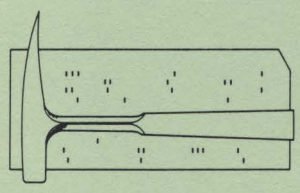
DANIEL F. MERRIAM, Editor

A GENERALIZED TWO-DIMENSIONAL REGRESSION PROCEDURE

By

JOHN R. DEMPSEY

Northern Natural Gas Company



COMPUTER CONTRIBUTION 2
State Geological Survey
The University of Kansas, Lawrence
1966

Editor's Remarks

Quantification is new to the earth sciences. Geologists have long thought in qualitative terms, although other physical sciences such as physics (geophysics) and chemistry (geochemistry) have been "mathematically oriented" for many years. Scientists use mathematical symbols and methods for several reasons: (1) for simplification of expression and for manipulation ease; (2) because of the manner in which general, and not just particular, solutions become apparent; and (3) because of the fundamentalism which underlies different problems and is revealed in their formulation. Two very important advantages of quantification are repeatibility and objectivity.

Acceptance of quantitative techniques by earth scientists, although slow, is readily apparent today, especially in publications, presentations of technical papers, and symposia. The number of geological publications dealing with various computer applications has increased since 1962 at an exponential rate. Several organizations are making computer programs available in regular or semi-regular publications, which, in addition to the Kansas Geological Survey, include: The Institute of Science and Technology at the University of Michigan; Seaver Laboratory at Pomona College; Office of Naval Research, Geography Branch, through Northwestern University, and Department of Oceanography, through the University of Washington; and U.S. Bureau of Mines. Many other organizations, including the Department of Geology at McMaster University, Department of Geography at the University of Michigan, Department of Mineralogy and Geochemistry at Pennsylvania State University, National Oceanographic Data Center, Sedimentology Research Laboratory at the University of Reading (England), and U.S. Geological Survey, have computer programs, along with operational instructions, available on request to interested persons. Undoubtedly, reports and programs of this type are available through other organizations as well.

The Kansas Survey is the only geological organization now known to be distributing computer program decks as well as data decks. The programs are written in ALGOL, BALGOL, FORTRAN II, and FORTRAN IV, and sold for a limited time at a nominal cost. The programs are for Burroughs B5500, Elliott 803C, IBM 1620, 7040, and 7090/1401 or 7094/1401 computer systems. A list of available decks is given below.

	ALGOL	BALGOL	FORTRAN II	FORTRAN IV
Marine Simulation (CC 1)		$20.00		
2D Regression (CC 2)	$10.00		$10.00	$10.00
Trend-3 (SDP 3)		$10.00		
Match-Coeff (SDP 4)			$ 2.00	
Correlation and distance Coeff (SDP 9)		$ 5.00 (each)		
Time-trend (SDP 12)			$ 5.00	$ 5.00
Covap (SDP 13)			$15.00	
Trend-3 (SDP 14)			$25.00	$25.00
Cross-Association (SDP 23)				$10.00
Single and double Fourier (SDP 24)		$ 5.00 $15.00		
Precambrian wells (SDP 25)	List of about 2,600 Precambrian wells	$50.00		
Trend-4 (SDP 26)			$ 7.50	
Sediment analysis (SDP 28)			$10.00	
4D Trend (KGS B171)		$10.00		
Conversion of T&R to Cartesian coordinates (Bull 170-3)			$ 5.00	$ 5.00

The number of computer-oriented papers being presented at different international, national, and regional meetings is increasing. In addition, many societies, such as the American Association of Petroleum Geologists, present symposia on computers within the framework of their regular meetings. An annual symposium on computers and computer applications in the mineral industries is sponsored jointly by the University of Arizona, Stanford University, Colorado School of Mines, the Pennsylvania State University, and the Society of Mining Engineers of AIME. Other institutions such as the University of Michigan and Oklahoma Research Institute are sponsoring computer conferences. In a program for continuing education, the American Association of Petroleum Geologists have guest lecturers who speak on the "state of the art" of computers and computer applications in the earth sciences.

This report, "A generalized two-dimensional regression procedure," by John R. Dempsey is the second in a series of COMPUTER CONTRIBUTIONS published by the Kansas Geological Survey. This series complements other Survey "regular" publications and allows new developments in computer methods and computer applications in the earth sciences to be presented in the quickest and easiest manner. The COMPUTER CONTRIBUTION series is an outgrowth of a very successful trial venture of publishing research results in regard to the rapidly expanding field of computer techniques in the Survey's series of Special Distribution Publications. The objectives of the new series are the same as those of the Special Distribution Publication series - that is, presenting as concisely and objectively as possible details of computer programs and their applications by geologists and petroleum engineers. Initial response to the COMPUTER CONTRIBUTION series indicates international acceptance and use of the information.

Comments and suggestions concerning the COMPUTER CONTRIBUTION series are welcome and should be addressed to the editor.

A GENERALIZED TWO-DIMENSIONAL REGRESSION PROCEDURE

By

JOHN R. DEMPSEY

INTRODUCTION

Frequently, in the solution of an engineering problem with the aid of a digital computer, the investigator wishes to use values in the calculations that originate from a chart or table of values. These values may have been derived by discrete variations of some complicated function or may have been derived experimentally. In the former case the trend is usually smooth in nature whereas the latter may be quite erratic. With the latter, he normally desires the most general trend of the tabulated data, but this often negates much of the experimental inconsistency. However, he must be very cautious with this assumption.

Several methods are at the programmer-engineer's disposal to represent data for automatic computation. A few of these are:

(1) Read in the value of the dependent variable at each step,
(2) Represent a table by matrices and use a table look-up procedure,
(3) Apply differencing techniques (polynomial approximation), and
(4) Compute normal least squares.

There are many inherent problems in using any of the above methods. The first one obviously is extremely time consuming and expensive, because the computer is idle while the value of the dependent variable corresponding to the specified independent variable is found.

The second method is often satisfactory when accuracy is desired. It normally requires a great deal of computer space, however, and is relatively slow in the evaluation of the desired value.

Differencing techniques are easily evaluated once the polynomial has been defined; however, there are many problems in evaluating the required differences to specify the polynomial. Some of the problems are: spacing, boundary conditions, and error accumulation.

The fourth is probably the best method, at least from the theoretical and utility standpoints. The theory states that the function must have a minimum deviation for any set of discrete data and for any order. It is also easily evaluated for digital computation.

Several problems are inherent in the computation of a normal least squares. One disadvantage is that a system of simultaneous equations must be reduced to evaluate the coefficients. The matrix formed by these equations is of the well-known Hilbert matrix form. It is extremely ill-conditioned and fails to converge at higher orders. The numerical round-off is also a problem during the matrix inversion. A second major disadvantage of the normal least-squares method is in the evaluation of the minimum variance for a set of different orders. Because the coefficients themselves are not analytically independent (i.e. they depend on both the order and set of data), one must evaluate a separate set of coefficients for each order and then proceed to evaluate the associated variance.

Due to the many problems inherent to function approximation by the above techniques, an alternate technique should be appropriate. The method described here or use of the procedure utilizing this method will eliminate or partly nullify the adverse conditions described above.

MATHEMATICAL DESCRIPTION

A family of functions may be defined as being an orthogonal system on (a, b), if for each pair of distinct members, ϕ_k, ϕ_j, $k \neq j$;

$$(\phi_i(x), \phi_k(x)) = \int_a^b w(x)\,\phi_i(x)\,\phi_k(x)\,dx = 0 \quad k \neq 0 \qquad (1)$$

where

w(x) = some arbitrary weighting function.

Assuming that the family contains no null functions, i.e., for any real function there exists nowhere the following relation:

$$(\phi_i, \phi_k) \leq 0 \quad i = k \qquad (2)$$

Using the above criteria (1) and (2), one can describe a linearly independent system $\phi_i(x)$ such as:

$$C_o\,\phi_o(x) + C_1\,\phi_1(x) + \ldots + C_k\,\phi_k(x) = 0 \qquad (3)$$

To prove the linear independence of (3) one can multiply successively (3) by $\phi_i(x)$, $i = 0, 1, 2, \ldots k$. Taking the first multiplication as an example,

$$C_o\phi_o(x)\,\phi_o(x) + C_1\phi_1(x)\,\phi_o(x) + \ldots + C_k\phi_k(x)\phi_o(x) = 0$$

Every product (known as the Kronecker delta), $\phi_i(x)\phi_j(x)$, where $i \neq j$, is zero by (1). Using (2),

$$C_o\phi_o(x)\,\phi_o(x) = 0 \tag{4}$$

and, $\phi_o(x)\phi_o(x)$ cannot be zero; therefore $C_o=0$. Repeating this analogy for i = 0, 1, 2, ...,k,

$$C_o = C_1 = \ldots = C_k = 0 \tag{5}$$

LEAST-SQUARES PROPERTY

It is shown that a system of orthogonal functions is indeed linearly independent with respect to the scalar products. To use this system for the approximation of a real function or discrete sets of data, it must be proved that the integral or sum of the weighted square of the error does, in fact, exhibit a minimum over (a,b).

Let f(x) be expanded in a linear combination of orthogonal polynomials such as

$$f(x) = A_oP_o(x) + A_1P_1(x) + \ldots + A_nP_n(x) + \sum_{i=n+1}^{\infty} A_iP_i(x) \tag{6}$$

However, for practicality the finite series is taken up to i = n.

$$r_n(x) = \sum_{0}^{\infty} A_iP_i(x) - \sum_{0}^{n} A_iP_i(x) \tag{7}$$

where $r_n(x)$ = remainder or truncation error. Rewriting (7) for convenience,

$$r_n(x) = f(x) - S_n(x) \tag{8}$$

where

$S_n(x)$ = partial sum or finite series through the n^{th} degree. Repeating the process with which linear independency was proved,

$$\int_a^b w(x)f(x)P_k(x)dx = A_k \tag{9}$$

since

$$(f(x),\ P_k(x)) = A_o\,(P_o(x),\ P_k(x)) + A_1(P_1(x),\ P_k(x)) + \ldots + A_k(P_k(x),\ P_k(x)) + \ldots \tag{10}$$

and

$$\begin{aligned}(P_i(x),\ P_i(x)) &= 0 \qquad i \neq j\\ (P_i(x),\ P_i(x)) &= C \qquad i \neq j\end{aligned} \tag{11}$$

Now let $\lambda_n(x)$ be any arbitrary polynomial of degree $\leq$ n such that

$$\lambda_n(x) - S_n(x) = \sigma_n(x) = \sum_{0}^{n} d_kP_k(x) \tag{12}$$

and $\lambda_n(x) = S_n(x)$ when all the d_k's are zero, i.e. no error exists. If (8) and (12) are combined,

$$f(x) - \lambda_n(x) = r_n(x) - \sigma_n(x) \tag{13}$$

which describes the error term.

The integral of the square of the error term in using $\lambda_n(x)$ to approximate f(x) weighted over (a,b) by w(x) is

$$\int_a^b w(x)\left[f(x) - \lambda_n(x)\right]^2 dx \int_a^b w(x)\left[r_n(x) - \sigma_n(x)\right]^2 dx \tag{14}$$

By expanding (14) to

$$\int_a^b w(x)\left[f(x) - \lambda_n(x)\right]^2 dx - \int_a^b w(x)\left[r_n(x)\right]^2 dx - 2\int_a^b w(x)r_n(x)\,\sigma_n(x)dx + \int_a^b w(x)\left[\sigma_n(x)\right]^2 dx \tag{15}$$

it is shown that

$$\int_a^b w(x)r_n(x)\ \ \sigma_n(x)dx = 0 \tag{16}$$

By replacing $\sigma_n(x)$ by $\sum_{0}^{n} d_kP_k(x)$ which now involves scalar products of unlike degree (all terms in $r_n(x)$ are of degree n+1 or greater), the following can now be written,

$$\int_a^b w(x)\,\sigma_n{}^2(x)\,dx = \sum_{0}^{n} d_k{}^2 \tag{17}$$

because all scalar products are zero except those involving polynomials of the same degree, i.e., $(P_k(x), P_k(x))^2 = d_k{}^2$. From (13), (14), (15), (16), and (17) the following inequality, which shows the minimum property of the approximation, can be written

$$\int_a^b w(x)\left[f(x) - \lambda_n(x)\right]^2 dx > \int_a^b w(x)\left[r_n(x)\right]^2 dx \tag{18}$$

except for the isolated case when all d_k's = 0 and then equality holds.

GENERATION OF APPROXIMATING POLYNOMIALS

Using the approximating function to f(x),

$$Y = \lambda_n(x) = b_oP_o(x) + b_1P_1(x) + \ldots + b_nP_n(x) \tag{19}$$

where P_i is an orthogonal polynomial of degree i. The standard error term used in least-squares minimization is

$$\epsilon = Y_i - \sum_{k=0}^{n} b_kP_k(x_i) \tag{20}$$

Squaring this term and differentiating with respect to b_k and setting the resulting equations to zero, the following set of n simultaneous equations is obtained:

$$b_o\sum_{1}^{N} P_o(x_i)P_o(x_i) + b_1\sum_{1}^{N} P_o(x_i)P_1(x_i) + \ldots$$

$$+ b_n \sum_1^N P_o(x)P_n(x_i) = \sum_1^N Y_i P_o(x_i)$$

$$b_o \sum_1^N P(x_i)P_o(x_i) + b_1 \sum_1^N P(x_i)P_1(x_i) + \ldots$$

$$+ b_n \sum_1^N P(x_i)P_n(x_i) = \sum_1^N Y_i P_1(x_i)$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$b_o \sum_1^N P_o(x_i)P_n(x_i) + b_1 \sum_1^N P_1(x_i)P_n(x_i) + \ldots$$

$$+ b_n \sum_1^N P_n(x_i)P_n(x_i) = \sum_1^N Y_i P_n(x_i) \qquad (21)$$

where

N = number of data points.

Applying (1) to the system, everything except the major diagonal vector and constant vector is reduced to zero. Thus, the coefficients b_o, b_1, ... b_n can be evaluated independently with only one division per coefficient.

After the polynomials have been generated, the evaluation of the coefficients $b_o, b_1, \ldots, b_n$ is relatively simple. Therefore, the automatic generation of the polynomials themselves, $P_o(x)$, $P_1(x)$, ..., $P_n(x)$ would be of great value for digital computation.

There are many methods of generating orthogonal polynomials (Hamming, 1962; Sansone, 1959). A method described by Householder (1953) is suited for computer usage. In this method, only two polynomials need be known to start the generation procedure.

To illustrate the generation of an automatic recursion formula, begin with the generalized formula,

$$P_{i+1}(x) = xP_i(x) - \alpha_{i+1}P_i(x) - \beta_i P_{i-1}(x) \qquad (22)$$

This formula is multiplied through by $P_i(x)$ which, after reducing and applying (1), gives

$$0 = (x - \alpha_{i+1})P_i(x)P_i(x) \qquad (23)$$

Thus for a set of discrete data

$$\alpha_{i+1} = \frac{\sum_1^N x_i \left[P_i(x_i) \right]^2}{\sum_1^N \left[P_i(x_i) \right]^2} \qquad (24)$$

To evaluate β_i, the above process is repeated using $P_{i-1}(x)$ instead of $P_i(x)$. This gives (25) for a set of discrete data points.

$$\beta_i = \frac{\sum_1^N x_i \left[P_i(x_i) \right]^2}{\sum_1^N \left[P_{i-1}(x_i) \right]^2} \qquad (25)$$

NOTE: This illustration was derived for w(x) = 1. However, the same analogy can be used for w(x) = f(x). The first two orthogonal polynomials needed to start the generalized recursion formula may be generated by any technique, such as Gram-Schmidt (in Kunz, 1957) or Rushton (1951).

USE OF PROCEDURES "ORTH", "EVAL" AND "COEF"

The procedure ORTH generates the orthogonal polynomials to be used in the least-squares fit. These are normalized on the interval (-1, +1). Also, two arrays of recursion coefficients which will enable the calling of EVAL to find any specific value of the dependent variable are generated in the process. These must be available when EVAL is called.

If the regression portion of the program is desired, KSELCT is set equal to +1 and MAXO is set equal to the highest order to be included in the analysis. This causes the program to compute an array of variances σ_i^2 (Lapidus, 1962)

$$\sigma_i^2 = \frac{\sum_{k=1}^N (Y_k - Yc_k)^2}{(N - i + 1))} \qquad (26)$$

where

Y_k = k^{th} value of dependent variable used,

Yc_k = k^{th} value of dependent variable calculated by use of the orthogonal function generated by using EVAL with $XC = X_k$,

N = number of data points used, and

i = index equaling the order of the orthogonal function used to evaluate Yc_{k_2}.

The minimum value of σ_i^2 is then derived, and the value of i for this value determines the optimum regression order from an independent statistical test function.

If the user desires a specific order without regard to minimum variance, KSELCT is set equal to zero and MO becomes the order of the least-squares fit. Intermediate output may be obtained by setting KOUT equal to 1 (see sample output).

The least-squares polynomial is of the form

$$Y = b_1P_1(x) + b_2P_2(x) + b_3P_3(x) + \ldots + b_nP_n(x) \quad (27)$$

where

b_i's = least-squares coefficients,

p_i's = orthogonal polynomials of order i, and

n = order of the analysis.

Because it is difficult to evaluate (27) into an ordinary polynomial where each coefficient is multiplying f(x) of only one order, the procedure EVAL was written to automatically evaluate a form of (27) at any x. If for some reason, this polynomial of ordinary form is desired, it can be obtained through the use of the procedure COEF. The technique used in COEF is described in the Appendix. The equation used for analytic interpolation is of the following form:

$$Y = c_1 + c_2z + c_3z^2 + c_4z^3 + \ldots + c_nz^n \quad (28)$$

where

c_i = i^{th} polynomial coefficient,

z = (2x - (A-BB))/(BB-A),

A = smallest value of dependent array,

BB = largest value of independent array, and

n = order of analysis.

A and BB are available globally from ORTH.

SAMPLE CALL SEQUENCE

```
ORTH (X, Y, N, MO, B, KOUT, ALP, BET, KSELCT,
  MAXO, A, BB, FILENAME)
EVAL (B, MO, ALP, BET, A, BB, XC)
COEF (ALP, BET, B, MO, C)
```

SYMBOLIC DICTIONARY

(ORTH)

Variable	Type	S/A	I/O	Units	Comments
X	R	A	I	NA	An array of values of the independent variable.
Y	R	A	I	NA	An array of values of the dependent variable.
N	I	S	I	NA	Number of discrete data points.
MO	I	S	I	NA	Order of the orthogonal polynomial to be used when regression is not used.
B	R	A	O	NA	An array of the orthogonal coefficient.
KOUT	I	S	I	NA	Switch for the intermediate output (+1 is on, o is off).
ALP	R	A	O	NA	Array of recursive coefficients.
BET	R	A	O	NA	Array of recursive coefficients.
KSELCT	I	S	I	NA	Switch to use the regression portion (+1 is on, o is off).
MAXO	I	S	I	NA	Maximum order to be used by the regression routine.
A	R	S	O	NA	Smallest value in independent data array.
BB	R	S	O	NA	Largest value in independent data array.
C	R	A	O	NA	Ordinary polynomial coefficients
FILENAME					Identifier of the output file.

(EVAL)

Variable	Type	S/A	I/O	Units	Comments
B	R	A	I	NA	Array of orthogonal coefficients.
MO	I	S	I	NA	Order of highest approximating polynomial.
ALP	R	A	I	NA	Array of recursion coefficients.
BET	R	A	I	NA	Array of recursion coefficients.
A	R	S	I	NA	Smallest value in independent variable data array.
BB	R	S	I	NA	Largest value in independent variable data array.
XC	R	S	I	NA	Value of the independent variable for which the value of the respective dependent variable is desired.

Note: The variables B, MO, ALP, BET, A, and BB are output from the procedure ORTH.

Variable	Type	S/A	I/O	Units	Comments
ALP	R	A	I	NA	Array of recursive coefficients.
BET	R	A	I	NA	Array of recursive coefficients.
B	R	A	I	NA	Array of orthogonal coefficients.
MO	I	S	I	NA	Order of the orthogonal polynomial which is also order of ordinary polynomial.
C	R	A	O	NA	Array of ordinary polynomial coefficients.

Note: The variables ALP, BET, B, and MO are output from the procedure ORTH.

PROGRAM LISTING

```
PROCEDURE ORTH(X,Y,N,MO,B,KOUT,ALP,BET,KSELCT,MAXO,A,
    BB,C,FILENAME);
VALUE X,Y,N,MO,KOUT,KSELCT,MAXO;
REAL ARRAY X,Y,B,ALP,BET,C[1];
INTEGER N,MO,KOUT,KSELCT,MAXO;
FILE FILENAME;
REAL A,BB;
      BEGIN COMMENT INSERT FORMATS HERE AND EVAL AND COEF
          SOURCES;
REAL PROCEDURE EVAL(B,MO,ALP,BET,A,BB,XC);
VALUE B,MO,ALP,BET,A,BB;
REAL ARRAY ALP,BET,B[1];
REAL A,BB,XC;
INTEGER MO;
      BEGIN
            INTEGER M,I;
           REAL ARRAY PO[1:MO+1];
            REAL ZN,SUM,PM;
            ZN ← (2×XC - (A+BB)) / (BB - A);
            SUM ← 0;
           PO[1] ← 1;
            M ← MO+1;
            FOR I ← 2 STEP 1 UNTIL M DO
      BEGIN
            IF I=2 THEN PM←0  ELSE PM ← PO[I - 2];
          PO[I] ← (ZN - ALP[I]) × PO[I-1] - BET[I]× PM;
      END;
            FOR I←1 STEP 1 UNTIL M DO
            SUM ← SUM + B[I]x PO[I];
            EVAL ← SUM;
      END EVAL;
PROCEDURE COEF (ALP,BET,B,MO,C);
VALUE ALP,BET,B,MO;
REAL ARRAY ALP,BET,B,C[1];
INTEGER MO;
      BEGIN
            REAL ARRAY P[0:MO+1, 0:MO+1];
            INTEGER I,J,K,L;
```

```
            FOR J ←1 STEP 1 UNTIL MO + 1 DO
            FOR K ←1 STEP 1 UNTIL (MO+ 1) DO
            P[K,J] ←    0;
            FOR K←1 STEP 1 UNTIL (MO+1) DO
            P[K,K] ← 1;
            FOR K ← 2 STEP 1 UNTIL (MO+ 1) DO
      BEGIN
            L ← K - 1;
            FOR I←1 STEP 1 UNTIL L DO
      P[J,K] ←P[J-1,K-1] - ALP[K] × P[J,K-1] - BET[K] ×
          P[J,K-2];
END;
      FOR I ←1 STEP 1 UNTIL (MO + 1) DO
BEGIN
      C[I] ← 0;
      FOR K←I STEP 1 UNTIL (MO + 1) DO
      C[I] ← C[I] + B[K] × P[I,K];
END;
END COEF;
      INTEGER I, J,       L,KV,MOO,MOO1,   MOP ,M;
    REAL ARRAY P [1:MAXO+1,1:N], SIG 2 [1:MAXO], Z [1: N];
      REAL XMAX,XMIN,AN,ANN,AD,BD,BN,PM,EMOP,VMAX,VAR,
          PERC,S,R,XN,YN,SUM,      EM,EN;
      LABEL L1,L2;
      FORMAT OUT INFO 1 ("RECURRENCE COEFFICIENTS"/),
          INFO 2 (" BETA",X12," ALPHA",X11," ORTH COEF",
              X7,"POLY COEF"),
          INFO 3 (4(E 12.5, X 5)),
          INFO 4 (/" ORDER OF BEST APPROXIMATING ORTHOGO",
                    " NAL FUNCTION = ",I 3/"(BASED ON MINIMU",
                    " MUM VARIANCE)    VARIANCE = ", E13.6),
          INFO 5 ("HIGHEST ORDER POLYNOMIAL USED IN THIS",
              " ANALYSIS = ",I3),
          INFO 6 (/"HIGHEST ORDER OF ORTHOGONAL POLYNOMI",
              " AL = ",I3,X5/"VARIANCE = ",E 13.6),
          INFO 7 (/" X-DATA",X8," Y-DATA",X8," Y-CALC",X8,
              " DIFF",X10," PERCENT-DIFF"/),
          INFO 8 (5 (E11.4, X 4));
      EM←M←MO+1;
      MOO←MO;
      EN ← N;
      COMMENT LOCATION OF XMAX AND XMIN FOR NORMALIZATION;
      XMIN ←XMAX← X[1];
      FOR I←2 STEP 1 UNTIL N DO
BEGIN
      IF XMIN > X[I] THEN XMIN ←X[I];
      IF X[I] > XMAX THEN XMAX ←X[I];
END;
      A ←XMIN;
      BB ←XMAX;
      COMMENT NORMALIZATION OF INDEPENDENT VARIABLES;
      FOR I ←1 STEP 1 UNTIL N DO
      Z[I] ←(2 × X[I] - (A+ BB))/(BB - A);
      FOR I←1 STEP 1 UNTIL N DO
```

```
            P[1,I] ← 1;
            BET[1] ← ALP[1] ←          BET[2] ← 0;
            IF KSELCT = 1 THEN M ← MAXO + 1;
            FOR J ←2 STEP 1 UNTIL M DO
      BEGIN
            AN ← ANN ← AD ← BN ← BD ← 0;
            FOR I ←1 STEP 1 UNTIL N DO
      BEGIN
            ANN ← ANN + Z[I] x P[J-1,I]*2;
            AD ← AD + P[J-1,I]*2;
            ALP[J] ← ANN / AD;
            IF J > 2 THEN
      BEGIN
            BN ← BN + Z[I] x P[J-1,I] x P[J-2,I];
            BD ← BD + P[J-2,I]*2;
            BET[J] ← BN / BD;
            GO TO L1;
      END;
L1:         PM ← 0;
            FOR L ←1 STEP 1 UNTIL N DO
      BEGIN IF J>2 THEN PM ← P[J-2,L];
            P[J,L] ← (Z[L] - ALP[J]) x P[J-1,L] - BET[J] x PM;
      END
      END; END;
            FOR J ←1 STEP 1 UNTIL M DO
      BEGIN
            S ← R ← 0;
            FOR L ←1 STEP 1 UNTIL N DO
      BEGIN
            S ← S + Y[L] x P[J,L];
            R ← R + P[J,L]*2;
      END;
            B[J] ← S / R;
      END;
            IF KSELCT ≤ 0 THEN MOP ← MO ELSE MOP ← 1;
L2:
            SUM ← 0;
            FOR I ←1 STEP 1 UNTIL N DO
      BEGIN
            YN ← EVAL(B,MOP,ALP,BET,A,BB,X[I]);
            COMMENT INSERT CALL FOR AN EVALUATION PROCEDURE HERE;
            VAR ← YN - Y[I];
            SUM ← SUM + VAR*2;
      END;
            EMOP ← MOP;
            SIG2[MOP] ← SUM / (EN - (EMOP + 1));
            IF KSELCT = 1 THEN
      BEGIN
            IF (MOP < MAXO) THEN
      BEGIN
            MOP ← MOP + 1;
            GO TO L2;
      END;
            COMMENT LOCATION OF MINIMUM VARIANCE;
```

```
            VMAX ← SIG 2 [ 1 ];
            KV ← 1;
            FOR I←2 STEP 1 UNTIL MAXO DO
      BEGIN IF VMAX > SIG 2 [I ]  THEN
      BEGIN
            KV←I;
            VMAX ←SIG 2 [ I ];
      END;
      END;
            MOO← KV;
            MOP ← MOO;
      END;
            IF KOUT = 1 THEN
      BEGIN
            WRITE ( FILENAME,INFO1 );
            WRITE ( FILENAME,INFO2 );
            COMMENT  CALL POLY COEFFICIENT GENERATOR HERE;
            COEF ( ALP,BET,B,MOP,C );
            WRITE ( FILENAME, INFO3,FOR I←1 STEP 1 UNTIL (MOO+ 1)
                DO[ BET [I ] ,ALP [I ] ,B [I] ,C [I ] ] );
            IF KSELCT = 1 THEN
      BEGIN
            WRITE ( FILENAME,INFO4,MOP,SIG2 [ MOP]);
            WRITE ( FILENAME,INFO5,MAXO);
      END ELSE
            WRITE (FILENAME, INFO6,MO,SIG2 [ MO]);
            WRITE (FILENAME, INFO7);
            FOR I←1 STEP 1 UNTIL N DO
      BEGIN
            YN ← EVAL (B,MOP,ALP,BET,A,BB,X [I ]);
            VAR ← YN - Y [I ];
COMMENT     IF Y [I ] EQUALS 0.0, THE PERCENT DIFFERENCE IS SET
            EQUAL TO 99999999999;
            IF Y [I ] ≠ 0.0 THEN
            PERC ← ABS(VAR) × 100/Y[I]
            ELSE PERC ← 99999999999;
      WRITE (FILENAME,INFO8,X [I ],  Y[I ],YN,VAR,PERC);
      END;
      END;
      END ORTH;
```

SAMPLE OUTPUT

RECURRENCE COEFFICIENTS

BETA	ALPHA	ORTH COEF	POLY COEF
0.00000@+00	0.00000@+00	4.53437@+00	4.51639@+00
0.00000@+00	9.67169@-02	1.85892@-01	1.85892@-01

HIGHEST ORDER OF ORTHOGONAL POLYNOMIAL = 1

VARIANCE = 7.986890@-04

X-DATA	Y-DATA	Y-CALC	DIFF	PERCENT-DIFF
6.3703@+00	4.6622@+00	4.6286@+00	-3.3572@-02	7.2009@-01
6.3030@+00	4.6055@+00	4.5521@+00	-5.3455@-02	1.1607@+00

6.2813@+00	4.5724@+00	4.5274@+00	-4.4971@-02	9.8354@-01
6.3800@+00	4.6312@+00	4.6396@+00	8.3872@-03	1.8110@-01
6.4352@+00	4.6723@+00	4.7023@+00	2.9950@-02	6.4101@-01
6.3795@+00	4.6260@+00	4.6390@+00	1.3033@-02	2.8173@-01
6.3566@+00	4.6081@+00	4.6130@+00	4.9152@-03	1.0667@-01
6.2304@+00	4.4527@+00	4.4697@+00	1.7015@-02	3.8213@-01
6.1787@+00	4.3891@+00	4.4109@+00	2.1844@-02	4.9770@-01
6.1079@+00	4.3246@+00	4.3305@+00	5.9448@-03	1.3747@-01
6.3306@+00	4.5555@+00	4.5835@+00	2.7970@-02	6.1398@-01
6.2019@+00	4.4321@+00	4.4373@+00	5.2562@-03	1.1859@-01
6.1804@+00	4.4152@+00	4.4129@+00	-2.3185@-03	5.2511@-02

SAMPLE OUTPUT

KOUT = 1
KSELCT = 1
MAXO = 7

RECURSION COEFFICIENTS

BETA	ALPHA	POLY COEF
0.00000@-99	0.00000@-99	5.21666@-01
0.00000@-99	-7.17504@-02	-5.76652@-01
2.08837@-01	-1.01705@-02	1.32841@-01

ORDER OF BEST APPROXIMATING ORTHOGONAL FUNCTION = 2
(BASED ON MINIMUM VARIANCE) VARIANCE = 2.320492@-03
HIGHEST ORDER POLYNOMIAL USED IN THIS ANALYSIS = 7

X-DATA	Y-DATA	Y-CALC	DIFF	PERCENT-DIFF
5.1900@-01	2.0000@-02	1.9717@-02	-2.8225@-04	-1.4112@-00
2.5400@-01	5.5000@-01	5.2218@-01	-2.7811@-02	-5.0565@-00
4.2200@-01	1.0000@-01	1.6500@-01	6.5002@-02	6.5002@+01
2.4700@-01	5.7000@-01	5.3997@-01	-3.0023@-02	-5.2673@-00
4.6700@-01	1.3000@-01	9.2054@-02	-3.7945@-02	-2.9189@+01
2.2100@-01	5.8000@-01	6.0807@-01	2.8077@-02	4.8409@-00
4.1100@-01	2.4000@-01	1.8429@-01	-5.5704@-02	-2.3210@+01
2.4000@-01	6.0000@-01	5.5799@-01	-4.2004@-02	-7.0006@-00
3.4100@-01	2.6000@-01	3.2050@-01	6.0508@-02	2.3272@+01
2.2100@-01	6.1000@-01	6.0807@-01	-1.9224@-03	-3.1515@-01
3.4300@-01	2.7000@-01	3.1629@-01	4.6294@-02	1.7146@+01
2.4400@-01	6.1000@-01	5.4767@-01	-6.2329@-02	-1.0217@+01
3.4300@-01	3.0000@-01	3.1629@-01	1.6294@-02	5.4314@-00
2.1100@-01	6.2000@-01	6.3512@-01	1.5123@-02	2.4393@-00
3.5700@-01	3.2000@-01	2.8732@-01	-3.2674@-02	-1.0210@+01
2.2100@-01	6.3000@-01	6.0807@-01	-2.1922@-02	-3.4797@-00
3.3500@-01	3.4000@-01	3.3326@-01	-6.7345@-03	-1.9807@-00
2.2600@-01	6.3000@-01	5.9473@-01	-3.5267@-02	-5.5980@-00
2.7600@-01	3.5000@-01	4.6779@-01	1.1779@-01	3.3656@-01
2.3000@-01	6.3000@-01	5.8414@-01	-4.5858@-02	-7.2791@-00
2.8200@-01	4.4000@-01	4.5336@-01	1.3363@-02	3.0371@-00
1.3700@-01	8.4000@-01	8.5000@-01	1.0005@-02	1.1911@-00
2.8500@-01	4.6000@-01	4.4620@-01	-1.3790@-02	-2.9978@-00
1.1100@-01	8.8000@-01	9.3166@-01	5.1669@-02	5.8715@-00
2.6700@-01	4.7000@-01	4.8977@-01	1.9771@-02	4.2067@-00
1.0400@-01	1.0400@-00	9.5420@-01	-8.5796@-02	-8.2496@-00
2.9400@-01	4.8000@-01	4.2500@-01	-5.4995@-02	-1.1457@+01

6.7900@-02	1.0700@-00	1.0741@-00	4.1043@-03	3.8357@-01
2.3700@-01	4.9000@-01	5.6578@-01	7.5989@-02	1.5467@+01
4.5600@-02	1.1200@-00	1.1512@-00	3.1258@-02	2.7909@-00

SAMPLE OUTPUT

RECURRENCE COEFFICIENTS

BETA	ALPHA	ORTH COEF	POLY COEF
0.00000@+00	0.00000@+00	1.52839@+03	1.59465@+03
0.00000@+00	1.45519@-12	1.18879@+03	1.18879@+03
4.07407@-01	0.00000@+00	-1.62644@+02	-1.62644@+02

HIGHEST ORDER OF ORTHOGONAL POLYNOMIAL = 2
VARIANCE = 8.596587@+02

X-DATA	Y-DATA	Y-CALC	DIFF	PERCENT-DIFF
5.0000@+03	2.6512@+03	2.6208@+03	-3.0368@+01	1.1455@+00
4.5000@+03	2.3701@+03	2.4209@+03	5.0776@+01	2.1423@+00
4.0000@+03	2.1947@+03	2.2049@+03	1.0178@+01	4.6374@-01
3.5000@+03	1.9885@+03	1.9728@+03	-1.5692@+01	7.8911@-01
3.0000@+03	1.7493@+03	1.7247@+03	-2.4567@+01	1.4044@+00
2.5000@+03	1.4767@+03	1.4606@+03	-1.6170@+01	1.0950@+00
2.0000@+03	1.1759@+03	1.1803@+03	4.3955@+00	3.7379@-01
1.5000@+03	8.6133@+02	8.8402@+02	2.2686@+01	2.6338@+00
1.0000@+03	5.5268@+02	5.7165@+02	1.8973@+01	3.4329@+00
5.0000@+02	2.6343@+02	2.4322@+02	-2.0210@+01	7.6719@+00

SAMPLE OUTPUT

RECURRENCE COEFFICIENTS

BETA	ALPHA	ORTH COEF	POLY COEF
0.00000@+00	0.00000@+00	2.32434@+04	2.33007@+04
0.00000@+00	1.45519@-12	1.93718@+04	1.98310@+04
4.07407@-01	0.00000@+00	-1.40644@+02	-1.40644@+02
3.16049@-01	-1.41269@-12	-6.34740@+02	-6.34740@+02

HIGHEST ORDER OF ORTHOGONAL POLYNOMIAL = 3
VARIANCE = 2.100886@+00

X-DATA	Y-DATA	Y-CALC	DIFF	PERCENT-DIFF
4.9000@+02	4.2358@+04	4.2356@+04	-1.2163@+00	2.8714@-03
4.4100@+02	3.8339@+04	3.8341@+04	2.2355@+00	5.8309@-03
3.9200@+02	3.4165@+04	3.4166@+04	4.0903@-01	1.1972@-03
3.4300@+02	2.9873@+04	2.9872@+04	-1.3397@+00	4.4847@-03
2.9400@+02	2.5503@+04	2.5502@+04	-1.2293@+00	4.8203@-03
2.4500@+02	2.1096@+04	2.1096@+04	1.4531@-01	6.8879@-04
1.9600@+02	1.6697@+04	1.6698@+04	1.1670@+00	6.9892@-03
1.4700@+02	1.2348@+04	1.2349@+04	6.1728@-01	4.9989@-03
9.8000@+01	8.0911@+03	8.0902@+03	-9.3292@-01	1.1530@-02
4.9000@+01	3.9636@+03	3.9638@+03	1.4408@-01	3.6351@-03

REFERENCES

Hamming, R.W., 1962, Numerical methods for scientists and engineers: McGraw-Hill Co., Inc., New York, 411 p.
Householder, A.S., 1953, Principles of numerical analysis: McGraw-Hill Co., Inc., New York, 274 p.
Kunz, K.S., 1957, Numerical analysis: McGraw-Hill Co., Inc., New York, 381 p.
Lapidus, L., 1962, Digital computation for chemical engineers: McGraw-Hill Co., Inc., New York, 407 p.
Rushton, S., 1951, On least squares fitting by orthonormal polynomials using the Choleski method: Jour. Royal Stat. Soc. (B), v. 13, p. 92-99.
Sansone, G., 1959, Orthogonal functions: Intersciences Publishers, New York, 411 p.

APPENDIX

GENERATION OF ACTUAL POLYNOMIALS

The orthogonal polynomials are generated by the relationship

$$P_{j+1} = (X - \alpha_{j+1})P_j - \beta_{j+1}P_{j-1} \qquad (1)$$

where α_j = ALPHA(J) and β_j = BETA(J) in the sample output.

NOTE: The subscripts on α and β differ from those appearing in the mathematical description because these are written to correspond with the ALPHA and BETA arrays in the sample output.

The sample output previously described is used for this example. The data has been normalized over (-1,+1) by the following relationship,

$$Z = (2X - (A + BB))/(BB - A) \qquad (2)$$

where A = smallest value of the independent array (X-data).
BB = largest value of the independent array, and
Z = transformed independent variable on the interval (-1,+1).

First, P_1 is set equal to 1 to start the recursion relationship (1)

$$P_1 = 1.0.$$

Then

$$P_2 = (Z - \alpha_2)P_1 - \beta_2 P_o$$

since

$$\beta_2 = \text{BETA}(2) = 0,$$

$$P_2 = (Z - \alpha_2)P_1, \text{ and}$$

$$P_2 = (Z + 0.0717504).$$

This gives

$$P_3 = (Z - \alpha_3)P_2 - \beta_3 P_1$$

$$P_3 = (Z + 0.0101705)(Z + 0.0717504) - 0.208837\ (1.0)$$

$$P_3 = (Z^2 + 0.0819209Z - 0.208107).$$

Using a linear combination (3) of these orthogonal polynomials, the actual polynomial used to approximate the real function can be obtained

$$u = g(Z) = b_1P_1 + b_2P_2 + b_3P_3 + \ldots + b_{n+1}P_{n+1} \qquad (3)$$

where

b_j = POLYCOEF(J)

$$u = 0.521666 - 0.576652(Z + 0.0717504) + 0.132841(Z^2 + 0.0819209Z - 0.208107)$$

$$u = 0.452553 - 0.565770Z + 0.132841Z^2 \qquad (4)$$

The first data point is used to test the polynomial just generated.

X = 0.519, Y = 0.02

BB = 0.519, A = 0.0456

$$Z = (2(0.519) - 0.0456 - 0.519)/(0.519 - 0.0456)$$

Z = 1.0

Substituting Z = 1 into (4),

$$u = 0.452553 - 0.565770 + 0.132841$$

$$u = 0.019624$$

which agrees very closely to the value obtained by the computer.

KANSAS GEOLOGICAL SURVEY COMPUTER PROGRAM
THE UNIVERSITY OF KANSAS, LAWRENCE

PROGRAM ABSTRACT

Title (If subroutine state in title):

Generalized Two-Dimensional Orthogonal Regression Procedure

Computer: B5500, IBM 7040 Date: April 2, 1966

Programming language: ALGOL (Burroughs), FORTRAN II, IV

Author, organization: John R. Dempsey, Northern Natural Gas Company

Direct inquiries to: Author or

Name: D. F. Merriam Address: Kansas Geological Survey
University of Kansas, Lawrence, Kansas

Purpose/description: To enable selection of "best" fit of a set of observations without regeneration of lower order coefficients.

Mathematical method: Minimization property of Orthogonal function

Restrictions, range: Number of observations and maximum order depends on dimension statement in the FORTRAN version

Storage requirements:

Equipment specifications:

Memory 20K 40K 60K K

Automatic divide: Yes No Indirect addressing: Yes No

Other special features required

Additional remarks (include at author's discretion: fixed/float, relocatability; optional: running time, approximate number of times run successfully, programming hours) Both FORTRAN and ALGOL versions have been run many times with no problems.

COMPUTER CONTRIBUTIONS

Kansas Geological Survey
University of Kansas
Lawrence, Kansas

Daniel F. Merriam, Editor

Computer Contribution

1.	Mathematical simulation of marine sedimentation with IBM 7090/7094 computers, by J. W. Harbaugh, 1966	$1.00
2.	A generalized two-dimensional regession procedure, by J. R. Dempsey, 1966	$0.50

Special Distribution Publication

3.	BALGOL program for trend-surface mapping using an IBM 7090 computer, by J. W. Harbaugh, 1963	$0.50
4.	FORTRAN II program for coefficient of association (Match-Coeff) using an IBM 1620 computer, by R. L. Kaesler, F. W. Preston, and D. I. Good, 1963.	$0.25
9.	BALGOL programs for calculation of distance coefficients and correlation coefficients using an IBM 7090 computer, by J. W. Harbaugh, 1964	$0.75
11.	Trend-surface analysis of regional and residual components of geologic structure in Kansas, by D. F. Merriam and J. W. Harbaugh, 1964	$0.75
12.	FORTRAN and FAP program for calculating and plotting time-trend curves using an IBM 7090 or 7094/1401 computer system, by W. T. Fox, 1964	$0.75
13.	FORTRAN program for factor and vector analysis of geologic data using an IBM 7090 or 7094/1401 computer system, by Vincent Manson and John Imbrie, 1964	$1.00
14.	FORTRAN II trend-surface program for the IBM 1620, by D. I. Good, 1964	$1.00
15.	Application of factor analysis to petrologic variations of Americus Limestone (Lower Permian), Kansas and Oklahoma, by J. W. Harbaugh and Ferruh Demirmen, 1964	$1.00
23.	ALGOL program for cross-association of nonnumeric sequences using a medium-size computer, by M. J. Sackin, P. H. A. Sneath, and D. F. Merriam, 1965	$0.75
24.	BALGOL program and geologic application for single and double Fourier series using IBM 7090/7094 computers, by F. W. Preston and J. W. Harbaugh, 1965	$1.00
26.	FORTRAN II trend-surface program with unrestricted input for the IBM 1620 computer, by R. J. Sampson and J. C. Davis, 1966	$0.50
27.	Application of factor analysis to a facies study of the Leavenworth Limestone (Pennsylvanian-Virgilian) of Kansas and environs, by D. F. Toomey, 1966.	$0.75
28.	FORTRAN II program for standard-size analysis of unconsolidated sediments, by J. W. Pierce and D. I. Good, 1966	$0.75

Report of Studies

170-3	Mathematical conversion of section, township, and range notation to Cartesian Coordinates, by D. I. Good, 1964.	$0.50

Bulletin

171	A computer method for four-variable trend analysis illustrated by a study of oil-gravity variations in southeastern Kansas, by J. W. Harbaugh, 1964.	$1.00

Reprints (available for limited time)

Pattern recognition studies of geologic structure using trend-surface analysis, by D. F. Merriam and R. H. Lippert (reprinted from Colorado School Mines Quarterly, v. 59, no. 4, 1964)	no charge
Trend-surface mapping of hydrodynamic oil traps with the IBM 7090/7094 computer, by J. W. Harbaugh (reprinted from Colorado School Mines Quarterly, v. 59, no. 4, 1964)	no charge
Finding the ideal cyclothem, by W. C. Pearn (reprinted from Symposium on cyclic sedimentation, D. F. Merriam, editor, Kansas Geological Survey Bulletin 169, v. 2, 1964)	no charge
Fourier series characterization of cyclic sediments for stratigraphic correlation, by F. W. Preston and J. H. Henderson (reprinted from Symposium on cyclic sedimentation, D. F. Merriam, editor, Kansas Geological Survey Bulletin 169, v. 2, 1964).	no charge
Fourier series analysis in geology, by J. W. Harbaugh and F. W. Preston (reprinted from College of Mines, Arizona University, v. 1, 1965)	no charge
Geology and the computer, by D. F. Merriam (reprinted from New Scientist, v. 26, no. 444, 1965)	no charge
Application of factor analysis to the Upper Cambrian Reagan Sandstone of central and northwest Kansas, by M. N. McElroy and R. L. Kaesler (reprinted from The Compass, v. 42, no. 3, 1965)	no charge
Quantitative comparison of contour maps, by D. F. Merriam and P. H. A. Sneath (reprinted from Journal of Geophysical Research, v. 71, no. 4, 1966)	no charge
Trend-surface analysis of stratigraphic thickness data from some Namurian rocks east of Sterling, Scotland, by W. A. Read and D. F. Merriam (reprinted from Scottish Journal of Geology, v. 2, pt. 1, 1966)	no charge