# Smoothing and Interpolation by Moving-Window Least Squares Polynomial Fits: Application to Energy- and Charge-Deposition Distributions by Electrons

Tatsuo TABATA

Reprinted from Bulletin of Osaka Prefecture University,
Series A, Vol. 46, No. 2, 1997

Bulletin of Osaka Prefecture University
Series A, Vol. 46, No. 2, 1997, pp. 71−78

# Smoothing and Interpolation by Moving-Window Least Squares Polynomial Fits: Application to Energy- and Charge-Deposition Distributions by Electrons

Tatsuo TABATA*

(Received October 24, 1997)

A FORTRAN code for moving-window least squares polynomial fits has been developed, and used to smooth and interpolate energy- and charge-deposition distributions of electrons in materials. It has worked well not only for main distributions with different peak widths but also for regions going into the bremsstrahlung tail in the energy-deposition distributions. Preliminary processing, by which a few points have been inserted by quadratic interpolation, has sometimes proved useful in the positive-charge build-up region of the charge-deposition distribution.

## 1. Introduction

In processing experimental or computer-simulation data, it is frequently necessary to apply smoothing for reducing the effect of statistical fluctuations. The method of moving-window averaging and spline-fit smoothing are available for this purpose. However, the former method cannot fully remove fluctuations, and causes the deterioration of resolution; the latter is not applicable when a cubic polynomial is a poor approximation to a segment of the curve. Examples of inapplicability of a spline-fit algorithm[1] can be seen in the paper of Frederickson and Woolf[2] on the charge-deposition distribution by electrons. Their results of smoothing were unable to express the rapid change in slope due to positive charge build-up at shallow depths.

Interpolation or extrapolation of data is also important for the effective use of available data, and smoothing and interpolation are sometimes required at the same time. A combination of smoothing and interpolation can be made by moving-window least squares polynomial fits. This algorithm is also called local polynomial regression, and discussed in detail in the recent textbook by Simonoff[3]**. For the purpose of smoothing the data points that are equally spaced, a method equivalent to the above algorithm is available, and is called Savitzky Golay smoothing filters[4, 5]. Savitzky-Golay algorithm performs smoothing much faster than the direct use of least-squares fit, but it cannot be used for interpolation.

In the present paper a simple and versatile FORTRAN code for smoothing and interpolation is given. The algorithm used is the method of moving-window least squares polynomial fits. Some examples are presented of applying this code to the energy- and charge-deposition distributions of electrons in materials.

## 2. FORTRAN Programs

We consider an algorithm to carry out smoothing and interpolation at the same time. The central part of the algorithm finds a smoothed and interpolated (or extrapolated) value $y_{\mathrm{out}}$ of the dependent variable for an input value $x_{\mathrm{in}}$ of the independent variable. A polynomial of an arbitrary degree $m$ is fitted by the method of least squares to $n$ data points $[(X_i, Y_i),$

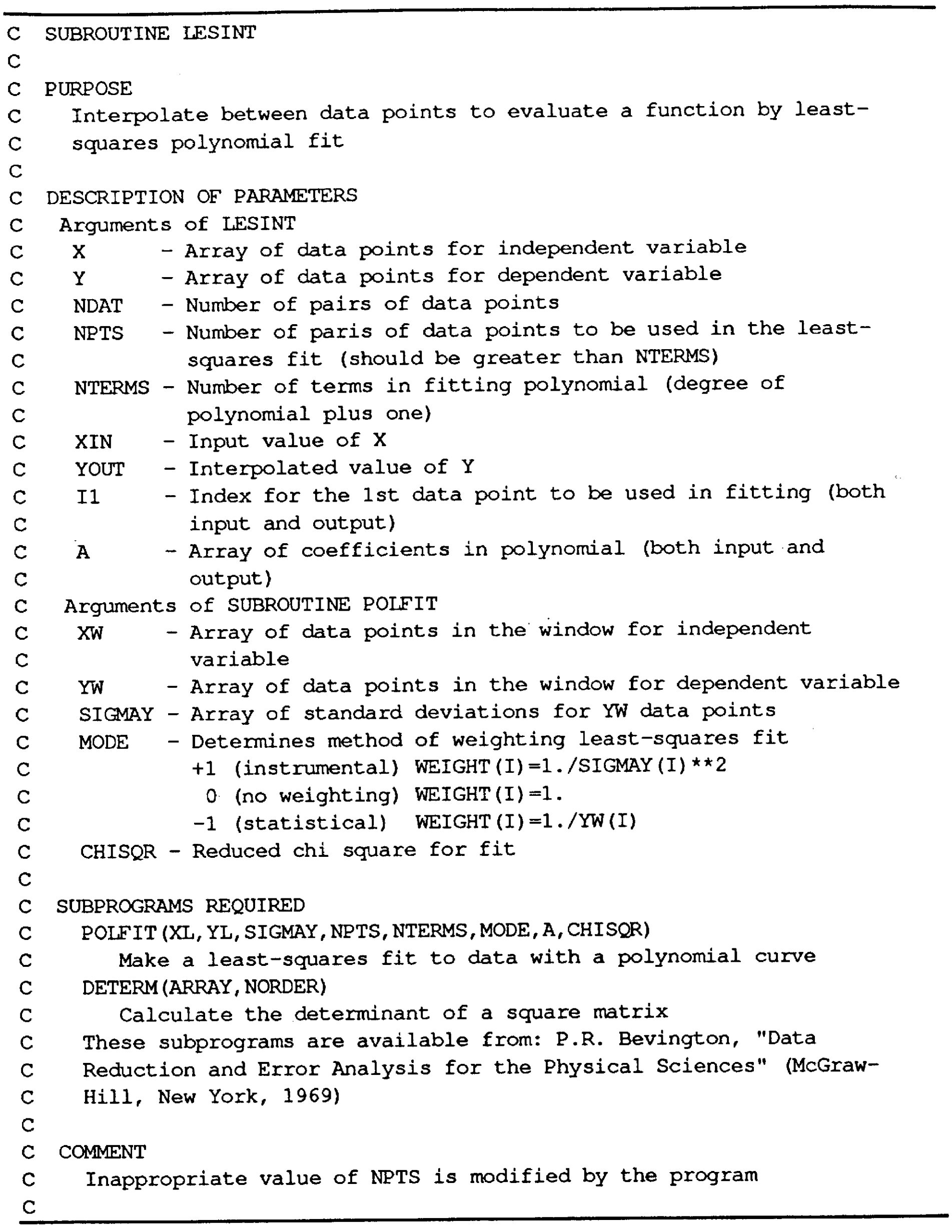
* Department of Fundamental Science, Research Institute for Advanced Science and Technology

** Simonoff's book includes more than 750 references. Learning the algorithm of local polynomial regression by this book is however rather inconvenient for the urgent need.

$(i=i_1,\ i_1+1,\ \ldots,\ i_1+n-1)$], which is present in a window taken from a total set of $N$ data points [$(X_i, Y_i)$, $(i=1,\ 2,\ \ldots,\ N)$]. The number $n$ should be greater than $(m+1)$, and the position of the window is chosen so that $x_{in}$ is located as nearly as possible at the center of the window.

A FORTRAN subprogram for such an algorithm can simply be constructed by modifying the program INTERP for polynomial interpolation given by Bevington[6]. It is a subroutine subprogram to calculate $y_{out}$ for $x_{in}$ by $m$th-degree polynomial fit to $(m+1)$ data points. In this program, we replace the fitting procedure by least-squares $m$th-degree polynomial fit to $n$ data points. A subroutine subprogram for the least-squares polynomial fit is also available from Bevington's textbook; it is named POLFIT. Modifying INTERP and using POLFIT, therefore, we obtain a subprogram named LESINT as given in Fig. 1. The POLFIT program needs a function subprogram called DETERM to calculate the determinant of a square matrix; it is also available from Bevington's book.

The description of parameters in LESINT is given by comment statements in Fig. 1. Note that the parameter NPTS represents $n$, and NTERMS, $(m+1)$. Statements 11−27 are the same as those of INTERP, and find the appropriate window of the independent variable to be used for fitting. Least-squares fitting



```
C  SUBROUTINE LESINT
C
C  PURPOSE
C    Interpolate between data points to evaluate a function by least-
C    squares polynomial fit
C
C  DESCRIPTION OF PARAMETERS
C   Arguments of LESINT
C    X      - Array of data points for independent variable
C    Y      - Array of data points for dependent variable
C    NDAT   - Number of pairs of data points
C    NPTS   - Number of paris of data points to be used in the least-
C             squares fit (should be greater than NTERMS)
C    NTERMS - Number of terms in fitting polynomial (degree of
C             polynomial plus one)
C    XIN    - Input value of X
C    YOUT   - Interpolated value of Y
C    I1     - Index for the 1st data point to be used in fitting (both
C             input and output)
C    A      - Array of coefficients in polynomial (both input and
C             output)
C   Arguments of SUBROUTINE POLFIT
C    XW     - Array of data points in the window for independent
C             variable
C    YW     - Array of data points in the window for dependent variable
C    SIGMAY - Array of standard deviations for YW data points
C    MODE   - Determines method of weighting least-squares fit
C             +1 (instrumental) WEIGHT(I)=1./SIGMAY(I)**2
C              0 (no weighting) WEIGHT(I)=1.
C             -1 (statistical)  WEIGHT(I)=1./YW(I)
C    CHISQR - Reduced chi square for fit
C
C  SUBPROGRAMS REQUIRED
C    POLFIT(XL,YL,SIGMAY,NPTS,NTERMS,MODE,A,CHISQR)
C       Make a least-squares fit to data with a polynomial curve
C    DETERM(ARRAY,NORDER)
C       Calculate the determinant of a square matrix
C    These subprograms are available from: P.R. Bevington, "Data
C    Reduction and Error Analysis for the Physical Sciences" (McGraw-
C    Hill, New York, 1969)
C
C  COMMENT
C    Inappropriate value of NPTS is modified by the program
C
```

Fig. 1 FORTRAN program LESINT for smoothing and interpolation

```
      SUBROUTINE LESINT(X,Y,NDAT,NPTS,NTERMS,XIN,YOUT,I1,A)
      DIMENSION A(NTERMS),X(NDAT),Y(NDAT),XW(50),YW(50)
      DATA MODE/0/
C  Search for appropriate value of X(I1)
      I1=I1M
   11 DO 19 I=1,NDAT
      IF(XIN-X(I)) 13,13,19
   13 I1=I-NPTS/2
      IF(I1) 15,15,21
   15 I1=1
      GO TO 21
   19 CONTINUE
      I1=NDAT-NPTS+1
   21 I2=I1+NPTS-1
      IF(NDAT-I2) 23,31,31
   23 I2=NDAT
      I1=I2-NPTS+1
   25 IF(I1) 26,26,31
   26 I1=1
   27 NPTS=I2-I1+1
C  Select data for fitting
   31 IF(I1.EQ.I1M) GOTO 50
      DO 41 I=1,NPTS
      J=I1+I-1
      XW(I)=X(J)
   41 YW(I)=Y(J)
C  Determine coefficients A
   45 CALL POLFIT(XW,YW,SIGMAY,NPTS,NTERMS,MODE,A,CHISQR)
C  Evaluate polynomial
   50 YOUT=A(NTERMS)
      DO 51 I=2,NTERMS
   51 YOUT=YOUT*XIN+A(NTERMS-I+1)
   61 RETURN
      END
```

Fig. 1 (continued)

is made by calling the subroutine POLFIT by statement 45.

The partial algorithm described above has to be repeated for a set of input values [$x_i$,($i$= 1, 2, ..., $n_{in}$)] of the independent variable to obtain output values [$y_i$, ($i$= 1, 2, ..., $n_{in}$)] of the dependent variable. This repetition can be made by a calling program for LESINT. A sample calling program, named LESINC, is given in Fig. 2. For each of $x_i$'s an appropriate window is sought by LESINT. When the result of the search is the same as the previous result, the previous coefficients of the polynomial are used to determine the smoothed and interpolated value. Otherwise, coefficients are newly determined. This is controlled by statement 31 of LESINT.

In the examples given later the spacing of the input values [$x_i$, ($i$=1, 2, ..., $n_{in}$)] is not larger than that of the original data. In such cases we can repeat the smoothing for the total curve, using the smoothed and interpolated points, [($x_i$, $y_i$),($i$=1, 2, ..., $n_{in}$)], as new input data instead of the original ones. This reduces any residual fluctuations or small jumps of values; the latter can be caused by the movement of the window in the first run. For further smoothing, another repetition can be made. Procedures for such repetition are included in the sample calling program of Fig. 2. The description of all the parameters whose values have to be defined or read by the calling program is given by the comment statements of the program.

The whole code consists of LESINC, LESINT, POLFIT and DETERM. The first two programs have recently been modified a little. Therefore, the code has been checked by compiling it by Pro Fortran system for Power Macintosh (manufactured by Absoft Corporation) and by running it for the example problems given in the next section.

## 3. Examples of Application and Discussion

### 3. 1 Data used

The present code was applied to two sets of data.

```
C  SAMPLE CALLING PROGRAM LESINC FOR LESINT
C
C  SUBPROGRAMS REQUIRED
C    LESINT(X,Y,NDAT,NPTS,NTERMS,XIN,YOUT,I1,A)
C       Interpolate between data points to evaluate a function by least-
C       squares polynomial fit
C
C  DESCRIPTION OF PARAMETERS WHOSE VALUES ARE TO BE DEFINED OR READ
C    NPTS   - Nnmber of paris of data points used for fitting, to
C             determine a single interpolated point (should be greater
C             than NTERMS)
C    NTERMS - Number of terms in fitting polynomial (degree of
C             polynomial plus one)
C    NINT   - Number of points at which interpolation is to be made
C    XINT   - Array of the values of independent variable at which
C             interpolation is to be made
C    DX     - Interval between XINT(I) and XINT(I+1)
C    COM    - Character data for comment
C    X      - Array of data points for independent variable
C    Y      - Array of data points for dependent variable
C
      PROGRAM LESINC
      CHARACTER*80 COM
      DIMENSION A(10),X(500),Y(500),XINT(500),YINT1(500),YINT2(500),
     + YINT3(500)
      DATA I1,A,DX,NINT/0,10*0.,0.01,141/
      OPEN(5,FILE="",STATUS="OLD")
      OPEN(6,FILE="")
C   Read NPTS and NTERMS
      WRITE(*,100)
      READ(*,*) NPTS,NTERMS
C   Prepare XINT
      DO 1 I=1,NINT
    1 XINT(I)=FLOAT(I-1)*DX
C   Read COM, X, Y
      READ(5,500) COM
      NDAT=0
    5 NDAT=NDAT+1
      READ(5,*,END=10) X(NDAT),Y(NDAT)
      GO TO 5
   10 NDAT=NDAT-1
C   Smooth and interpolate
      DO 15 I=1,NINT
   15 CALL LESINT(X,Y,NDAT,NPTS,NTERMS,XINT(I),YINT1(I),I1,A)
C   Repeat smoothing for the smoothed and interpolated data
      DO 20 I=1,NINT
   20 CALL LESINT(XINT,YINT1,NINT,NPTS,NTERMS,XINT(I),YINT2(I),I1,A)
      DO 25 I=1,NINT
   25 CALL LESINT(XINT,YINT2,NINT,NPTS,NTERMS,XINT(I),YINT3(I),I1,A)
C   Write results
      WRITE(6,601)
      DO 30 I=1,NINT
   30 WRITE(6,602) XINT(I),YINT3(I)
      CLOSE(5)
      CLOSE(6)
      STOP
  100 FORMAT(1H ,'NPTS,NTERMS?')
  500 FORMAT(A80)
  601 FORMAT(1H ,4X,'XINT',10X,'YINT')
  602 FORMAT(1H ,1P6E14.4)
      END
```

Fig. 2 FORTRAN sample calling program for LESINT

One was the energy-deposition distributions of electrons calculated[7] by the TIGER code in the ITS-3.0 Monte Carlo system[8], and the other was the charge-deposition distributions of electrons experimentally obtained[9]. The results of smoothing and interpolation for the former distributions were published[10]. The results for the latter distributions were used for further interpolation of distributions as a function of incident-electron energy to yield a set of benchmarks[11]. In the following subsections some examples are taken from these results to demonstrate how the code has been used and how it worked.

### 3. 2 Energy-deposition distributions

In Fig. 3 examples of smoothed and interpolated results are shown together with original data for the energy-deposition distribution of electrons. The electrons of 0.5-MeV energy were assumed to be incident on effectively semi-infinite absorbers, which consisted of various elemental materials. The abscissa denoted as the scaled depth $X$ is defined as the depth divided by the continuous slowing-down approximation range $r_0$ of electrons in the absorber material. The ordinate denoted as the scaled energy deposition $Y$ is defined as the energy deposition per incident electron multiplied by $r_0/T_0$, where $T_0$ is the initial electron energy. Thus $X$ and $Y$ are dimensionless quantities. Points represent the original data, which are spaced by the scaled-depth interval of 0.0125. The smoothed and interpolated results with 0.01 spacing are shown by connecting them by straight lines. The statistical fluctuations of the data were not so large that plots with an expanded scale, as given later, are required fully to appreciate the effect of smoothing.

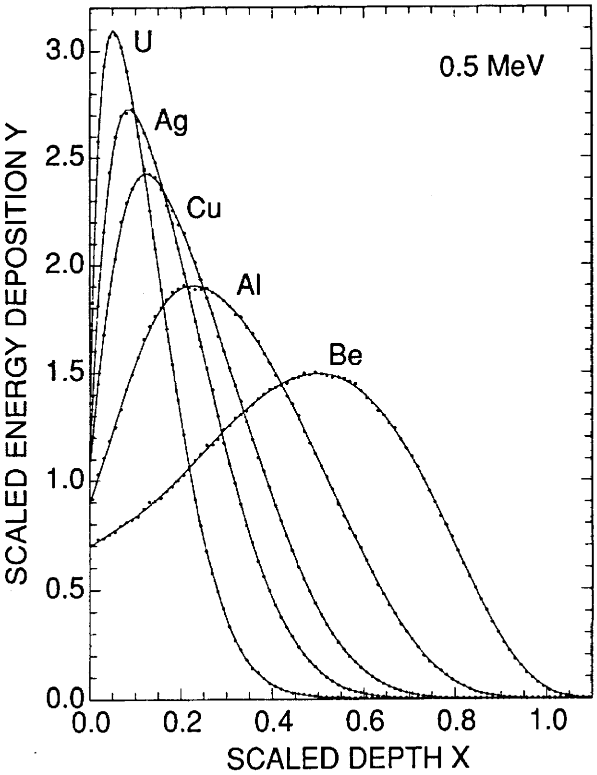


Fig. 3 Energy-deposition distributions of 0.5-MeV electrons incident on various absorbers of effectively semi-infinite thickness. Points, Monte Carlo data; lines, results of smoothing and interpolation.

Table 1 The degree $m$ of the polynomial, the number $n$ of the data points in the window and the width $w$ of the window used for smoothing and interpolation of data in Fig. 1. The values of $w$ are given as the scaled depth interval, which is defined as the ratio of the depth interval to the continuous slowing-down approximation range $r_0$ of incident electrons in the material.

| Material | Main part | | | Transient and tail regions | | |
|---|---|---|---|---|---|---|
| | $m$ | $n$ | $w$ | $m$ | $n$ | $w$ |
| $_4$Be | 3 | 18 | 0.225 | 1 | 3 | 0.038 |
| $_{13}$Al | 3 | 18 | 0.225 | 2 | 10 | 0.125 |
| $_{29}$Cu | 3 | 12 | 0.150 | 3 | 18 | 0.225 |
| $_{47}$Ag | 3 | 8 | 0.100 | 3 | 14 | 0.175 |
| $_{92}$U | 3 | 6 | 0.075 | 2, 1 (see note[a]) | 10 | 0.125 |

[a] For the transient and tail regions of the distribution in U, two values of $m$ were used, the first for the transient region and the second for the tail region.

Values of $m$, $n$ and the width $w$ of the window, used in applying the present code to the data in Fig. 1, are given in Table 1. For the main part of the distribution, $m=3$ was always good. However, a different choice was necessary at large depths, where the main distribution due to collision energy loss is merged into a low tail caused by radiative energy loss (bremsstrahlung process). Smaller values of $n$ (accordingly, smaller $w$) were used for narrower main distributions. The values of $m$ and $n$ for the transient and tail regions were determined to compromise the opposing requirements by the rapid and slow changes of curvature in these regions.

The curve for the main distribution was connected to the curve for the transient and tail regions at the point where the two curves show the best agreement. An example of such connection is given in Fig. 4. For the distribution in this figure, the results given by the dotted line were used on the left-hand side of the arrow, and the results shown by the solid line, on the right-hand side.

The effect of repeated smoothing can be seen from Fig. 5. In this figure the ratio $Y/Y_{\rm final}$ is plotted as a function of scaled depth $X$. The symbol $Y$ represents here either the original data or the intermediate curves, and $Y_{\rm final}$, the final curve. The root-mean-square (rms) of $(Y/Y_{\rm final} - 1)$ between the scaled thicknesses 0 and 0.8 is $6.7\times10^{-3}$ for the original data, $8.1\times10^{-4}$ for the first smoothed curve and $2.4\times10^{-4}$ for the second smoothed curve. These values, as well as Fig. 5, indicate the effectiveness of

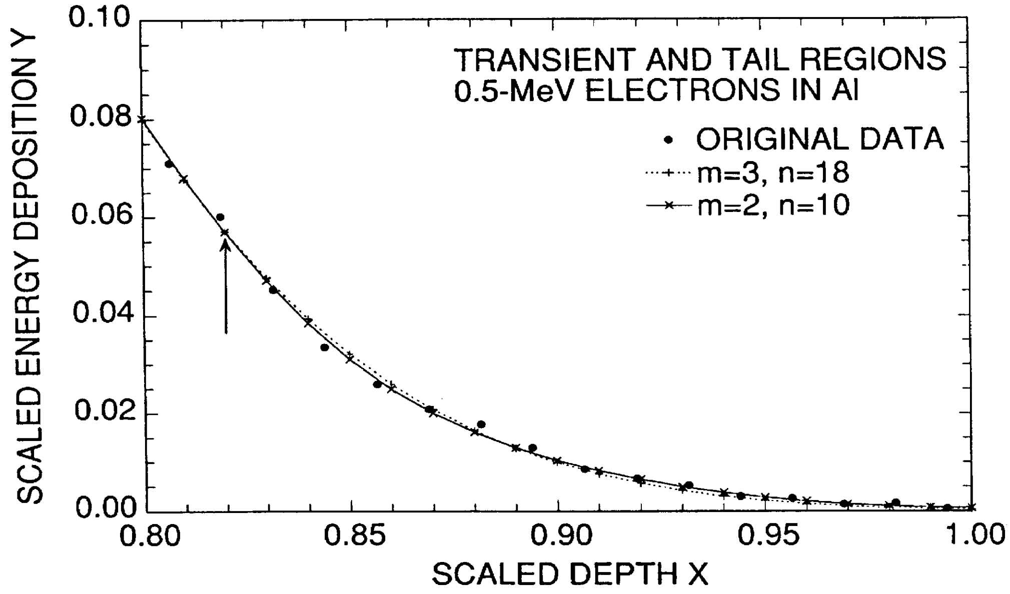


Fig. 4 Transient and tail regions of energy-deposition distribution, showing the connection of the results of different choices of $m$ and $n$. In the region on the right-hand side of the arrow, the results given by the solid line are adopted instead of the dotted line. The incident electron energy is 0.5 MeV, and the absorber material is aluminum.

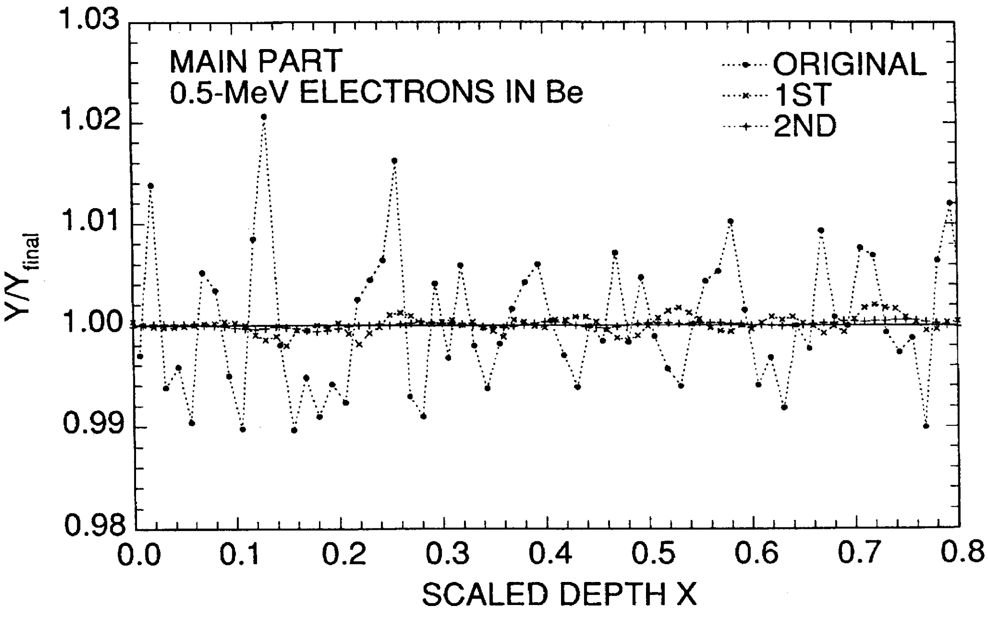


Fig. 5 Ratios of the original, the first interpolated and the second interpolated values of the main part of energy-deposition distribution to the final values, plotted as a function of scaled depth. The incident electron energy is 0.5 MeV, and the absorber material is beryllium.

repeated smoothing and the high degree of smoothness obtained by the present code.

A requirement for good smoothing is that errors caused by smoothing, if any, should be negligible compared with the fluctuations of the data. A check of this point is especially important when the fluctuations are small as in the present application. The rms of $(Y/Y_{\text{final}} - 1)$ for the original data given above, $6.7 \times 10^{-3}$, is not worse than the rms of the relative statistical uncertainties of the data, $7.0 \times 10^{-3}$, evaluated by the Monte Carlo code. This implies that the requirement mentioned is satisfied in the present application.

In addition to examples given here, we have successfully applied the code to widely different energy-deposition distributions of electrons[10], whose initial energies ranged between 0.1 and 100 MeV. The code has been found to be useful not only for main distributions with different peak widths but also for regions going, with different degrees of rapidity, into a slowly changing bremsstrahlung tail. A few times of cut and try were sometimes necessary to choose appropriate values of $m$ and $n$ for the transient and tail regions. For the main distribution, however, it was easy to make a good choice of them by the visual inspection of the plot of data.

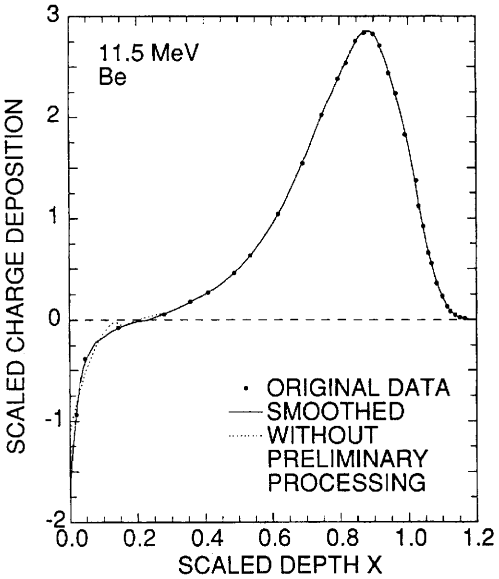


Fig. 6 The charge-deposition distributions of 11.5-MeV electrons incident on a beryllium absorber of effectively semi-infinite thickness. Points, experimental data; solid line, results of smoothing and interpolation; dotted line, results of smoothing and interpolation without preliminary processing.

### 3. 2 Charge-deposition distributions

In Fig. 6 an example of smoothing and interpolation of charge-deposition distributions by electrons is given. The results obtained by applying the present code (solid line) is compared with the original data (points). The initial electron energy is 11.5 MeV, and the absorber material is beryllium. The abscissa of the figure is again the scaled depth. The ordinate is the scaled charge deposition defined as the charge deposited per incident electron per unit depth multiplied by $r_0/e$, where $e$ is the charge on the electron. A sharp rise of the curve due to positive charge build-up is seen near the surface. This is similar to the distributions for part of which Frederickson and Woolf[2] failed to apply the spline-fit smoothing.

In contrast to the spline-fit smoothing, the present algorithm has the following advantages. Different set of $m$ and $n$ values can be used for the build-up region and the main distribution, and the smoothing processes can be repeated by changing $m$ and $n$, if necessary. However, if the same choice of $m$ and $n$ is made possible for the build-up and main regions by some preliminary processing, the application of the present code would become simpler. For the data in Fig. 6, therefore, we used preliminary processing, which was to insert a few data points in the build-up region by quadratic interpolation. This can also be made by the special use* of LESINT with $m=2$ and $n=3$. After inserting new points in the middle of neighboring points for the first four original data, we made a regular application of the present code to the whole region with $m=3$ and $n=5$. As can be seen from Fig. 6, the results obtained (solid line) are reasonable for both the build-up region and the main distribution. For comparison, the results without preliminary processing are shown by the dotted line in Fig. 6.

The present code has also been applied to other charge-deposition distributions by electrons for initial energies from 4 to 24 MeV. The results have shown that the code is useful even when the curve has a local rise of different degrees of sharpness.

* For doing this, POLFIT has to be modified to skip the calculation of the reduced $\chi^2$ for avoiding the error of dividing by zero.

## Acknowledgment

The author would like to thank Nissin – High Voltage Co. for continued financial support.

## References

1) International Mathematical & Statistical Libraries, Inc., in "IMSL Library", 8th ed., Vol. 2, Chapter I (1980).

2) A. R. Frederickson and S. Woolf, IEEE Trans. Nucl. Sci., **NS-28**, 4186 (1981).

3) J. S. Simonoff, "Smoothing Methods in Statistics", Springer Series in Statistics, Springer, New York (1996).

4) A. Savitzky and M. J. E. Golay, Anal. Chem., **36**, 1627 (1964).

5) W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical Recipes: The Art of Scientific Computing", Cambridge University Press, Cambridge (1992).

6) P. R. Bevington, "Data Reduction and Error Analysis for the Physical Sciences", McGraw-Hill, New York (1969).

7) P. Andreo, R. Ito and T. Tabata, "Tables of Charge- and Energy-Deposition Distributions in Elemental Materials Irradiated by Plane-Parallel Electron Beams with Energies between 0.1 and 100 MeV", Technical Report RIAST-UOP-TR 1, Res. Inst. Adv. Sci. Tech. Univ. Osaka Pref. (1992).

8) J. A. Halbleib, R. P. Kensek, G. Valdez, S. M. Seltzer and M. J. Berger, IEEE Trans. Nucl. Sci., **39**, 1025 (1992).

9) T. Tabata, R. Ito, S. Okabe and Y. Fujita, Phys. Rev., **B3**, 572 (1971).

10) T. Tabata, P. Andreo and R. Ito, At. Data & Nucl. Data Tables, **56**, 105 (1994).

11) T. Tabata, P. Andreo, K. Shinoda and R. Ito, Nucl. Instrum. Methods B, **95**, 289 (1995).

*Note added in proof*: With regard to distributions generated by Monte Carlo calculations, better results of smoothing would be obtained by applying the present method to cumulative distributions and taking the derivative of the polynomial used for each point.