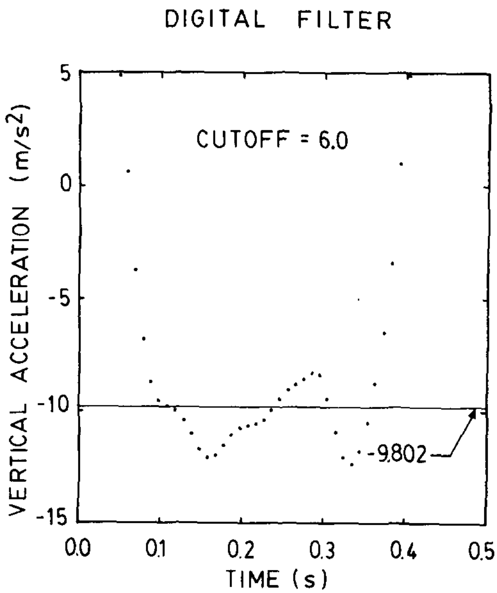

Fig. 3. Vertical acceleration for ball-drop experiment (digital filter and first order finite differences).

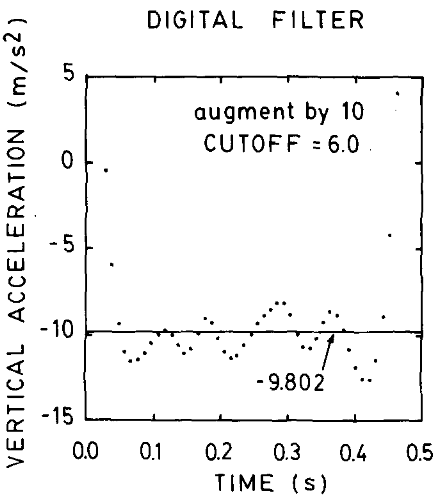

Fig. 4. Vertical acceleration for ball-drop experiment (digital filter and first order finite differences, augmenting displacement data by 10).

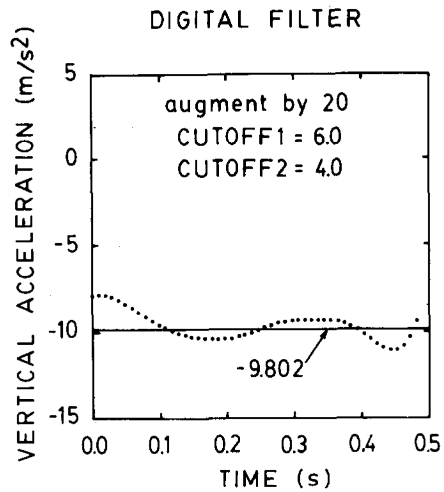

Fig. 5. Vertical acceleration for ball-drop experiment (digital filter and first order finite differences, augmenting displacement data by 20 and filtering velocity data).

user is working in a mini- or micro-computer environment. This problem prompted further work with the digital filter which together with the finite difference program (cf. Appendix) require relatively little memory.

Comparison of Figs. 3 and 4 shows that augmentation by 10 does lead to some improvement, although the acceleration is still quite 'noisy'. Augmentation by 20 plus filtering the velocity data gives a substantial improvement although the result is still not as good as that of the quintic spline (Fig. 5). It may be argued that an alternative means of augmenting is to digitise extra frames before and after the period of interest rather than to augment artifically as was done in this study. However, digitising can be a long and tedious procedure (approximately 1 h/patient for a full gait cycle) so more digitising than that necessary is not recommended.

Conclusions

Based on the limited data provided in this paper – which concern a body falling in the earth's gravitational field, a situation quite similar to some biomechanical activities – the following conclusions may be drawn:

(1) the quintic spline is a good method to use for smoothing and differentiating raw displacement data provided one has sufficient core available; and
(2) the digital filter, with some form of augmentation and/or velocity smoothing, can yield satisfactory results for the user who has a small machine.

References

Hase, D.H., Campbell, R.B. and van Eck, O.J., Iowa gravity base station network, Report of Investigations 8, Iowa Geological Survey, Iowa City, 1969.

Hatze, H., The use of optimally regularized Fourier series for estimating higher-order derivatives of noisy biomechanical data, *J. Biomech.*, **14** (1981) pp. 13–18.

Herriot, J.G. and Reinsch, C.H., Procedures for quintic natural spline interpolation, *ACM Trans. Math. Software,* **2** (1976) pp. 281–289.

International Mathematical and Statistical Libraries, *The IMSL Library Vols. 1 and 2,* (1975).

Jennings, L.S., Cubic and quintic spline fitting least squares and interplation, Department of Mathematics, University of Western Australia, 1979.

Lesh, M.D., Mansour, J.M. and Simon, S.R., A gait analysis subsystem for smoothing and differentiation of human motion data, *J. Biomech. Eng.*, **101** (1979) pp. 205–212.

McLaughlin, T.M., Dillman, C.J. and Lardner, T.J., Biomechanical analysis with cubic spline functions, *Res. Q.*, **48** (1977) pp. 569–582.

Miller, D.I., Body segment contributions to sport skill performance: two contrasting approaches, *Res. Q. Exerc. Sport,* **51** (1980) pp. 219–233.

Miller, D.I. and Nelson, R.C., *Biomechanics of Sport,* Lea and Febiger, Philadelphia, 1973, pp. 245–246.

Pezzack, J.C., Norman, R.W. and Winter, D.A., An assessment of derivative determining techniques used for motion analysis, *J. Biomech.* **10** (1977) pp. 377–382.

Phillips, S.J. and Roberts, E.M., The end-point problem in biomechanical acceleration data, presented at AAHPERD Convention, Detroit, 1980.

Radar, C.M. and Gold, B., Digital filtering design techniques in the frequency domain, *Proc. IEEE,* **55** (1967) pp. 149–171.

Reinsche, C.H., Smoothing by spline functions, *Numer. Math.*, **10** (1967) pp. 177–183.

Reinsche, C.H., Smoothing by spline functions II, *Numer. Math.*, **16** (1971) pp. 451–454.

Winter, D.A., *Biomechanics of Human Movement,* John Wiley and Sons, New York, 1979.

Winter, D.A., Sidwall, H.G. and Hobson, D.A., Measurement and reduction of noise in kinematics of locomotion, *J. Biomech.*, **7** (1974) pp. 157–159.

Wood, G.A. and Jennings, L.S., An automatic spline for cinematographic data smoothing, in *Biomechanische Untersuchungsmethoden im sport,* G. Marhold (Ed.), International Symposium, Karl-Marx Stadt, East Germany, 1978.

Wood, G.A. and Jennings, L.S., On the use of spline functions for data smoothing, *J. Biomech.*, **12** (1979) pp. 477–479.

Zernicke, R.F., Caldwell, G and Roberts, E.M., Fitting biomechanical data with cubic spline functions, *Res. Q.*, **47** (1976) 9–19.

APPENDIX

```
      SUBROUTINE DIGFIL(N,RD,FD,CUTOFF,DT)
C***************************************************************************
C
C   PROGRAMMER : KIT VAUGHAN   FEBRUARY  1980
C
C   THIS SUBPROGRAM SMOOTHS RAW DISPLACEMENT TIME DATA (SAMPLED AT EQUAL
C   TIME INTERVALS) USING A DIGITAL FILTER TECHNIQUE.  IT HAS BEEN ADAPTED
C   FROM THE FOLLOWING TWO REFERENCES
C
C      RADAR,C.M. AND B.GOLD
C      DIGITAL FILTERING DESIGN TECHNIQUES IN THE FREQUENCY DOMAIN
C      PROCEEDINGS IEEE, 55:149-171, 1967
C
C      WINTER,D.A., H.G.SIDWALL AND D.A.HOBSON
C      MEASUREMENT AND REDUCTION OF NOISE IN KINEMATICS OF LOCOMOTION
C      JOURNAL OF BIOMECHANICS, 7:157-159, 1974
C
C   N  IS THE NUMBER OF DATA POINTS,  RD  IS THE RAW DATA,  FD  IS THE FILTERED
C   DATA,  CUTOFF  IS THE CUTOFF FREQUENCY DESIREI FOR THE FILTERING PROCEDURE
C   AND  DT  IS THE INVERSE OF THE SAMPLING FREQUENCY
C
C***************************************************************************
      DIMENSION RD(N),FD(N),RT(500)
      CUTOFF=CUTOFF/0.802
      SV=1./(CUTOFF*DT)
      IF(SV.LT.4.0) GO TO 7
      PI=4.*ATAN(1.)
      Z=PI*CUTOFF*DT
      Z1=SIN(Z)
      Z2=COS(Z)
      WC=Z1/Z2
      A=2.*WC*SQRT(0.5)
      B=0.5*A*A


C***************************************************************************
C
C   DEFINE THE FILTER COEFFICIENTS
C
C***************************************************************************
      C1=B/(1.0+A+B)
      C2=2.*C1
      C3=C1
      C4=2.*(1.0-B)/(1.0+A+B)
      C5=(A-B-1.0)/(1.0+A+B)
      DO 1 I=1,N
1     RT(I)=RD(I)
      II=1
6     CONTINUE
      FD(1)=RT(1)
      FD(2)=RT(2)
C***************************************************************************
C
C   SMOOTH USING THE LOW-PASS DIGITAL FILTER
C
C***************************************************************************
      DO 2 I=3,N
2     FD(I)=C1*RT(I)+C2*RT(I-1)+C3*RT(I-2)+C4*FD(I-1)+C5*FD(I-2)
      J=1
      K=N
      DO 4 I=1,N
      RT(I)=FD(K)
      J=J+1
      K=N-J+1
4     CONTINUE
```

```
      II=II+1
      IF(II.LE.2) GO TO 6
      DO 5 I=1,N
5     FD(I) =RT(I)
      CUTOFF=0.802*CUTOFF
      RETURN
7     WRITE(6,100)
100   FORMAT(1X,' SAMPLING THEORY VIOLATED',/,
     1 ' (SAMPLE RATE/CUTOFF) .LT. 4')
      RETURN
      END


      SUBROUTINE FIDIFF(N,DT,X,XD,XDD)
C*****************************************************************************
C
C   PROGRAMMER : KIT VAUGHAN    FEBRUARY  1980
C
C   THIS SUBPROGRAM USES FORMULAE BASED ON FIRST ORDER FINITE
C   DIFFERENCES ( MILLER D.I. AND R.C. NELSON , BIOMECHANICS OF SPORT,
C   LEA AND FEBIGER, 1973) FOR OBTAINING THE FIRST AND SECOND DERIVATIVES
C   OF DISPLACEMENT TIME DATA
C
C   N  IS THE NUMBER OF DATA POINTS,  DT  IS THE INVERSE OF THE SAMPLING
C   FREQUENCY,X  IS DISPLACEMENT, XD  IS VELOCITY AND  XDD  IS ACCELERATION
C
C*****************************************************************************
      DIMENSION X(N),XD(N),XDD(N)
C*****************************************************************************
C
C   FORWARD DIFFERENCES
C
C*****************************************************************************
      XD(1)=(X(2)-X(1))/DT
      XDD(1)=(X(3)-2.*X(2)+X(1))/(DT*DT)


C*****************************************************************************
C
C   CENTRAL DIFFERENCES
C
C*****************************************************************************
      NN=N-1
      DO 1 I=2,NN
      XD(I)=(X(I+1)-X(I-1))/(2.*DT)
      XDD(I)=(X(I+1)-2.*X(I)+X(I-1))/(DT*DT)
1     CONTINUE
C*****************************************************************************
C
C   BACKWARD DIFFERENCES
C
C*****************************************************************************
      XD(N)=(X(N)-X(N-1))/DT
      XDD(N)=(X(N)-2.*X(N-1)+X(N-2))/(DT*DT)
      RETURN
      END
```