Computers Chem. Vol. 15, No. 1, pp. 59–72, 1991
Printed in Great Britain.

0097-8485/91 $3.00 + 0.00

GENERAL LINEAR REGRESSION ANALYSIS— APPLICATION TO THE ELECTRIC DIPOLE-MOMENT FUNCTION OF HCl

J. F. OGILVIE*

Department of Chemistry, National Tsing Hua University, Hsinchu 30043, Taiwan and Academia Sinica, Institute of Atomic and Molecular Sciences, P.O. Box 23-166, Taipei 10764, Taiwan

(*Received* 27 *February* 1989; *in revised form* 7 *March* 1990; *received for publication* 12 *June* 1990)

Abstract—The program GWLREG has been designed for the convenient and efficient treatment of general problems in linear regression, including both multiple linear and univariate polynomial forms. There is provision for transformation and weighting of input data, and output of indicators of goodness of fit. Two versions of the program GWLREG are presented, each based on a different computational algorithm and having slightly differing capabilities and performance. The program is applied to a new determination of the electric dipole-moment function of the diatomic molecule HCl from a critical assessment of the published experimental data.

1. INTRODUCTION

The minimum attributes of a satisfactory computer program for linear regression applications are the following:

(a) adequate accuracy;
(b) indicators of goodness of fit;
(c) simplicity of use and adaptation for particular problems;
(d) capability of weighting the input data;
(e) automatic transformation of input data;
(f) applicability of simple constraints, such as passage through the origin.

The author has been unable to discover a single program for microcomputers that has all these essential attributes and is applicable to both univariate polynomial and multiple uninomial linear regression. For this reason I have undertaken to develop the program GWLREG. For reasons of convenience, this program has been prepared in BASIC, a language commonly available on microcomputers, in order to generate in the most convenient analytic form a representation of these data. In this paper we discuss the algorithms and their new implementation in the two versions of the program GWLREG, and then apply it in order to determine the polynomial function to represent the radial dependence of the electric dipole moment of a diatomic molecule.

2. MATHEMATICAL BASIS OF THE ALGORITHM

If we suppose that we have a set of n observations involving the independent variables x_j, $1 \leqslant j \leqslant k$, and the dependent variable y, then as long as $k < n$, the coefficients β_j of x_j are overdetermined. In these circumstances the statistical fitting of the data, subject to some experimental error ε assumed to be entirely associated with the dependent variable y and to be independent of the particular value of y, becomes the appropriate procedure according to the standard theory of regression analysis. If, furthermore, the values of y depend upon the parameters β_j in a linear manner, then the approach of linear regression may be used, much simpler and more direct than nonlinear parameter estimation. The values of the parameters b_j are the best, i.e. most precise and unbiased, estimates of the unknown quantities β_j if the criterion to determine them is the minimum value of the sum of the squares of the residuals

$$\sum_{i=1}^{n}\left(y - \sum_{j=0}^{k} b_j x_j\right)^2$$

over the n items in the data set; if b_0 is present ($x_0 \equiv 1$) and has a value significantly different from zero, then the function does not pass the origin of the coordinate system having k dimensions.

The algorithms, to be discussed in the following paragraphs, were programmed in BASIC for the following reasons: this computing language is widely used on 16-bit microcomputers of the type called personal computer; the program is easily written in this language; unlike FORTRAN, in compiled BASIC a program is prepared into an efficiently executing form in only one brief stage, without several passes for compilation and linking. Also, despite the fact that interpreted BASIC executes less rapidly than compiled BASIC, program preparation and testing is more conveniently conducted in the former way; moreover, no changes may be necessary

*All correspondence should be addressed to the author at the Academia Sinica address above; electronic mail to ogilvie@twniams.bitnet.

if one later resorts to compilation in order to improve speed of execution.

As a strategy for the construction of the new program GWLREG, it was decided to use as a basis the program MULNRG (Ogilvie & Abu-Elgheit, 1981) that was designed for only unweighted multiple linear regression. Thus, we first describe the mathematical basis of the algorithms of GWLREG containing the enhancements and extensions beyond MULNRG for the present objectives. We state the basic problem in matrix form, for each data set:

$$((y)) = ((\mathbf{x}))((\mathbf{b}))$$

in which ((x)) is a row matrix (or vector) of length $k+1$ representing the values of the independent variables, ((b)) is a column matrix also of length $k+1$ representing the values of the parameters to be determined according to the principle of least squares and $((y))$ is thus a matrix of order 1 (equivalent to a scalar). The addend 1 in $k+1$ results from the fact that, in the first instance, the function is not constrained to pass the origin; in that case the numbering of the elements of ((x)) and ((b)) starts at zero and $x_0 \equiv 1$. The data to be analyzed consist of n sets, the ith set containing the k values of the independent variables x_{ji}, $1 \leqslant j \leqslant k$, the value of the corresponding dependent variable y_i, and the weight w_i which may simply be the reciprocal of the variance (the square of the standard deviation) of the value of y_i determined as the result of replicated trials under the same conditions (i.e. the same values of x_{ji}). In order to determine the ((b)) matrix, we must form two other matrices, the symmetric square matrix $((a))$ of order $k+1$ and the column matrix $((g))$ of order $k+1$ having elements as follows:

$$\begin{aligned} a_{0,0} &= \sum w_i, \\ a_{jl} &= \sum x_{ji} x_{li} w_i, \quad 0 \leqslant j, l \leqslant k,\ j \leqslant l, \\ g_0 &= \sum y_i w_i, \\ g_j &= \sum x_{ji} y_i w_i, \quad 1 \leqslant j \leqslant k. \end{aligned}$$

In each case it is to be understood, here and elsewhere unless otherwise specified, that the summation Σ is over the values of i, $1 \leqslant i \leqslant n$, in the data set. The ((b)) matrix is then the product of the inverse, $((d))$ of the $((a))$ matrix with the $((g))$ matrix:

$$((d)) = ((a))^{-1}$$

and

$$((\mathbf{b})) = ((d))((g)).$$

We need also to calculate the sum of the squared errors (SSE), which can be formed from the $((d))$ and $((g))$ matrices and the transpose $((g))'$ of the $((g))$ matrix:

$$\mathrm{SSE} = ((g))'((g)) - ((d))((g)),$$

although computationally it is better to calculate SSE directly from the definitions (expressed in both the matrix form and the non-matrix form actually used in the program):

$$\begin{aligned} \mathrm{SSE} &= ((y))'((y)) - 2((\mathbf{b}))((\mathbf{x}))'((y)) \\ &\quad + ((\mathbf{b}))((\mathbf{x}))'((\mathbf{x}))((\mathbf{b})) \\ &= \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k} b_j x_j \right)^2. \end{aligned}$$

We define another column matrix $((s))$ of order $k+1$ of which each element s_j is the standard error, the square root of the variance, of the corresponding elements of the parameter matrix ((b)):

$$s_j = (d_{jj}\mathrm{SSE}/f)^{1/2},$$

in which f is the number of degrees of freedom of the regression process, equal to the number n of data sets minus the number of parameters $k+1$. The other essential indicators of the goodness of fit are:

the standard deviation σ of the fit,

$$\sigma = \left[\mathrm{SSE}\, n \Big/ \left(f \sum w_i \right) \right]^{1/2};$$

the absolute value $|r|$ of the sample correlation coefficient r,

$$|r| = \left\{ 1 - \mathrm{SSE} \Big/ \left[\left(\sum y_i^2 w_i \right) - \left(\sum y_i w_i \right)^2 \Big/ \left(\sum w_i \right) \right] \right\}^{1/2};$$

and the F statistic or F-value (Ogilvie & Abu-Elgheit, 1981),

$$F = (f/k) r^2 \left[\left(\sum y_i^2 w_i \right) - \left(\sum y_i w_i \right)^2 \Big/ \sum w_i \right] \Big/ \mathrm{SSE}.$$

The square dispersion matrix $((v))$, the product of the inverse of the $((a))$ matrix with the error variance σ^2 of regresion,

$$((v)) = \sigma^2((d)),$$

contains as its elements v_{jj} along the principal diagonal the variances of the **b** parameters,

$$v_{jj} = s_j^2,$$

whereas the off-diagonal elements are the covariances that contain information about the correlation of the nominally independent parameters, the regressor matrix ((b)) that is formally the primary objective of the regression analysis. The elements c_{jl} of the parameter correlation matrix, a more useful visual indicator of the independence of the b_j parameters, are formed simply by dividing the covariances by the square root of the product of the corresponding product of variances,

$$c_{jl} = v_{jl}/(v_{jj}v_{ll})^{1/2},$$

but in the prediction of new values of the dependent variable y_{pred} from the specified values of the independent variables ((x)) and the determined parameter matrix,

$$y_{pred} = ((x))((b))$$

the variance, or square of the standard error, of the predicted value is most directly calculated from the variances and the covariances

$$\mathrm{var}(y_{pred}) = \sum_{j=0}^{k} b_j^2 v_{jj} + 2 \sum_{j=0}^{k-1} \sum_{l=j+1}^{k} b_j b_l v_{jl}.$$

In the case that the function $y(\mathbf{x})$ is to be constrained to pass the origin, i.e. when all $x_{ki} = 0$, $k > 0$, then $y_i = 0$, some equations above must be modified as follows, essentially to remove the zeroth row and the zeroth column from the ((a)) matrix and the zeroth element correspondingly from the ((g)) column matrix:

$$a_{jl} = \sum x_{ji} x_{li} w_i, \quad 1 \leqslant j,\ l \leqslant k,$$

and

$$g_j = \sum x_{ji} y_i w_i, \quad 1 \leqslant j \leqslant k.$$

Then the product of the ((d)) and ((g)) matrices is the ((b)) matrix containing k elements starting with b_1, leading to the variance of predicted values y_{pred} of the independent variable y according to the equation

$$\mathrm{var}(y_{pred}) = \sum_{j=1}^{k} b_j^2 v_{jj} + 2 \sum_{j=1}^{k-1} \sum_{l=j+1}^{k} b_j b_l v_{jl}.$$

The above description has been based upon the supposition that the independent variables x_j, each unrelated to the other, are related to the dependent variable y in a simply linear manner. For analysis according to the model of a polynomial in one independent variable, there is a trivial conversion,

$$x_j = x_1^j, \quad j > 1,$$

that has no other effect on the computational method. However, in practice, the magnitudes of the parameter correlation coefficients are then found generally to approach unity; this deficiency of a power series is well known, but the practical utility of this form of model nevertheless makes it widely used. Other transformations of the independent variables can just as easily be effected, with no great effect on the nature of the computations. If the dependent variable y is however transformed from Y, then there must be an appropriate global weighting factor $W = (\mathrm{d}Y/\mathrm{d}y)^2$ applied (de Levie, 1986) further to any specific weight w_i arising from the standard deviation of any particular value Y_i.

The coding of the above equations into a computer program in two versions, each with adequate accuracy, suffices to achieve the objectives of the present work.

3. IMPLEMENTATIONS OF THE ALGORITHM AND TESTING

In the Cd version, the program GWLREG in BASIC consists of ten sections: initiation; variable transformation; control; data input; data correction; computation of results; output of basic results; output of table of residuals; output of parameter correlation coefficients; and termination. The initiation section declares the variables to be either double precision in floating point or integers and prints a heading. The variable transformation section consists of three function statements that convert all the independent and dependent variables and the weights into the appropriate form according to the particular application. The function to form the weight can incorporate any global weighting factor that may be required. In the control section, the user is prompted to ascertain whether either multiple linear or polynomial regression is required, the number of independent variables if the former or the maximum degree of the polynomial and whether results for lesser degrees are required if the latter, and finally whether the constraint of passage through the origin is to be applied. In the data input section, the number n of data sets is demanded, followed by a prompt for each value of x_{ji}, y_i and the particular weighting factor w_i (further to the global weighting factor automatically applied according to the appropriate function statement). These values are displayed compactly on the monitor screen so that visual inspection of their correctness is easily achieved. If no input data are to be corrected, according to the prompt at the end of the data input, then the computation begins, and the output of the basic results automatically proceeds. Next the user is prompted for output of first the table of residuals then the table of parameter correlation coefficients c_{jl}. The user is then prompted to delete any data cases, and to add further data cases; if there are any such deletions or additions then the computation of results is repeated. If there are no such alterations to the input data, either the program terminates for the case of the multiple linear regression and the polynomial regression at the maximum degree, or the program continues with the next higher order of polynomial regression according to the control parameters input at the beginning of the run.

The testing of GWLREG has been conducted in several ways. First of all, because the entire accuracy of the analysis depends vitally on the inversion of the ((a)) matrix, this subroutine was separately tested. Three different methods of matrix inversion were tried; one employed the Gauss–Jordan elimination method and a second, using an alternative elimination approach, was the MINV subroutine translated from FORTRAN. As Crout's method of decomposition with partial pivoting (Golub & van Loan, 1983), followed by iterative improvement, gave superior results, although requiring a somewhat greater storage space for arrays (Press *et al.*, 1985),

this procedure was incorporated into the Cd version of GWLREG. The test was the Hilbert matrix in which the value of each element is the reciprocal of the sum of the subscripts, i.e.

$$a_{jl} = 1/(j + l);$$

this matrix is notoriously ill-conditioned and is therefore an exemplary, critical and pertinent test of the inversion process.

At this point it becomes necessary to specify the varieties of BASIC used for the calculations. Three varieties have been tested, BASICA, Turbo-BASIC and Professional BASIC. The latter is most convenient when one types in the program because it prompts as soon as most common typing errors are made, but because it, like BASICA, is interpreted BASIC it executes relatively slowly. Turbo-BASIC is compiled and consequently executes rapidly. Because these regression calculations require only a few seconds, the execution time, in any case commonly less than the time to print the results, residuals and correlation coefficients, is much less a matter of concern than the numerical accuracy. There is in principle a significant difference between BASICA which carries 17 decimal digits and the other two which carry only 16 decimal digits of precision. For instance, in the inversion of the Hilbert matrix of order 6 by the Gauss–Jordan method, BASICA yields 10.1 correct digits on average, whereas the other processors yield 9.3 correct digits on average; for the Hilbert matrix of order 8, BASICA yields 7.2 correct digits, compared with 6.3 digits for the other specified processors. By Crout's method, the average number of correct digits is 11.3, 10.2 or 11.0 for the matrix of order 6 and 9.0, 7.7 or 8.0 for order 8 by BASICA, Professional BASIC or Turbo-BASIC, respectively. However, it should be noted that the greater precision in BASICA is accompanied by the small range of exponent ($\sim 10^{\pm 38}$) and the evaluation of numerical functions in only single precision even if the argument is double precision, whereas both Professional BASIC and Turbo-BASIC provide a much greater range of exponent ($\sim 10^{\pm 308}$) and the evaluation of numerical functions of double-precision arguments (and in Turbo-BASIC even single-precision arguments) in double precision. For the range of values in the matrices under test, the difference in precision between these processors is significant, and obviously the requirement of matrix inversion may prove the factor ultimately limiting the accuracy in permitting any program incorporating such a subroutine to produce significant results for high orders of the matrix to be inverted. Furthermore, although the gain in precision in the matrix inversion from Crout's decomposition over the Gauss–Jordan method is substantial, in the subsequent solution of the normal equations the iterative approach provides even further improvement. An alternative algorithm involving the direct factorization of the matrix of the coefficients of the normal equation according to Doolittle's approach produced essentially identical results to those from Crout's decomposition.

The second version (svd) of the program GWLREG resembles the first (Cd) except that, instead of an algorithm intended to solve the normal equations, the method of singular-value decomposition used (Golub & van Loan, 1983). Thus, the substitution subroutine and the main section of the Cd version in which the matrix of the coefficients of the normal equations is effectively inverted are replaced by several loops which serve to solve directly the design matrix (the set of linear equations, one for each data set, in which the regression coefficients appear as the unknown parameters) by decomposition of its singular values.

The second kind of test of the accuracy and efficiency of both versions of GWLREG involved the test problems used by Wampler (1969) in this evaluation of the then available computers and programs for linear least squares. These are the two polynomials of order 5, identified as $Y1$ and $Y2$:

$$Y1: \qquad y = \sum_{j=0}^{5} x^j$$

and

$$Y2: \qquad y = \sum_{j=0}^{5} (x/10)^j,$$

with values of x_i being the integers between 0 and 20; for the test of the cases constrained to pass the origin, the constant term (1) was omitted from both $Y1$ and $Y2$. For these four cases, i.e. $Y1(1)$, $Y2(1)$, $Y1(0)$ and $Y2(0)$ with 1 and 0 values of the y-intercept, respectively, the results are given in Table 1 for the cases of the three processors and the two algorithms. The indicators of the goodness of fit that are shown in the table are the F-value, for which a greater value is desirable, the standard deviation of the fit, for which a smaller value is desirable, and the average number of correct digits in the derived parameters b_j, $0 \leqslant j \leqslant 5$ or $1 \leqslant j \leqslant 5$, for the two values, 0 or 1 respectively, of the y-intercept. The results appear to demonstrate that for the implementation of the Crout decomposition BASICA is the best of the three processors, based on the average number of correct digits being greatest in most test cases, although it should be noted that for $Y1(1)$ and $Y2(0)$ the standard errors of the regression parameters were in some cases significantly smaller for Turbo-BASIC than for BASICA. In contrast, the results (not shown) for BASICA for the svd algorithm are relatively poor. Due to the latter qualifications, to the generally useful more extensive range of exponents in Turbo-BASIC and to the greater speed of execution of Turbo-BASIC, this processor may be preferred for use with either version of GWLREG.

Comparison of the results for $Y1(1)$ and $Y2(1)$ in Table 1 should also be made with the results in Wampler's (1969) report. In that case the average

Table 1. Comparison of results for the test regression problems by means of the algorithms using Crout decomposition (Cd) or singular-value decomposition (svd) with the interpreters BASICA (BA) and Professional BASIC (PB) and the compiler Turbo-BASIC (TB); the criteria are the F-value (F), the standard deviation of the fit (σ) and the average number of correct digits (No.)

	Cd			svd	
	BA	PB	TB	PB	TB
$Y1(1)$					
F	1.1×10^{28}	6.0×10^{26}	2.0×10^{29}	1.7×10^{31}	7.3×10^{31}
σ	1.8×10^{-8}	7.9×10^{-8}	4.3×10^{-9}	4.7×10^{-10}	2.3×10^{-10}
No.	8.0	7.5	9.3	11.3	11.3
$Y2(1)$					
F	3.2×10^{27}	4.3×10^{26}	7.8×10^{26}	2.5×10^{25}	8.0×10^{24}
σ	6.4×10^{-13}	9.4×10^{-8}	1.3×10^{-12}	7.2×10^{-12}	1.2×10^{-11}
No.	10.0	8.8	9.6	9.7	9.7
$Y1(0)$					
F	1.8×10^{27}	1.9×10^{26}	2.0×10^{26}	2.1×10^{31}	4.1×10^{31}
σ	4.5×10^{-8}	2.7×10^{-12}	1.3×10^{-7}	4.3×10^{-10}	3.0×10^{-10}
No.	8.4	9.6	7.4	10.2	11.2
$Y2(0)$					
F	6.9×10^{28}	3.7×10^{26}	1.6×10^{26}	2.6×10^{25}	8.0×10^{24}
σ	1.4×10^{-13}	1.9×10^{-12}	2.9×10^{-12}	7.2×10^{-12}	1.3×10^{-11}
No.	10.4	9.4	9.4	9.5	9.7

numbers of correct digits for double-precision runs with 16 digits were 6.9 and 7.9 for $Y1$ and 6.2 and 10.0 for $Y2$, although for double-precision runs with 18 digits the average numbers of correct digits were at least 9.3 for $Y1$ and 11.8 for $Y2$ when elimination methods were used and generally more for other methods. Thus, because the present results producing up to 11.3 correct digits for $Y1(1)$ and 10.0 for $Y2(1)$, respectively, are at least of the same order of magnitude as for the programs that Wampler (1969) tested, one may conclude that GWLREG is thus likely to be acceptable as a general routine for applications in linear regression, within the limitations mentioned above.

4. APPLICATION TO THE ELECTRIC DIPOLE-MOMENT FUNCTION OF HCl

In this application of the program GWLREG, we use not only the features discussed above but also the provision for weighting the input data. The latter provision is crucial in this case because both the values of the dependent variables, the matrix elements $\langle vJ|M(x)|v'J'\rangle$ of the dipole-moment function between vibration–rotational states specified by the quantum numbers v for vibration and J for rotation, and their nominal standard deviations vary over comparatively large ranges. The independent variables in this problem are the matrix elements $\langle vJ|x^j|v'J'\rangle$ of the displacement x to various powers j, which are roughly proportional to 10^{-j}. Therefore, the test problem $Y2(0)$ with $k = 8$ and using the Crout decomposition method is the most suitable to simulate the actual problem; running this problem with $0 \leqslant x \leqslant 32$ produced an F-value 5.7×10^{21} and a standard deviation of the fit 2.6×10^{-8} with at least 5 accurate digits for each regressor. As the conditions of this test are much more severe than in the actual problem, this performance is considered acceptable.

The electric dipole-moment function $M(x)$ of HCl may be expressed (Ogilvie & Tipping, 1983) in the form of a polynomial in the reduced internuclear displacement variable x in terms of the instantaneous R and equilibrium R_e internuclear separations. The power series which is truncated as required to fit the finite data, in this case at the seventh order,

$$M(x) = \sum_{j=0}^{7} M_j x^j, \quad x \equiv (R - R_e)/R_e,$$

has been determined anew from the available experimental data, namely the expectation values $\langle vJ|M(x)|vJ\rangle$ of the dipole moment in particular vibration–rotational states, measured by means of experiments on the electric resonance spectra at radio frequencies on molecular beams using the Stark effect, and the matrix elements $\langle v|M(x)|v'\rangle$ related to the intensities of vibration–rotational bands in the IR and VIS spectral regions. The data of Kaiser (1970) for the expectation values of the dipole moments of $H^{35}Cl$ and $D^{35}Cl$ were corrected for the deviation from the standard for such measurements [ground state of OCS (de Leeuw & Dymanus, 1970); corrected value for 1986 values (Cohen & Taylor, 1987) of the physical constants is $(2.385558 \pm 0.00010) \times 10^{-30}$ C m] and for the change in the fundamental physical constants since 1970; these corrections are not much greater than the experimental uncertainties (taken as 3.5 SE) quoted by Kaiser. The analogous datum of de Leeuw & Dymanus (1973) for $H^{35}Cl$ was also corrected, but only for the change in the physical constants. The data for the intensities of the vibration–rotational bands were used without corrections, and in most cases with the acceptance of their nominal standard deviations. Measurements of these intensities have been carried out for several decades, during which period both spectral resolving power and the precision of measurement have increased greatly. For the fundamental and first-overtone bands, many data are available, listed by Pugh & Rao

(1976) and by Smith *et al.* (1985). Among these data for either band, the values range widely, far beyond the nominal uncertainties of the measurements. In particular, the data of Benedict *et al.* (1957), part of an extensive collection, have been found to be too small by comparison with more recent measurements; for this reason, all the data reported by these authors were arbitrarily assigned a further weighting that caused these values to have practically no effect on the final results. Other data listed in the specified compilations have been omitted because the values are obviously inconsistent with more recent and presumably more accurate values. Specifically, the value by Smith (1973) for $\langle 0|M(x)|2\rangle$, a weighted mean of data that he then considered useful, has been omitted here because it is significantly smaller than real experimental values obtained both before and since that time. Similarly, the value of Benedict *et al.* for (1957) for $\langle 0|M(x)|3\rangle$ is confirmed to be too small (Stanton & Silver, 1988). For Kaiser's (1970) data of DCl, it proved impossible to fit simultaneously these data and all the data of HCl within reasonable bounds of uncertainties in relation to the nominal experimental errors; for this reason, these data of DCl were also arbitrarily assigned the same standard further weighting that removed the sensitivity of the results to these values. Kaiser attributed the inconsistency of these values of the expectation values of HCl and DCl to a possible breakdown of the Born–Oppenheimer approximation, but an alternative explanation based on the inadequacy of the treatment of the rotational dependence of the expectation values has also been proposed (Ogilvie, 1988).

Thus, weighted by the reciprocal of the square of the nominal standard deviations and then multiplied by a further factor (100) in the special cases specified above, the data, listed in Table 2, were thus fitted according to the program GWLREG in the mode for multiple linear weighted regression. The method involved the solution (Ogilvie & Tipping, 1983) of the overdetermined system of 33 simultaneous linear equations,

$$\langle vJ|M(x)|v'J'\rangle = \sum_{j=0}^{7} M_j \langle vJ|x^j|v'J'\rangle,$$

to find the best values of the eight coefficients M_j, $0 \leqslant j \leqslant 7$, from the selected 33 experimental data, namely the expectation values $\langle v|M(x)|v\rangle$ and matrix elements $\langle v|M(x)|v'\rangle$, $v \neq v'$, and the calculated values of the matrix elements $\langle v|x^j|v'\rangle$ of x^j according to the accurate analytic expressions

Table 2. List of the experimental expectation values and the matrix elements used in the present work

Quantity $\langle J\|M(x)\|v'J'\rangle$	Value/10^{-30} C m		Reference for observed
	Observed	Calculated	
$H^{35}Cl$			
$\langle 0,1\|M(x)\|0,1\rangle$	3.697315 ± 0.00047	3.697419	Kaiser (1970)
$\langle 0,2\|M(x)\|0,2\rangle$	3.697349 ± 0.00053	3.697419	Kaiser (1970)
$\langle 1,1\|M(x)\|1,1\rangle$	3.798748 ± 0.00053	3.798766	Kaiser (1970)
$\langle 2,1\|M(x)\|2,1\rangle$	3.897046 ± 0.00073	3.897164	Kaiser (1970)
$\langle 0,1\|M(x)\|0,1\rangle$	3.698434 ± 0.0010	3.697419	de Leeuw & Dymanus (1973)
$\langle 0\|M(x)\|1\rangle$	0.243372 ± 0.000053	0.243427	Pine *et al.* (1985)
$\langle 0\|M(x)\|1\rangle$	0.2268 ± 0.0043	0.243427	Toth *et al.* (1970)
$\langle 0\|M(x)\|2\rangle$	−0.02669 ± 0.00053	−0.026876	Toth *et al.* (1970)
$\langle 0\|M(x)\|1\rangle$	0.2235 ± 0.011*	0.243427	Benedict *et al.* (1957)
$\langle 0\|M(x)\|2\rangle$	−0.02345 ± 0.0012*	−0.026876	Benedict *et al.* (1957)
$\langle 0\|M(x)\|3\rangle$	0.00172 ± 0.000087*	0.001876	Benedict *et al.* (1957)
$\langle 1\|M(x)\|2\rangle$	0.3039 ± 0.0153*	0.33991	Benedict *et al.* (1957)
$\langle 2\|M(x)\|3\rangle$	0.3959 ± 0.0197*	0.40865	Benedict *et al.* (1957)
$\langle 0\|M(x)\|1\rangle$	0.2442 ± 0.0063	0.243427	Atwood *et al.* (1972)
$\langle 0\|M(x)\|2\rangle$	−0.02515 ± 0.00077	−0.026876	Atwood *et al.* (1972)
$\langle 0\|M(x)\|3\rangle$	0.002182 ± 0.000087*	0.001876	Atwood *et al.* (1972)
$\langle 0\|M(x)\|3\rangle$	0.001875 ± 0.00024	0.001876	Ogilvie & Lee (1989)
$\langle 0\|M(x)\|2\rangle$	−0.02712 ± 0.00030	−0.026876	Boulet *et al.* (1975)
$\langle 0\|M(x)\|1\rangle$	0.2480 ± 0.0223	0.243427	Penner & Weber (1953)
$\langle 0\|M(x)\|2\rangle$	−0.0270 ± 0.00108	−0.026876	Penner & Weber (1953)
$\langle 0\|M(x)\|2\rangle$	−0.02652 ± 0.00267	−0.026876	Jaffe *et al.* (1962)
$\langle 0\|M(x)\|4\rangle/10^{-4}$	−1.0217 ± 0.00333	−1.0217	Gelfand *et al.* (1981)
$\langle 0\|M(x)\|5\rangle/10^{-5}$	−2.8086 ± 0.0120	−2.8083	Gelfand *et al.* (1981)
$\langle 0\|M(x)\|6\rangle/10^{-5}$	2.2048 ± 0.0080	2.2045	Gelfand *et al.* (1981)
$\langle 0\|M(x)\|7\rangle/10^{-5}$	−1.0907 ± 0.00367	−1.0907	Gelfand *et al.* (1981)
$\langle 0\|M(x)\|5\rangle/10^{-5}$	−2.65 ± 0.265	−2.8083	Reddy (1980)
$\langle 0\|M(x)\|6\rangle/10^{-5}$	2.002 ± 0.200	2.2045	Reddy (1980)
$H^{37}Cl$			
$\langle 0\|M(x)\|1\rangle$	0.243665 ± 0.00011	0.24343	Pine *et al.* (1985)
$D^{35}Cl$			
$\langle 0,1\|M(x)\|0,1\rangle$	3.679470 ± 0.000267*	3.682987	Kaiser (1970)
$\langle 0,2\|M(x)\|0,2\rangle$	3.679670 ± 0.00117*	3.682987	Kaiser (1970)
$\langle 1,1\|M(x)\|1,1\rangle$	3.752952 ± 0.00083*	3.756166	Kaiser (1970)
$\langle 0\|M(x)\|1\rangle$	0.188 ± 0.00093*	0.2067	Benedict *et al.* (1957)
$\langle 0\|M(x)\|2\rangle$	−0.0166 ± 0.00083*	−0.01879	Benedict *et al.* (1957)
$\langle 0\|M(x)\|3\rangle$	0.00103 ± 0.000050*	0.001225	Benedict *et al.* (1957)

* Statistical weighting factor modified as described in the text.

(Bouanich *et al.*, 1986) including terms containing (consistently) the potential-energy coefficients a_j (Coxon & Ogilvie, 1982) in the Dunham (1932) function,

$$V(x) = a_0 x^2 \left(1 + \sum_{j=1}^{8} a_j x^j\right)$$

up to a_8. The distinction between $\langle vJ|M(x)|v'J'\rangle$ and $\langle v||M(x)|v'\rangle$ or between $\langle vJ|x^j|v'J'\rangle$ and $\langle v|x^j|v'\rangle$ may be ignored because the higher-order rotational effects, quite small for small values of J, have been ignored in the analysis of the experimental data to produce the experimental values $\langle vJ|M(x)|v'J'\rangle$. The results of the fit, M_j and their standard errors, are listed in Table 3, whereas the calculated values of the expectation values and matrix elements are compared with the experimental values in Table 2. The F-value of the fit is 5.1×10^7 and the standard deviation is 1.4×10^{-7} (almost an order of magnitude greater than the apparent limit of the numerical precision); the sample correlation coefficient is 0.99999997, whereas in the parameter correlation matrix only 1 value of the 28 elements c_{jl} below the principal diagonal has a magnitude >0.9, and only 4 further have magnitudes >0.8. Moreover, the same results, well within the experimental errors, were given by all three BASIC processors. All these indicators are consistent with a successful fit of an adequate model. From the values of the dipole-moment coefficients M_j, together with both the calculated values of the expectation value $\langle 0|M(x)|0\rangle$ and the matrix elements $\langle 0|M(x)|v'\rangle$, $0 < v' < 8$, and the potential-energy coefficients a_j and including the standard errors of all these quantities, the values of the coefficients $C_0^{v'}$ and $D_0^{v'}$ in the Herman–Wallis factor $F(J, J', v, v')$ were calculated from the analytic relations (Tipping & Ogilvie, 1982; Ogilvie & Tipping, 1985); the standard errors of these values of the Herman–Wallis coefficients were also calculated by means of a Monte-Carlo procedure (Ogilvie, 1984). The calculated values are compared with the experimentall results in Table 4.

In general, the values of the dipole-moment coefficeints M_j calculated here are similar to those calculated by Kobayashi & Suzuki (1986). In particular, although the signs of the experimental matrix elements $\langle v|M(x)|v'\rangle$ in the present work agree with those determined by Kobayashi & Suzuki (1986) in the course of the generation of their septic function,

Table 3. Derived values of the coefficients M_j of the electric dipole-moment function of HCl

j	$M_j/10^{-30}$ C m
0	3.645867 ± 0.000253
1	4.123336 ± 0.00147
2	0.006882 ± 0.0177
3	−5.10960 ± 0.0438
4	−3.0648 ± 0.0876
5	−1.174 ± 0.142
6	−1.457 ± 0.242
7	1.172 ± 0.687

the value of M_7 has the opposite sign but a similar magnitude. One reason for the differences may be in inclusion by the latter authors of superseded, too small values of the matrix elements $\langle 0|M(x)|2\rangle$ and $\langle 0|M(x)|3\rangle$, but in any case their value of M_7 is much less statistically significant than the value here. Despite the fact that they also tried to incorporate the experimental values of the Herman–Wallis coefficients into their fit, there remain large discrepancies between their calculated and observed values of these coefficients (for which they failed to provide any indication of the uncertainty); for this reason also, some values of the matrix elements, for instance $\langle 0|M(x)|5\rangle$ and $\langle 0|M(x)|7\rangle$, fit poorly. [Commonly the experimental matrix elements from recent experiments have much smaller relative errors than the small corrections in the Herman–Wallis factor (that are especially prone to model errors) and thus the former constitute a much more reliable basis of production of dipole-moment functions.] The deviations of the present calculated values of the Herman–Wallis coefficients from the experimental values are generally as good as those in their work, although only the signs, not the magnitudes, of the experimental values entered the present calculations directly, in order to resolve the ambiguity of sign of the matrix elements (because the IR intensities are proportional to the squares of such quantities). Because the values of the potential-energy coefficients a_j generated by Kobayashi & Suzuki (1986) differ greatly from those accurately determined by means of the analysis of all the spectra of all the isotopic variants of HCl (Coxon & Ogilvie, 1982) and found to reproduce the spectral frequencies and wavenumbers within 1.37 SD, the basis of their determination of the dipole-moment function is in any case suspect. Their claim also to take account of the data of Clayton *et al.* (1983) is irrelevant because

Table 4. Comparison of observed and calculated values of the coefficients $C_0^{v'}$ and $D_0^{v'}$ in the Herman–Wallis factor

	$C_0^{v'}/10^{-2}$		$D_0^{v'}/10^{-4}$		
v'	Observed	Calculated	Observed	Calculated	Reference for observed
0	—	0	—	1.119 ± 0.0004	—
1	−2.560 ± 0.008	−2.60 ± 0.00058	3.20 ± 0.23	2.63 ± 0.008	Pine *et al.* (1985)
2	−0.86 ± 0.15	−0.498 ± 0.0056	4.1 ± 2.0	3.04 ± 0.043	Toth *et al.* (1970)
3	1.7 ± 0.3	1.27 ± 0.060	—	4.22 ± 0.23	Benedict *et al.* (1957)
4	2.77 ± 0.11	2.87 ± 0.11	14.2 ± 1.7	8.3 ± 0.69	Gelfand *et al.* (1981)
5	1.74 ± 0.13	1.67 ± 0.19	4.61 ± 2.4	8.1 ± 1.2	Gelfand *et al.* (1981)
6	3.35 ± 0.11	2.62 ± 0.12	7.99 ± 1.8	10.5 ± 0.73	Gelfand *et al.* (1981)
7	4.37 ± 0.13	3.24 ± 0.13	—	13.6 ± 0.86	Gelfand *et al.* (1981)

the latter spectral data were also included in the analysis made by Coxon & Ogilvie (1982). It should be noted that the analysis of Clayton *et al.* (1983) is also suspect; because, for instance, the range of values of a_6 determined by Clayton *et al.* (1983) for $H^{35}Cl$ and $H^{37}Cl$ separately is 10 times the stated standard error, it is difficult to have confidence in the mathematical significance of at least these error estimates. In a recent determination of the dipole-moment function of HCl, because Bouanich (1987) used terms up to only a_6 in the potential-energy function and in other quantities, his results are not as accurate as those presented here; moreover, his use of the superseded values of $\langle 0|M(x)|2\rangle$ and $\langle 0|M(x)|3\rangle$ means that, even within the limited accuracy of the terms he retains, the results are obsolescent.

5. CONCLUSIONS

We have presented the mathematical basis for two versions of a new program for the general analysis of linear regression incorporating weights for the input data that may be based on both the standard deviations of these data and a global weighting factor that is required if the dependent variable has been transformed in a nontrivial manner in order to permit the linear regression. The implementation of this basis in both versions, Cd and svd, of the program GWLREG, listed in the appendices, has been tested both in its critical subroutine for matrix inversion and by means of standard functions (Wampler, 1969). For data sets with strongly correlated regressor variables, the svd version is preferable. The program GWLREG has then been applied to the new determination of the electric dipole-moment function of the diatomic molecule HCl from the experimental data for the expectation values from the Stark effect and the matrix elements from IR intensities. The results will be used in a separate determination of the vibration–rotational Einstein coefficeints of HCl (to be published).

It should be noted that the new program GWLREG, based on previously developed algorithms and programs (Ogilvie & Abu-Elgheit, 1981), has many advantages because of both its flexibility (within the confines of linear regression models) and the range of indicators of goodness of fit. Even though one version may, on occasion, prove limited by the numerical presision involved in the procedure of matrix inversion, such cases will require extreme precision. Used in this version, the Crout method has been demonstrated to be very efficient in this application (Press *et al.*, 1985). Although producing even greater precision, the other algorithm based on singular-value decomposition requires about 60% greater storage area for arrays, has 30% more statements, and thus executes correspondingly less rapidly than the version based on the Crout decomposition. Moreover, the application of the constraint of passage of the fitted function through the origin is more difficult to accomplish in the svd version than in the Cd version, although in fact better results (i.e. much smaller residuals and correspondingly a smaller standard deviation of the fit) are obtained in the test programs *Y*1 and *Y*2 if this constraint is not enforced. The comparative tests of the two versions to the solution of the normal equations or of the equivalent design matrix indicate that under these conditions the svd algorithm does not produce results greatly superior to the other; certainly in the actual application to the dipole-moment data the experimental error of the latter is much more important to the overall accuracy than the numerical limitations of either algorithm. Also, the implementation of the svd algorithm operates poorly with the BASICA interpreter, presumably because of the limitations of the exponent range thereof. For these reasons we present here both versions, Cd and svd, of the program GWLREG.

For comparison, another general routine for unconstrained parameter estimation (in FORTRAN and therefore much less easy to use on personal computers) in use in this laboratory can provide at least 14 significant digits of accuracy in the same regression problems *Y*1 and *Y*2; this program utilizes the algorithm developed by Levenberg *et al.* (Osborne, 1972) and avoids the explicit inversion of a matrix by applying Gauss–Newton procedure starting from supplied initial estimates and (hopefully) converging to the correct results. However, in this indirect approach, unlike the direct approach in GWLREG, the final results are dependent in principle on the choice of the initial estimates; such a disadvantage may be tolerable when for nonlinear regression models no direct approach is possible, but one would in general prefer to avoid any possible arbitrariness of the solutions to a physical problem. The limitations due to numerical precision in GWLREG can essentially be ignored if and when a further variety of BASIC for personal microcomputers becomes available that carries at least 20—and preferably 30—significant digits; the time is now ripe for such a development. Further extensions to GWLREG that one might contemplate are the provision of an automatic plotting routine (Ogilvie, 1986), which would however make the program less portable between different BASIC processors, and a subroutine for prediction (interpolation) of further values of the dependent variable for a given set of values of the independent variables, with full measure of the standard deviation of such predictions through the use of the parameter correlation matrix. The latter extension is trivial to program, but would require a specific inverse transformation of the dependent variable relative to that used in the regression analysis. Other transformations of input variable can be easily accommodated by means of the insertion of the necessary statements in the input section. The flexibility of BASIC as the language of this program GWLREG makes such modifications easy

to carry out, with rapid subsequent recompilation if necessary.

Acknowledgements—I thank Dr L. Farnell for helpful advice and the National Science Council of the Republic of China for the support of a visiting research professorship.

REFERENCES

Atwood M. R., Vu H. & Vodar B. (1972) *J. Phys. (Paris)* **33,** 495.
Benedict W. S., Herman R., Moore G. E. & Silverman S. (1957) *J. Chem. Phys.* **26,** 1671.
Bouanich J. P. (1987) *J. Quant. Spectrosc. Radiat. Transfer* **38,** 89.
Bouanich J. P., Ogilvie J. F. & Tipping R. H. (1986) *Comput. Phys. Commun.* **39,** 439.
Boulet C., Piollet-Mariel E. & Levy A. (1975) *Spectrochim. Acta* **31A,** 463.
Clayton C. M., Merdes D. W., Pliva J., McCubbin T. K. & Tipping R. H. (1983) *J. Mol. Spectrosc.* **98,** 168.
Cohen E. R. & Taylor B. N. (1987) *Rev. Mod. Phys.* **59,** 1121.
Coxon J. A. & Ogilvie J. F. (1982) *J. Chem. Soc. Faraday Trans. II* **78,** 1345.
Dunham J. L. (1932) *Phys. Rev.* **41,** 721.
Gelfand J., Zughul M., Rabitz H. & Han C. J. (1981) *J. Quant. Spectrosc. Radiat. Transfer* **26,** 303.
Golub G. H. & van Loan C. F. (1983) *Matrix Computations.* Johns Hopkins University Press, Baltimore, Md.
Jaffe, J. H., Kimel S. & Hirshfeld M. A. (1962) *Can. J. Phys.* **40,** 113.
Kaiser E. W. (1970) *J. Chem. Phys.* **53,** 1686.
Kobayashi M. & Suzuki I. (1986) *J. Mol. Spectrosc.* **116,** 422.
de Leeuw F. H. & Dymanus A. (1970) *Chem. Phys. Lett.* **7,** 288.
de Leeuw F. H. & Dymanus A. (1973) *J. Mol. Spectrosc.* **48,** 427.
de Levie R. (1986) *J. Chem. Educ.* **63,** 10.
Ogilvie J. F. (1984) *Comput. Chem.* **8,** 205.
Ogilvie J. F. (1986) *Coll. Microcomput.* **4,** 283.
Ogilvie J. F. (1988) *J. Phys.* **B21,** 1663.
Ogilvie J. F. & Abu-Elgheit M. A. (1981) *Comput. Chem.* **5,** 33; and references therein.
Ogilvie J. F. & Lee Y. P. (1989) *Chem. Phys. Lett.* **159,** 239.
Ogilvie J. F. & Tipping R. H. (1983) *Int. Rev. Phys. Chem.* **3,** 3.
Ogilvie J. F. & Tipping R. H. (1985) *J. Quant. Spectrosc. Radiat. Transfer* **33,** 145.
Osborne M. (1972) In *Conference on Numerical Methods for Nonlinear Optimisation* (Edited by Lootsma F. A.). Academic Press, New York.
Penner S. S. & Weber D. (1953) *J. Chem. Phys.* **21,** 649.
Pine A. S., Fried A. & Elkins J. W. (1985) *J. Mol. Spectrosc.* **109,** 30.
Press W. H., Flannery B. P., Teukolsky S. A. & Vetterling W. T. (1985) *Numerical Recipes: the Art of Scientific Computing.* Cambridge Univ. Press, London.
Pugh L. A. & Rao K. N. (1976) In *Molecular Spectroscopy: Modern Research,* Vol. 2 (Edited by Rao K. N.), p. 165. Academic Press, New York.
Reddy K. V. (1980) *J. Mol. Spectrosc.* **82,** 127.
Smith F. G. (1973) *J. Quant. Spectrosc. Radiat. Transfer* **13,** 717.
Smith M. A. H., Rinsland C. P., Fridovich B. & Rao K. N. (1985) In *Molecular Spectroscopy: Modern Research,* Vol. 3 (Edited by Rao K. N.), p. 112. Academic Press, New York.
Stanton A. C. & Silver J. A. (1988) *Appl. Opt.* **27,** 5009.
Tipping R. H. & Ogilvie J. F. (1982) *J. Mol. Spectrosc.* **96,** 442.
Toth R. A., Hunt R. H. & Plyler E. K. (1970) *J. Mol. Spectrosc.* **35,** 110.
Wampler R. H. (1969) *J. Res. Nat. Bur. Stand.* **73B,** 59.

APPENDIX A

Listing of GWLREG Based on Crout Decomposition

```
10 DEFDBL A-H,O-Z : DEFINT I-N
20 LPRINT "General Weighted Linear Regression (Cd) @ J.F. Ogilvie 1990 1 1"
30 LPRINT : PRINT "General Weighted Linear Regression " : PRINT
40 ' VARIABLE TRANSFORMATION * * * * *
50 DEF FNX(X,K,I)=X ' CDBL(I-1) ' for test case
60 DEF FNY(Y,I)=Y ' 1.#+1.D-0*CDBL(I-1)+1.D-0*CDBL(I-1)^2+1.D-0*CDBL(I-1)^3
   +1.D-0*CDBL(I-1)^4+1.D-0*CDBL(I-1)^5 ' for test cases
70 DEF FNW(X,Y,W)=1#/W^2
80 ' CONTROL SECTION
90 DIM X(11,100),Y(100),W(100),A(11,11),G(11),B(11),S(11),AA(11,11),IX(11)
100 DIM AI(11,11)
110 INPUT "For polynomial regression type 'POLY' or for linear regression
    type 'LIN'";A$
120 IF A$="POLY" THEN 130 ELSE IF A$="LIN" THEN 160 ELSE PRINT
    "Error in input" : GOTO 110
130 INPUT "Maximum degree of polynomial (<11) = ";K4
140 INPUT "Are results for lesser degrees wanted?  Type 'YES' or 'NO'.";B$
150 GOTO 180
160 INPUT "Number of independent variables (<11) = ";K2
170 B$= "NO"
180 INPUT "Is y-intercept to be zero?  Type 'YES' or 'NO'. ";D$
190 IF D$="YES" THEN IY=0 ELSE IY=1
200 ' DATA INPUT SECTION * * * * *
210 INPUT "Number of data cases (<101) = ";N
220 K5=1
230 IF A$="POLY" THEN K2=1
240 FOR I=K5 TO N
250   FOR K=1 TO K2
260     PRINT "  x(";K;",";I;")="; : INPUT ; X(K,I)
270     X(K,I)=FNX(X(K,I),K,I)
280   NEXT K
290   IF A$<>"POLY" THEN 310
300   FOR K=2 TO K4 : X(K,I)=X(1,I)^K : NEXT K
```

```
310   PRINT "    y(";I;")="; : INPUT ; Y(I)
320   Y(I)=FNY(Y(I),I)
330   PRINT "   w(";I;")="; : INPUT W(I)
340   IF ABS(W(I))<.00000000000001# THEN W(I)=1#
350   W(I)=FNW(X(1,I),Y(I),W(I))
360 NEXT I
370 INPUT "Are any data to be corrected before processing?
    Type 'YES' or 'NO'.";D$
380 IF D$="YES" THEN GOSUB 1290
390 ' COMPUTATION OF RESULTS--PREPARATION OF NORMAL EQUATIONS * * * * *
400 IF A$<>"POLY" THEN 440
410 IF B$="YES" THEN K7=1 ELSE K7=K4
420 K6=K7
430 K2=K6
440 K1=K2+IY
450 FOR I=1 TO K1
460   G(I)=0# : S(I)=0# : B(I)=0#
470   FOR J=1 TO K1 : A(I,J)=0# : NEXT J
480 NEXT I
490 SW=0# : FOR I=1 TO N : SW=SW+W(I) : NEXT I
500 IF IY=0 THEN G1=0# ELSE A(1,1)=SW
510 Y5=0#
520 FOR K=1 TO K1
530   FOR J=1 TO K1
540     FOR I=1 TO N
550       IF K>1 THEN 630
560       IF IY=1 THEN IF J=1 THEN 600
570       G(J)=G(J)+X(J-IY,I)*Y(I)*W(I)
580       IF IY=0 THEN 610 ELSE A(1,J)=A(1,J)+X(J-1,I)*W(I)
590       GOTO 640
600       G(1)=G(1)+Y(I)*W(I)
610       IF IY=1 THEN Y5=Y5+Y(I)^2*W(I) ELSE IF J=1 THEN Y5=Y5+Y(I)^2*W(I)
620       IF IY=1 THEN 640 ELSE IF J=1 THEN G1=G1+Y(I)*W(I)
630       IF J>=K THEN A(K,J)=A(K,J)+X(K-IY,I)*X(J-IY,I)*W(I)
640     NEXT I
650     A(J,K)=A(K,J)
660   NEXT J
670 NEXT K : IF IY=1 THEN G1=G(1)
680 ' SOLUTION OF NORMAL EQUATIONS BY THE CROUT METHOD * * * * *
690 FOR I=1 TO K1 : FOR J=1 TO K1 : AA(I,J)=A(I,J) : NEXT J
700   NEXT I
710 FOR I=1 TO K1 : R5=0#
720   FOR J=1 TO K1 : IF ABS(AA(I,J))>R5 THEN R5=ABS(AA(I,J))
730   NEXT J : IF ABS(R5)<1D-38 THEN 1680
740   S(I)=1#/R5 : NEXT I
750 FOR J=1 TO K1 : IF J=1 THEN 800
760   FOR I=1 TO J-1 : SS=AA(I,J) : IF I=1 THEN 790
770     FOR K=1 TO I-1 : SS=SS-AA(I,K)*AA(K,J)
780       NEXT K : AA(I,J)=SS
790   NEXT I
800   R5=0# : FOR I=J TO K1 : SS=AA(I,J) : IF J=1 THEN 830
810     FOR K=1 TO J-1 : SS=SS-AA(I,K)*AA(K,J) : NEXT K
820     AA(I,J)=SS
830     R6=S(I)*ABS(SS)
840     IF R6<R5 THEN 860
850     K3=I : R5=R6
860     NEXT I : IF J=K3 THEN 900
870   FOR K=1 TO K1 : R6=AA(K3,K) : AA(K3,K)=AA(J,K)
880     AA(J,K)=R6 : NEXT K
890   S(K3)=S(J)
900   IX(J)=K3 : IF J=K1 THEN 930
910   IF ABS(AA(J,J))<1D-38 THEN AA(J,J)=1D-32
920   R6=1#/AA(J,J) : FOR I=J+1 TO K1 : AA(I,J)=AA(I,J)*R6 : NEXT I
930 NEXT J : IF ABS(AA(K1,K1))<1D-38 THEN AA(K1,K1)=1D-32
940 FOR I=1 TO K1 : FOR J=1 TO K1 : AI(I,J)=0# : NEXT J
950   AI(I,I)=1# : NEXT I
960 FOR K=1 TO K1 : FOR I=1 TO K1 : S(I)=AI(I,K) : NEXT I
970   GOSUB 1580 : FOR I=1 TO K1 : AI(I,K)=S(I) : NEXT I : NEXT K
980 FOR I=1 TO K1 : S(I)=G(I) : NEXT I : GOSUB 1580
990 FOR I=1 TO K1 : B(I)=S(I) : NEXT I
1000 FOR I=1 TO K1 : S(I)=-G(I)
1010   FOR J=1 TO K1 : S(I)=S(I)+A(I,J)*B(J) : NEXT J
1020   NEXT I : GOSUB 1580
1030 FOR I=1 TO K1 : B(I)=B(I)-S(I) : NEXT I
1040 SS=0# : FOR I=1 TO N
1050   IF IY=0 THEN YY=0# ELSE YY=B(1)
1060   FOR K=1 TO K2 : YY=YY+B(K+IY)*X(K,I) : NEXT K
1070   SS=SS+W(I)*(YY-Y(I))^2
1080 NEXT I
1090 FOR K=1 TO K1 : S(K)=SQR(AI(K,K)*SS/CDBL(N-K1)) : NEXT K
1100 R5=SS/(Y5-G1^2/SW)
1110 R6=1#-R5 : F=CDBL(N-K1)*R6/(R5*CDBL(K2)) : R6=SQR(R6)
1120 ' OUTPUT OF RESULTS * * * * *
```

```
1130 LPRINT : IF A$="POLY" THEN LPRINT "Degree of polynomial = ";K6 : LPRINT
1140 LPRINT "No.                Coefficient                Standard Error"
1150 FOR K=1 TO K1 : LPRINT K-IY,B(K),S(K) : NEXT K
1160 LPRINT : LPRINT "F-value = "; : LPRINT USING "##.####^^^^";F;
1170 LPRINT "       Standard deviation of fit = ";
1180 LPRINT USING "##.####^^^^"; SQR(SS*CDBL(N)/(CDBL(N-K1)*SW))
1190 LPRINT "Absolute value of sample Correlation Coefficient = "; R6
1200 INPUT "Is table of residuals wanted?  Type 'YES ' or 'NO'.";C$
1210 IF C$="YES" THEN GOSUB 1450
1220 INPUT "Is table of correlation coefficients wanted?
     Type 'YES' or 'NO'.";C$
1230 IF C$="YES" THEN GOSUB 1530
1240 GOSUB 1290
1250 IF A$<>"POLY" THEN 1700
1260 K6=K6+1
1270 IF K6<=K4 THEN 430 ELSE 1700
1280 ' DATA CORRECTION SECTION * * * * *
1290 PRINT : INPUT "Number of data cases to be deleted = ";K7
1300 IF K7<1 THEN 1400 ELSE IF K7=1 THEN 1320
1310 PRINT "Enter case numbers in descending order."
1320 FOR J=1 TO K7
1330   N=N-1
1340   INPUT "Case number to be deleted = ";K3
1350   FOR I=K3 TO N
1360     FOR K=1 TO K2 : X(K,I)=X(K,I+1) : NEXT K
1370     Y(I)=Y(I+1) : W(I)=W(I+1)
1380   NEXT I
1390 NEXT J
1400 INPUT "Number of data cases to be added = ";K3
1410 K5=N+1 : N=N+K3
1420 IF K7>0 OR K3>0 THEN 230
1430 RETURN
1440 ' OUTPUT OF TABLE OF RESIDUALS * * * * *
1450 LPRINT : LPRINT " Table of Residuals " : LPRINT
1460 LPRINT "Case No.    X              Y calc          Y obs          Ycalc-Yobs
1470 FOR I=1 TO N
1480   IF IY=0 THEN YY=0# ELSE YY=B(1)
1490   FOR K=1 TO K2 : YY=YY+B(K+IY)*X(K,I) : NEXT K
1500   LPRINT "   ";I;"   ";X(1,I);"    ";YY;"     ";Y(I);"    ";YY-Y(I)
1510 NEXT I
1520 LPRINT : RETURN
1530 ' OUTPUT OF COVARIANCE (CORRELATION) COEFFICIENTS * * * * *
1540 LPRINT "   Parameter Correlation Matrix " : LPRINT
1550 FOR I=1 TO K1 : FOR J=1 TO I
1560     LPRINT USING "###.####";AI(I,J)/SQR(AI(I,I)*AI(J,J));
1570   NEXT J : LPRINT : NEXT I : RETURN
1580 ' SUBSTITUTION SUBROUTINE * * * * *
1590 K5=0 : FOR I=1 TO K1 : K3=IX(I) : SS=S(K3)
1600   S(K3)=S(I) : IF K5=0 THEN 1620
1610   FOR J=K5 TO I-1 : SS=SS-AA(I,J)*S(J) : NEXT J : GOTO 1630
1620   IF ABS(SS)>1D-36 THEN K5=I
1630   S(I)=SS : NEXT I
1640 FOR I=K1 TO 1 STEP -1 : SS=S(I)
1650   IF I=K1 THEN 1670
1660   FOR J=I+1 TO K1 : SS=SS-AA(I,J)*S(J) : NEXT J
1670   S(I)=SS/AA(I,I) : NEXT I : RETURN
1680 PRINT "Singular matrix -- inversion aborted"
1690 ' TERMINATION * * * * *
1700 LPRINT : PRINT : PRINT "Analysis completed"
1710 D$=INKEY$ : IF D$="" THEN 1710
1720 END
```

APPENDIX B

Listing of GWLREG Based on Singular-value Decomposition

```
  10 DEFDBL A-H,O-Z : DEFINT I-N
  20 LPRINT "General Weighted Linear Regression (svd) @ J.F. Ogilvie 1990 1 1"
  30 LPRINT : PRINT "General Weighted Linear Regression " : PRINT
  40 ' VARIABLE TRANSFORMATION * * * * *
  50 DEF FNX(X,K,I)=X ' CDBL(I-1) ' for test case
  60 DEF FNY(Y,I)=Y ' 1.#+1.D-0*CDBL(I-1)+1.D-0*CDBL(I-1)^2+1.D-0*CDBL(I-1)^3
     +1.D-0*CDBL(I-1)^4+1.D-0*CDBL(I-1)^5 ' for test cases
  70 DEF FNW(X,Y,W)=1#/W
  80 ' CONTROL SECTION
  90 DIM X(11,100),Y(100),W(100),U(100,11),G(100),B(11),CV(11,11),T(11)
 100 DIM V(11,11),Q(11)
 110 INPUT "For polynomial regression type 'POLY' or for linear regression
     type 'LIN'";A$
```

```
120 IF A$="POLY" THEN 130 ELSE IF A$="LIN" THEN 160 ELSE PRINT "Error in
    input" : GOTO 110
130 INPUT "Maximum degree of polynomial (<11) = ";K4
140 INPUT "Are results for lesser degrees wanted?  Type 'YES' or 'NO'.";B$
150 GOTO 180
160 INPUT "Number of independent variables (<11) = ";K2
170 B$= "NO"
180 ' DATA INPUT SECTION * * * * *
190 INPUT "Number of data cases (<101) = ";N
200 K5=1
210 IF A$="POLY" THEN K2=1
220 FOR I=K5 TO N
230   FOR K=1 TO K2
240     PRINT "  x(";K;",";I;")="; : INPUT ; X(K,I)
250     X(K,I)=FNX(X(K,I),K,I)
260   NEXT K
270   IF A$<>"POLY" THEN 290
280   FOR K=2 TO K4 : X(K,I)=X(1,I)^K : NEXT K
290   PRINT "    y(";I;")="; : INPUT ; Y(I)
300   Y(I)=FNY(Y(I),I)
310   PRINT "   w(";I;")="; : INPUT W(I)
320   IF ABS(W(I))<.00000000000001# THEN W(I)=1#
330   W(I)=FNW(X(1,I),Y(I),W(I))
340 NEXT I
350 INPUT "Are any data to be corrected before processing?  Type 'YES' or
    'NO'.";D$
360 IF D$="YES" THEN GOSUB 1540
370 ' COMPUTATION OF RESULTS--PREPARATION OF DESIGN MATRIX * * * * *
380 IF A$<>"POLY" THEN 420
390 IF B$="YES" THEN K7=1 ELSE K7=K4
400 K6=K7
410 K2=K6
420 K1=K2+1
430 FOR I=1 TO N
440   FOR J=1 TO K2 : U(I,J+1)=X(J,I)*W(I) : NEXT J
450   U(I,1)=W(I) : G(I)=Y(I)*W(I) : NEXT I
460 ' COMPUTATION OF RESULTS--SOLUTION BY SINGULAR-VALUE DECOMPOSITION * *
470 D=0# : P5=0# : P4=0#
480 FOR I=1 TO K1 : L=I+1 : Q(I)=P5*D : D=0# : S=0# : P5=0#
490   IF I>N THEN 590
500   FOR K=I TO N : P5=P5+ABS(U(K,I)) : NEXT K
510   IF ABS(P5)<0# THEN 590
520   FOR K=I TO N : U(K,I)=U(K,I)/P5 : S=S+U(K,I)*U(K,I) : NEXT K
530   F=U(I,I) : D=-SQR(S)*SGN(F) : H=F*D-S : U(I,I)=F-D
540   IF I=K1 THEN 580
550   FOR J=L TO K1 : S=0#
560     FOR K=I TO N : S=S+U(K,I)*U(K,J) : NEXT K : F=S/H
570     FOR K=I TO N : U(K,J)=U(K,J)+F*U(K,I) : NEXT K : NEXT J
580   FOR K=I TO N : U(K,I)=P5*U(K,I) : NEXT K
590   T(I)=P5*D : D=0# : S=0# : P5=0#
600   IF I>N THEN 720
610   IF I=K1 THEN 720
620     FOR K=L TO K1 : P5=P5+ABS(U(I,K)) : NEXT K
630     IF ABS(P5)<0# THEN 720
640       FOR K=L TO K1 : U(I,K)=U(I,K)/P5 : S=S+U(I,K)*U(I,K) : NEXT K
650       F=U(I,L) : D=-SQR(S)*SGN(F) : H=F*D-S : U(I,L)=F-D
660       FOR K=L TO K1 : Q(K)=U(I,K)/H : NEXT K
670       IF I=N THEN 710
680         FOR J=L TO N : S=0#
690           FOR K=L TO K1 : S=S+U(J,K)*U(I,K) : NEXT K
700           FOR K=L TO K1 : U(J,K)=U(J,K)+S*Q(K) : NEXT K : NEXT J
710         FOR K=L TO K1 : U(I,K)=P5*U(I,K) : NEXT K
720   S=ABS(T(I))+ABS(Q(I)) : IF P4<S THEN P4=S
730 NEXT I
740 FOR I=K1 TO 1 STEP -1 : IF I>=K1 THEN 810
750   IF ABS(D)<0# THEN 800
760     FOR J=L TO K1 : V(J,I)=(U(I,J)/U(I,L))/D : NEXT J
770     FOR J=L TO K1 : S=0#
780       FOR K=L TO K1 : S=S+U(I,K)*V(K,J) : NEXT K
790       FOR K=L TO K1 : V(K,J)=V(K,J)+S*V(K,I) : NEXT K : NEXT J
800     FOR J=L TO K1 : V(I,J)=0# : V(J,I)=0# : NEXT J
810   V(I,I)=1# : D=Q(I) : L=I : NEXT I
820 FOR I=K1 TO 1 STEP -1 : L=I+1 : D=T(I)
830   IF I<K1 THEN FOR J=L TO K1 : U(I,J)=0# : NEXT J
840   IF ABS(D)<0# THEN 910
850   D=1#/D : IF I=K1 THEN 900
860   FOR J=L TO K1 : S=0#
870     FOR K=L TO N : S=S+U(K,I)*U(K,J) : NEXT K
880       F=(S/U(I,I))*D
890       FOR K=I TO N : U(K,J)=U(K,J)+F*U(K,I) : NEXT K : NEXT J
900   FOR J=I TO N : U(J,I)=U(J,I)*D : NEXT J : GOTO 920
910   FOR J=I TO N : U(J,I)=0# : NEXT J
920   U(I,I)=U(I,I)+1# : NEXT I
```

```
930 FOR K=K1 TO 1 STEP -1 : FOR K3=1 TO 50 : FOR L=K TO 1 STEP -1
940       K5=L-1 : IF ((ABS(Q(L))+P4)-P4)<0# THEN 1030
950       IF ((ABS(T(K5))+P4)-P4)<0# THEN 970
960         NEXT L
970       C=0# : S=1# : FOR I=L TO K : F=S*Q(I)
980         IF ((ABS(F)+P4)-P4)<0# THEN 1020
990         D=T(I) : H=SQR(F*F+D*D) : T(I)=H : H=1#/H
1000          C=D*H : S=-(F*H) : FOR J=1 TO N : P2=U(J,K5) : P3=U(J,I)
1010            U(J,K5)=P2*C+P3*S : U(J,I)=-P2*S+P3*C : NEXT J
1020        NEXT I
1030      P3=T(K) : IF L<>K THEN 1060
1040      IF P3>=0# THEN 1210
1050      T(K)=-P3 : FOR J=1 TO K1 : V(J,K)=-V(J,K) : NEXT J : GOTO 1210
1060      IF K3=50 THEN PRINT "No convergence in 50 iterations" : STOP
1070      P1=T(L) : K5=K-1 : P2=T(K5) : D=Q(K5) : H=Q(K)
1080      F=((P2-P3)*(P2+P3)+(D-H)*(D+H))/(2#*H*P2) : D=SQR(F*F+1#)
1090      F=((P1-P3)*(P1+P3)+H*((P2/(F+ABS(D)*SGN(F)))-H))/P1
1100      C=1# : S=1# : FOR J=L TO K5 : I=J+1 : D=Q(I) : P2=T(I)
1110        H=S*D : D=C*D : P3=SQR(F*F+H*H) : Q(J)=P3 : C=F/P3
1120        S=H/P3 : F=P1*C+D*S : D=-P1*S+D*C : H=P2*S
1130        P2=P2*C : FOR K5=1 TO K1 : P1=V(K5,J) : P3=V(K5,I)
1140          V(K5,J)=P1*C+P3*S : V(K5,I)=-P1*S+P3*C : NEXT K5
1150        P3=SQR(F*F+H*H) : T(J)=P3 : IF ABS(P3)<0# THEN 1170
1160        P3=1#/P3 : C=F*P3 : S=H*P3
1170        F=C*D+S*P2 : P1=-S*D+C*P2 : FOR K5=1 TO N
1180          P2=U(K5,J) : P3=U(K5,I) : U(K5,J)=P2*C+P3*S
1190          U(K5,I)=-P2*S+P3*C : NEXT K5 : NEXT J
1200        Q(L)=0# : Q(K)=F : T(K)=P1 : NEXT K3
1210      NEXT K :  P5=0# : FOR J=1 TO K1 : IF T(J)>P5 THEN P5=T(J)
1220      NEXT J : FOR J=1 TO K1 : IF T(J)<.0000000000000001#*P5 THEN T(J)=0#
1230      NEXT J : FOR J=1 TO K1 : S=0# : IF ABS(T(J))<0# THEN 1250
1240      FOR I=1 TO N : S=S+U(I,J)*C(I) : NEXT I : S=S/T(J)
1250      Q(J)=S : NEXT J : FOR J=1 TO K1 : S=0#
1260        FOR K=1 TO K1 : S=S+V(J,K)*Q(K) : NEXT K : B(J)=S : NEXT J
1270 SS=0# : FOR I=1 TO N : S=B(1)
1280    FOR J=1 TO K2 : S=S+B(J+1)*X(J,I) : NEXT J
1290    SS=SS+((Y(I)-S)*W(I))^2 : NEXT I : G1=0#
1300 FOR I=1 TO K1 : Q(I)=0# : IF T(I)<>0# THEN Q(I)=1#/(T(I)*T(I))
1310    NEXT I : FOR I=1 TO K1 : FOR J=1 TO I : S=0#
1320      FOR K=1 TO K1 : S=S+V(I,K)*V(J,K)*Q(K) : NEXT K
1330      CV(I,J)=S : NEXT J : NEXT I : Y5=0# : SW=0#
1340 FOR I=1 TO N : SW=SW+W(I)^2 : Y5=Y5+(Y(I)*W(I))^2 : G1=G1+Y(I)*W(I)
1350    NEXT I :  R5=SS/(Y5-G1^2/SW)
1360 R6=1#-R5 : F=CDBL(N-K1)*R6/(R5*CDBL(K2)) : R6=SQR(R6)
1370 ' OUTPUT OF RESULTS * * * * *
1380 LPRINT : IF A$="POLY" THEN LPRINT "Degree of polynomial = ";K6 : LPRINT
1390 LPRINT "No.              Coefficient                    Standard Error"
1400 FOR K=1 TO K1 : LPRINT K-1,B(K),SQR(CV(K,K)*SS/CDBL(N-K1)) : NEXT K
1410 LPRINT : LPRINT "F-value = "; : LPRINT USING "##.####^^^^";F;
1420 LPRINT "        Standard deviation of fit = ";
1430 LPRINT USING "##.####^^^^"; SQR(SS*CDBL(N)/(CDBL(N-K1)*SW))
1440 LPRINT "Absolute value of sample Correlation Coefficient = "; R6
1450 INPUT "Is table of residuals wanted?  Type 'YES ' or 'NO'.";C$
1460 IF C$="YES" THEN GOSUB 1700
1470 INPUT "Is table of correlation coefficients wanted?  Type 'YES' or
     'NO'.";C$
1480 IF C$="YES" THEN GOSUB 1760
1490 GOSUB 1540
1500 IF A$<>"POLY" THEN 1830
1510 K6=K6+1
1520 IF K6<=K4 THEN 410 ELSE 1830
1530 ' DATA CORRECTION SECTION * * * * *
1540 PRINT : INPUT "Number of data cases to be deleted = ";K7
1550 IF K7<1 THEN 1650 ELSE IF K7=1 THEN 1570
1560 PRINT "Enter case numbers in descending order."
1570 FOR J=1 TO K7
1580    N=N-1
1590    INPUT "Case number to be deleted = ";K3
1600    FOR I=K3 TO N
1610      FOR K=1 TO K2 : X(K,I)=X(K,I+1) : NEXT K
1620      Y(I)=Y(I+1) : W(I)=W(I+1)
1630    NEXT I
1640 NEXT J
1650 INPUT "Number of data cases to be added = ";K3
1660 K5=N+1 : N=N+K3
1670 IF K7>0 OR K3>0 THEN 210
1680 RETURN
1690 ' OUTPUT OF TABLE OF RESIDUALS * * * * *
1700 LPRINT : LPRINT " Table of Residuals " : LPRINT
1710 LPRINT "Case No.    X              Y calc          Y obs          Ycalc-Yobs
1720 FOR I=1 TO N : S=B(1)
1730    FOR J=1 TO K2 : S=S+B(J+1)*X(J,I) : NEXT J
1740    LPRINT "  ";I;"   ";X(1,I);"    ";S;"     ";Y(I);"    ";S-Y(I)
```

```
1750 NEXT I : LPRINT : RETURN
1760 ' OUTPUT OF COVARIANCE (CORRELATION) COEFFICIENTS * * * * *
1770 LPRINT "   Parameter Correlation Matrix " : LPRINT
1780 FOR I=1 TO K1 : FOR J=1 TO I
1790     LPRINT USING "###.####";CV(I,J)/SQR(CV(I,I)*CV(J,J));
1800   NEXT J : LPRINT : NEXT I : RETURN
1810 PRINT "Singular matrix -- inversion aborted"
1820 ' TERMINATION * * * * *
1830 LPRINT : PRINT : PRINT "Analysis completed"
1840 D$=INKEY$ : IF D$="" THEN 1840
1850 END
```