ANALYTICAL BIOCHEMISTRY 110, 9–18 (1981)

A Nonlinear Regression Program for Small Computers

R. G. DUGGLEBY

Department of Biochemistry, University of Queensland, St. Lucia, Queensland, 4067, Australia

Received April 21, 1980

A BASIC computer program for performing weighted nonlinear regression is described and a listing of the program is given. The program, which is small and simple to use, has been designed to be run by users with little knowledge of mathematics or computers. Robust methods of analysis are described which may be applied to data in which experimental errors are not normally distributed, and the program incorporates one such method. It is shown that the program is useful for the analysis of data conforming to the Michaelis–Menten equation, a single exponential, and to binding equations, and other applications are discussed.

The quantitative analysis of experimental data frequently requires comparison with some sort of mathematical equation or model. All too often this analysis involves a transformation of the data which are then plotted and a straight line is drawn through them. The slope and intercept may then be transformed or combined in various ways to obtain the parameters of the original equation. For example Eq. [1] describes a

$$\hat{y} = y_0 e^{-kt} \qquad [1]$$

first-order decay curve,[1] and the classical method of analysis is to plot

ln (y) against t. From the slope (S) and the intercept (I) the parameters are obtained using $k = -S$ and $y_0 = e^I$. A much better method for the analysis of first-order decay curves is to fit Eq. [1] directly to the data by nonlinear regression. Although this type of analysis is relatively straightforward, nonlinear regression has made very little impact in biochemistry with the notable exception of enzyme-kinetic studies.

A great many packaged computer programs are available for performing nonlinear regression analysis but these are, without exception, long and sophisticated programs designed to be run on large computers. In this paper, a simple BASIC nonlinear regression program is presented which can be run on mini- or even microcomputers. Some of the underlying theory is presented but a understanding of this theory is not a prerequisite for using the program. The program has been deliberately limited to the situation in which the equation to be fitted has two parameters, as such equations occur quite commonly.

THEORY

Transforming experimental data into a form which may be plotted as a straight line is a useful method of displaying the data but it is not a reliable method for its analysis. Experimental errors can be grossly magnified as is the case with the Lineweaver–Burk plot of enzyme–kinetic data and fitting of a straight line to the transformed data will not, in general, yield the "best" values for the parameters. This is true, regardless of whether the fit is performed "by eye" or by linear least-squares analysis. Careful weighting of the transformed data may compensate for the distortion in certain in-

[1] Throughout the paper a distinction will be drawn between y, the observed value of the dependent variable; $\hat{y}$, a value calculated for a particular set of parameter values; and $\bar{y}$ the true, but usually unknown, value.

0003-2697/81/010009-10$02.00/0

stances but in others, such as a Scatchard plot in which the observed variable appears on both axes, distortion is unavoidable. If transformation of the data is to be eliminated it is necessary to fit the mathematical equation to the data directly. The form of the equation is not usually a matter of choice, but rather it depends on some underlying theoretical model. Generally, these theoretical considerations lead to an equation which is nonlinear in the parameters and the fitting procedure will involve nonlinear regression.

On the whole, biochemists regard nonlinear regression with a mixture of awe and suspicion, as something which is beyond their capacity to comprehend. In fact it is quite simple, requiring little more than a knowledge of elementary algebra and in this section the basic principles are set out. Later, a simple and flexible computer program, which embodies these principles, will be described.

Classical Methods

Nonlinear regression. The fundamental ideas underlying the Gauss–Newton method of nonlinear regression have been described by Wilkinson (1). These ideas are best understood against a background of the principles of linear regression which will be described briefly. Consider the case where we wish to fit Eq. [2] to a set of N observations, where a_1 and a_2 are parameters to

$$\hat{y} = a_1x_1 + a_2x_2 \qquad [2]$$

be estimated and x_1 and x_2 are independent variables. (This is not intended to imply that x_1 and x_2 are necessarily independent of one another; for example, x_2 may equal x_1^2). To estimate a_1 and a_2 we first form the sums

$$s_1 = \sum wx_1^2;\ s_2 = \sum wx_1x_2;\ s_3 = \sum wx_2^2;$$
$$s_4 = \sum wx_1y;\ s_5 = \sum wx_2y,$$

where w is a nonnegative "weight" attached to each observation, and which is discussed in detail below. The parameters may be calculated using

$$a_1 = (s_3s_4 - s_2s_5)/\Delta$$
$$a_2 = (s_1s_5 - s_2s_4)/\Delta$$

where

$$\Delta = s_1s_3 - s_2^2.$$

To calculate the standard errors of a_1 and a_2, we calculate the sum of squares of residuals (s_6) and the residual standard error (r_s) using

$$s_6 = \sum w(y - \hat{y})^2$$
$$r_s = [s_6/(N - 2)]^{1/2}$$

The standard errors of a_1 and a_2 are given by

$$\mathrm{SE}(a_1) = r_s(s_3/\Delta)^{1/2}$$
$$\mathrm{SE}(a_2) = r_s(s_1/\Delta)^{1/2}$$

The values of a_1 and a_2 calculated above are "best-fit" values in the sense that they minimize the weighted sum of squares, s_6. This is necessarily so because the formulae for a_1 and a_2 are found by differentiating s_6 with respect to a_1 and a_2, setting these derivatives equal to zero and solving the resultant simultaneous equations.

For nonlinear equations, a similar procedure does not lead to a simple solution and we cannot calculate the best-fit values for the parameters in a single step. What can be done in a single step is to take some estimates of these values and correct them to give better estimates.

Suppose we are trying to fit a nonlinear equation in which there is a single parameter, b:

$$\hat{y} = f(b)$$

An estimate, $\bar{b}$, will differ from the best-fit value, $\tilde{b}$, by an unknown amount q:

$$\tilde{b} = \bar{b} + q.$$

From the Taylor series we may write

$$\hat{y} = f(\tilde{b} + q)$$
$$= f(\bar{b}) + qf'(\bar{b}) + \frac{q^2}{2!}f''(\bar{b}) + \frac{q^3}{3!}f'''(\bar{b}) + \cdots,$$

where f', f'', an
differentiation w
all the terms in
$f(\bar{b})$ as $\bar{y}$, and r
the experimenta

y

In other words
q can be found t
the difference be
the calculated
dependent varia
is treated as th
value for q so
correct due to t
high-order term
ever, the newl
be refined by a
cedure repeate
"iteration."

This concept
in which ther
parameter. Co
described by

nonlinear para
values of one o
the initial estim
and $b_2^{(0)}$, we
(q_1 and q_2) to
the equation

in which z is
and p_2 are the
$\delta\bar{y}/\delta b_2$. The c
mated as desc
and are used
nonlinear par

These new es
then be refin
q_1 and q_2 ar
the standard
to those of q

where f', f'', and so on denote successive differentiation with respect to b. If we ignore all the terms in q^2 and beyond, represent $f(\bar{b})$ as $\bar{y}$, and replace the unknown $\hat{y}$ with the experimental y, we can rearrange to get

$$y - \bar{y} \simeq qf'(\bar{b}).$$

In other words, an approximate value for q can be found by linear regression in which the difference between the experimental and the calculated value of y is treated as the dependent variable while the derivative $f'(\bar{b})$ is treated as the independent variable. The value for q so derived will not be exactly correct due to the approximation in ignoring high-order terms of the Taylor series. However, the newly calculated value of b may be refined by applying this correction procedure repeatedly, a process known as "iteration."

This concept may be generalized to cases in which there are more than a single parameter. Consider the arbitrary function described by Eq. [3], where b_1 and b_2 are

$$\hat{y} = f(b_1, b_2; X), \qquad [3]$$

nonlinear parameters and X represents the values of one or more independent variables. If the initial estimates of the parameters are $b_1^{(0)}$ and $b_2^{(0)}$, we may calculate corrections (q_1 and q_2) to these parameters by fitting the equation

$$z = q_1 p_1 + q_2 p_2$$

in which z is the residual $(y - \bar{y})$ while p_1 and p_2 are the partial derivatives, $\delta\bar{y}/\delta b_1$ and $\delta\bar{y}/\delta b_2$. The coefficients, q_1 and q_2, are estimated as described above for the linear case and are used to correct the values of the nonlinear parameters

$$b_1^{(1)} = b_1^{(0)} + q_1$$

$$b_2^{(1)} = b_2^{(0)} + q_2.$$

These new estimates of the parameters may then be refined in further iterations. When q_1 and q_2 are negligible ("convergence"), the standard errors of b_1 and b_2 are equal to those of q_1 and q_2, respectively (1), and are calculated as described earlier.

Partial derivatives. We saw above that in nonlinear regression the calculation of the corrections (q) requires values of the partial derivatives (p) which are treated as independent variables. Ideally, these derivatives should be obtained by analytical differentiation of the nonlinear function (Eq. [3]), which may involve some tedious algebra. In practice, the derivatives can be calculated to the required precision by numerical differentiation which avoids the algebra. The function (Eq. [3]) is evaluated after the parameter b_1 is perturbed by an amount d_1:

$$\bar{y}' = f(b_1 + d_1, b_2; X)$$

and a first-order approximation of p_1 is given by

$$p_1 \simeq (\bar{y}' - \bar{y})/d_1.$$

A more accurate value may be found using a second-order approximation if the function is evaluated at a second point:

$$\bar{y}'' = f(b_1 - d_1, b_2; X)$$

$$p_1 \simeq (\bar{y}' - \bar{y}'')/2d_1.$$

A value for p_2 is found by applying this same procedure to b_2. In the computer program to be described later, d_1 and d_2 are chosen to be 2% of b_1 and b_2, respectively.

Weighting. It may happen that we have advance knowledge that some observations are more accurate than others and this information should be incorporated in the analysis. This is achieved by weighting each observation by an amount (w) which is inversely proportional to its variance, so that fitting involves minimizing the weighted sum of squares, s_6. This same weight must also be applied in forming the sums s_1–s_5 which are used to calculate regression coefficients. Frequently, these *a priori* weights are calculated from some simple weighting function. For example, the standard deviation of y may be approximately proportional to y in which case $w = 1/y^2$ will be used as the weighting function. One of the

methods of robust regression described below is based on weighting.

Robust Methods

Robust regression. The classical method for fitting a function to experimental data involves minimizing the sum of squares of residuals. Since each residual (z) is squared in the summation and as the worst observations will have the largest residuals, the fit tends to be dominated by these observations. A drastic solution to this difficulty is to discard the worst observations but to do this it is necessary to introduce an essentially arbitrary division between acceptable and unacceptable observations. A gentler procedure is to use a "robust" method in which the residuals are modified so that less emphasis is placed on the larger ones.

Wahrendorf (2) has described a robust method which he has applied to the analysis of Scatchard plot data. Briefly, the residual sum of squares is replaced with the function $\sum\rho(z)$:

$$\rho(z) = \begin{cases} z^2 & \text{if } |z| < c \\ 2c|z| - c^2 & \text{if } |z| \geq c, \end{cases}$$

where c is a "robustness constant." The value of $\rho(z)$ increases as the square of z when z is numerically less than c, but thereafter increases as the absolute value of z. There is a smooth transition at $z = c$. If c is chosen to be very large, this method is indistinguishable from the normal least-squares method.

A somewhat different method has been described by Mosteller and Tukey (3) in which each squared residual is multiplied by a "bisquare weight," b_w:

$$b_w = \begin{cases} (1 - u^2)^2 & \text{if } |u| \leq 1 \\ 0 & \text{if } |u| > 1, \end{cases}$$

where $u = z/c$ and c is the robustness constant. If $z > c$, that particular observation is "weighted-out" of the analysis (*i.e.*, ignored), while moderate-sized residuals acquire a fractional weight. Observations which agree well with the fitted function have a small residual and are given close to a full weight of 1.0. As with *a priori* weights, the bisquare weight is applied in calculating the sums s_1–s_6 from which the regression coefficients are calculated.

If c is chosen to be very large, b_w will equal 1.0 for all observations and we have the usual least-squares method. Usually we will want to choose a value of c which is large enough that $|u| < 1$ for the great majority of observations. In the computer program to be described later, a value equal to six times the mean absolute residual ($c = 6\sum|z|/N$) has been utilized but Mosteller and Tukey have pointed out that many other values will also work well. Bisquare weighting can be used in conjunction with *a priori* weights in which case the final weight applied will be the product of b_w and w. In the calculation of b_w and c for this latter case, we must use the weighted residual $zw^{1/2}$ in place of z alone.

Median methods. If experimental data were free of error, values for the two parameters of Eq. [3] could be obtained by measuring y at two points and solving the resultant nonlinear simultaneous equations. In practice, of course, data do contain some variability and more than two measurements are made. The purpose of the additional measurements is to increase the reliability of the parameter estimates and, more importantly, to permit the calculation of a measure of this reliability. Cornish-Bowden and Eisenthal (4) have suggested a robust method of analysis for the case where Eq. [3] represents the Michaelis–Menten equation, and this may be adapted to any two-parameter, nonlinear equation. Values for the parameters are calculated from each possible pair of measurements and these $N(N-1)/2$ values for b_1 and b_2 are used to determine the best estimates of the values. It was originally proposed (4) that the median values of b_1 and b_2 should be taken as the best estimates but it was subsequently pointed out (5) that the median values for b_1 and b_2 may be biased. It is usually

possible to fi
which are m
ferred proced
of measurem
and these are
unbiased cor
latter values
of their magn
values are lo
mates of b_1 a
median value
transformatio
involved is us
ample will s
The median
of the Michael
whereas $c_1 =$
dian–unbiase
which are use
the median va
simple: $V =$

Confidence
found by an ex
(6). Kendall's
find the ranks
interval at any
the values of
ranks are dete
eters are four
to b_1 and b_2.

An alternat
described by
on a special
determination
of experiment
of each set
estimate of $\bar{y}$,
are calculated
simultaneous
estimates of t
has the advant
of transformin
binations of t
tages have be

All median
solution of a
The solutions
equations an
ficult (but by

possible to find combinations of b_1 and b_2 which are median–unbiased and the preferred procedure is as follows. For each pair of measurements, b_1 and b_2 are calculated and these are transformed to the median–unbiased combinations, c_1 and c_2. These latter values are separately ranked in order of their magnitude and the centrally ranked values are located. Finally, the best estimates of b_1 and b_2 are calculated from the median values of c_1 and c_2 by reversing the transformation. The type of transformation involved is usually quite simple and an example will serve to illustrate this point. The median values of $b_1 = V$ and $b_2 = K$ of the Michaelis–Menten equation are biased whereas $c_1 = 1/V$ and $c_2 = K/V$ are median–unbiased. The reverse transformations which are used to calculate V and K from the median values of c_1 and c_2 are equally simple: $V = 1/c_1$ and $K = c_2/c_1$.

Confidence intervals for b_1 and b_2 may be found by an extension of this median method (6). Kendall's S^* statistic is calculated to find the ranks which enclose the confidence interval at any desired probability level, and the values of c_1 and c_2 which occupy these ranks are determined. Limits on the parameters are found by transforming c_1 and c_2 to b_1 and b_2.

An alternative median method has been described by Duggleby (7,8) which is based on a special experimental design. Multiple determinations of y are made under two sets of experimental conditions and the median of each set of replicates is taken as an estimate of $\bar{y}$. Values for b_1 and b_2, which are calculated by solving the resulting two simultaneous equations, will be the best estimates of these parameters. This method has the advantage that it avoids the necessity of transforming into median–unbiased combinations of the parameters. Other advantages have been described previously (7,8).

All median methods require the algebraic solution of a set of nonlinear equations. The solutions will depend on the form of the equations and for this reason it is difficult (but by no means impossible) to incorporate a median method into a general computer program. Thus, in the program described below, robustness has been approached by the bisquare weighting method. For the sake of completeness, the solutions required for median methods are also given for the specific models considered below. These solutions may be useful for calculating initial estimates of the parameters.

RESULTS

A computer program embodying the nonlinear regression principles outlined under the Theory section has been written in BASIC. The program was developed using BASIC-11, a version of this language which is used in the PDP-11 series of computers. Exploitation of special features of this version of the language was deliberately avoided to facilitate transfer of the program to other computers which will support the BASIC language.

A listing of the program is shown in Fig. 1 and while it might appear that the program is quite long, this impression is largely illusory. Of the 176 lines in the program, 68 are REM statements which contribute nothing to the operation of the program but serve solely to document it. Of the remaining 108 lines, 26 print either blank lines or headings. Thus, the heart of the program is less than one-half of the total and compression of the source code may be achieved readily, an important consideration for microcomputers where storage limitations are critical.

The only statement which depends on the equation to be fitted is line 2650 which, in Fig. 1, describes the Michaelis–Menten equation. Other models may be fitted by replacing this line with the appropriate expression. For example, the first-order decay curve described by Eq. [1] might be written:

2650 **G** = $B(1)$*exp(−$B(2)$*X).

Partial derivatives are calculated by nu-

```
1000 REM
1010 REM   NONLINEAR REGRESSION PROGRAM FOR TWO PARAMETER EQUATIONS
1020 REM
1030 REM ..................................................................
1040 REM   THE FOLLOWING ARRAYS ARE USED:
1050 REM          X1          INDEPENDENT VARIABLE
1060 REM          Y           DEPENDENT VARIABLE
1070 REM          W           A PRIORI WEIGHTS
1080 REM          B           THE PARAMETERS
1090 REM          P           PARTIAL DERIVATIVES
1100 REM          Q           CORRECTION VECTOR, AND STD ERRORS
1110 REM
1120 REM   THE FOLLOWING VARIABLES ARE USED:
1130 REM          I,J         LOOP INDICES
1140 REM          N1          NUMBER OF DATA POINTS
1150 REM          S1-S6       SUM USED IN REGRESSION
1160 REM          X           CURRENT VALUE OF X1
1170 REM          G           THEORETICAL VALUE OF Y
1180 REM          Z           RESIDUAL
1190 REM          U           VALUE OF G USED TO CALCULATE P
1200 REM          D           DETERMINANT USED IN REGRESSION
1210 REM          C           CONVERGENCE TEST QUANTITY
1220 REM          V           RESIDUAL VARIANCE
1230 REM          S           INPUT STD DEV
1240 REM          I1          ITERATION COUNTER
1250 REM          R1-R5       USED FOR BISQUARE WEIGHTING
1260 REM
1270 REM   THE STRINGS Q$ AND B$ ARE USED TO DETERMINE WEIGHTING
1280 REM ..................................................................
1290 REM
1300 DIM X1(20),Y(20),W(20)
1310 PRINT "HOW MANY DATA POINTS"
1320 INPUT N1
1330 PRINT
1340 PRINT "WHAT TYPE OF WEIGHTING ..."
1350 PRINT "  CONSTANT STD DEV        (C)"
1360 PRINT "  PROPORTIONAL STD DEV    (P)"
1370 PRINT "  BETWEEN THE ABOVE       (B)"
1380 PRINT "  STD DEV SUPPLIED        (S)"
1390 INPUT Q$
1400 PRINT
1410 PRINT "BISQUARE WEIGHTING TOO"
1420 INPUT B$
1430 PRINT
1440 PRINT "INPUT INDEPENDENT VARIABLE, DEPENDENT VARIABLE";
1450 IF Q$<>"S" THEN 1470
1460 PRINT " AND STD DEV";
1470 PRINT
1480 REM ..................................................................
1490 REM   INPUT THE DATA
1500 REM ..................................................................
1510 FOR I=1 TO N1
1520 IF Q$="S" THEN 1600
1530 INPUT X1(I),Y(I)
1540 W(I)=1
1550 IF Q$="C" THEN 1620
1560 W(I)=1/Y(I)
1570 IF Q$="B" THEN 1620
1580 W(I)=W(I)^2
1590 GO TO 1620
1600 INPUT X1(I),Y(I),S
1610 W(I)=1/S^2
1620 NEXT I
1630 PRINT
1640 PRINT "ENTER ESTIMATES OF THE PARAMETERS"
1650 INPUT B(1),B(2)
1660 PRINT
1670 PRINT "  B(1)          B(2)          SSQ"
1680 REM ..................................................................
1690 REM   BEGIN NONLINEAR ITERATIONS
1700 REM ..................................................................
1710 I1=0
1720 I1=I1+1
1730 IF I1>10 THEN 2570
1740 S1=0
1750 S2=0
1760 S3=0
1770 S4=0
1780 S5=0
1790 S6=0
1800 R1=0
1810 FOR I=1 TO N1
1820 REM ..................................................................
1830 REM   CALCULATE THE THEORETICAL VALUE OF Y, THE RESIDUAL
1840 REM   AND R1,R5 FOR BISQUARE WEIGHTING
1850 REM ..................................................................
1860 X=X1(I)
1870 GOSUB 2590
1880 Z=Y(I)-G
1890 R5=SQR(W(I))*Z
1900 R1=R1+ABS(R5)
1910 REM ..................................................................
1920 REM   CALCULATE THE PARTIAL DERIVATIVES
1930 REM ..................................................................
1940 FOR J=1 TO 2
1950 B(J)=1.02*B(J)
1960 GOSUB 2590
1970 U=G
1980 B(J)=B(J)*.98/1.02
1990 GOSUB 2590
2000 B(J)=B(J)/.98
2010 P(J)=(U-G)/(.04*B(J))
2020 NEXT J
2030 REM ..................................................................
2040 REM   CHECK IF BISQUARE WEIGHTING IS REQUIRED
2050 REM ..................................................................
2060 R3=1
2070 IF B$="N" THEN 2100
2080 IF I1=1 THEN 2100
2090 GOSUB 2680
2100 REM ..................................................................
2110 REM   FORM THE VARIOUS SUMS
2120 REM ..................................................................
2130 S1=S1+R3*W(I)*P(1)^2
2140 S2=S2+R3*W(I)*P(1)*P(2)
2150 S3=S3+R3*W(I)*P(2)^2
2160 S4=S4+R3*W(I)*P(1)*Z
2170 S5=S5+R3*W(I)*P(2)*Z
2180 S6=S6+R3*W(I)*Z^2
2190 NEXT I
2200 REM ..................................................................
2210 REM   CORRECT THE PARAMETERS AND CHECK FOR CONVERGENCE
2220 REM ..................................................................
2230 R2=6*R1/N1
2240 D=S1*S3-S2^2
2250 Q(1)=(S3*S4-S2*S5)/D
2260 Q(2)=(S1*S5-S2*S4)/D
2270 C=ABS(Q(1)/B(1))+ABS(Q(2)/B(2))
2280 B(1)=B(1)+Q(1)
2290 B(2)=B(2)+Q(2)
2300 PRINT B(1),B(2),S6
2310 IF C>1.00000E-05 THEN 1720
2320 REM ..................................................................
2330 REM   RUN HAS CONVERGED.  CALCULATE FINAL RESULTS
2340 REM ..................................................................
2350 V=S6/(N1-2)
2360 Q(1)=SQR(V*S3/D)
2370 Q(2)=SQR(V*S1/D)
2380 PRINT
2390 PRINT "FINAL VALUES ..."
2400 PRINT
2410 PRINT "B(1) = ",B(1),"  +/- ",Q(1)
2420 PRINT "B(2) = ",B(2),"  +/- ",Q(2)
2430 PRINT
2440 PRINT "  X             Y             Y(HAT)          DIFF"
2450 PRINT
2460 FOR I=1 TO N1
2470 X=X1(I)
2480 GOSUB 2590
2490 PRINT X1(I),Y(I),G,Y(I)-G
2500 NEXT I
2510 PRINT
2520 PRINT "END OF PROGRAM"
2530 STOP
2540 REM ..................................................................
2550 REM   TOO MANY ITERATIONS
2560 REM ..................................................................
2570 PRINT "TERMINATED AFTER 10 ITERATIONS"
2580 GO TO 2350
2590 REM ..................................................................
2600 REM   INSERT FITTED FUNCTION HERE IN THE FORM:
2610 REM
2620 REM            G=F(B(1),B(2),X)
2630 REM
2640 REM ..................................................................
2650 G=B(1)*X/(B(2)+X)
2660 REM ..................................................................
2670 RETURN
2680 REM ..................................................................
2690 REM   CALCULATE THE BISQUARE WEIGHT
2700 REM ..................................................................
2710 R3=0
2720 R4=(R5/R2)^2
2730 IF R4>=1 THEN 2750
2740 R3=(1-R4)^2
2750 RETURN
```

FIG. 1. BASIC computer program for weighted nonlinear regression analysis. In general, the variable names used in the program correspond to those used in the Theory section. The major exceptions are quantities used in calculating bisquare weights (here named R1–R5) and $\hat{y}$ (here named G). The only library functions used are the square root function (SQR) and the absolute value function (ABS). The circumflex (^) which appears in several places (e.g., line 1580) indicated exponentiation.

merical differentiation (lines 1940–2020) and this method has been found to be satisfactory for all the models which have been tested. This group of statements can be replaced with the appropriate expressions for analytical derivatives if this is felt to

be desirable. In rare instances when a parameter value happens to fall close to zero, the perturbation used to calculate the derivative may be too small. There is no universal remedy to this situation but the program user should be aware of it. Obviously, a value of zero should never be used as an initial estimate of a parameter. A number of weighting options are available including equal weighting, $1/y$ weighting and $1/y^2$ weighting and others may be easily included. Alternatively, the standard deviation of y (or a factor proportional to it) can be specified explicitly. Each of these weighting options is available both with and without bisquare weighting. Iteration is continued until the sum of the absolute values of the relative changes in the parameter values (i.e., $\sum |q/b|$) is less than 10^{-5} at which time the program is considered to have converged. If convergence is not reached in 10 iterations, a warning is issued and the current values of the parameters are printed.

The program has been tested with a variety of models and three of these will be described. These do not represent the limit of flexibility of the program. For each model, the equations necessary for the median methods described under the Theory section are given as well as the median–unbiased combinations of the parameters.

Substrate saturation kinetics. Saturation of an enzyme by its substrate frequently obeys the familiar Michaelis–Menten equation [4] and the program shown in Fig. 1

$$\hat{y} = \frac{V \cdot x}{K + x}, \qquad [4]$$

will fit this equation to experimental data. The accuracy of numerical differentiation was assessed using this model by comparing the values of derivatives calculated by this method with those obtained using the analytical derivatives $\cdot y/\delta V = \delta\hat{y}/V$ and $\delta\hat{y}/\delta K = -\hat{y}^2/Vx$. Over a range of x values from zero to $800K$, the derivative for V calculated by the numerical method was found to be accurate to within 0.0003%, a not unexpected finding since the model is exactly linear in V. The derivative for K is somewhat less acccurate but in no case did the error exceed 0.05% of the value. Such errors are of no consequence when fitting to experimental data as is indicated using some data for the enzyme prephenate dehydratase (9), assuming constant standard deviation in y. The fit obtained using analytical derivatives gave[2] $V = 18.1554 \pm 0.4877$ U/mg and $K = 491.075 \pm 30.793$ μM while the corresponding values using numerical differentiation were 18.1555 ± 0.4876 U/mg and 491.078 ± 30.787 μM. For all practical purposes, the figures obtained by the two methods are identical.

If median methods are used to estimate the kinetic parameters, solutions for the resultant simultaneous equation are

$$K = (y_2 - y_1)/(y_1/x_1 - y_2/x_2)$$

$$V = (K + x_1)y_1/x_1.$$

The median–unbiased combinations are $1/V$ and K/V (5).

First-order decay. The equation for a first-order decay curve has been given previously (Eq. [1]) while the solutions required for median methods are

$$k = (\ln y_2 - \ln y_1)/(t_1 - t_2)$$

$$y_0 = y_1 \, e^{kt_1}.$$

Both k and y_0 are median–unbiased. Nimmo and Atkins (11) have compared various methods for analyzing this type of data and have observed that their computer program, which uses a rather sophisticated nonlinear regression method, failed to converge with 30–40 out of 500 sets of simulated data. This simulation was repeated using data containing normally distributed errors with a standard deviation equal to $\hat{y}^{1/2}$ (their case

[2] The absurd number of decimal places given is necessary to illustrate that analytical and numerical derivatives do, in fact, give different results. There is no suggestion intended that V and K are determined to an accuracy of six significant figures.

TABLE 1

STABILIZATION BY BISQUARE WEIGHTING OF THE FIT TO BINDING DATA WHEN AN OUTLIER IS PRESENT[a]

Value of y at $x = 3.0$	Without bisquare weighting		With bisquare weighting	
	K	N	K	N
2.8	1.701	1.061	0.955	0.996
2.6	1.311	1.021	0.955	0.996
2.5	1.170	1.009	0.958	0.996
2.4	1.054	1.001	0.994	0.998
2.3	0.957	0.996	0.963	0.998
2.2	0.876	0.993	0.919	0.996
2.1	0.808	0.993	0.952	0.996
2.0	0.749	0.994	0.955	0.996
1.8	0.653	1.001	0.955	0.996

[a] A simulated set of data was obtained by solving Eq. [8] for $\hat{y}$ at x values of 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0, and 10.0, assuming $K = N = 1$. These theoretical data were rounded to one decimal place and the point at $x = 3.0$ changed from this simulated value of 2.3 to the value indicated. These data were then fitted to Eq. [8] assuming constant variance both with and without bisquare weighting to obtain the values of K and N. Due to the errors introduced by rounding, the fit obtained using the value of 2.3 does not give values for K and N of exactly 1.

N2) and using weights[3] of $1/y^2$ (equivalent to their method WNL). The program described here failed to converge within 10 iterations for 30 of these sets of simulated data, so while the program is no better than that used by Nimmo and Atkins, it is no worse either. Ten of these failures were selected for further study and in each case satisfactory convergence could be achieved by allowing more than 10 iterations[4] or by adjusting the initial estimates of y_0 and k.

Binding equations. The binding of a ligand to a comparable concentration of an acceptor is described by Eq. [5] in which F and B represent the concentrations of free and

$$B = \frac{N \cdot F}{K + F} \qquad [5]$$

bound ligand, respectively, N is the total concentration of binding sites and K is the dissociation constant of the ligand–acceptor complex. This equation is similar in form to the Michaelis–Menten equation but now we must take account of the fact that there is significant depletion of free ligand by complex formation. Usually, only one of B or F will be measured while the other is calculated from the fact that B plus F equals the total ligand concentration (x). Thus we may consider two different cases depending on whether B or F is measured.

If B is the measured quantity (y), Eq. [5] is rewritten as Eq. [6] which, upon rearrangement, gives Eq. [7], a quadratic which

$$\hat{y} = \frac{N(x - \hat{y})}{K + x - \hat{y}} \qquad [6]$$

$$\hat{y}^2 - \hat{y}(K + N + x) + Nx = 0 \qquad [7]$$

may be solved for $\hat{y}$ by the usual methods. Solutions for the simultaneous equations generated by median methods are

$$K = \frac{(y_2 - y_1)(x_1 - y_1)(x_2 - y_2)}{y_1 x_2 - y_2 x_1}$$

$$N = \frac{y_1(K + x_1 - y_1)}{x_1 - y_1}.$$

The median–unbiased combinations of K and N are $1/N$ and K/N.

In the situation where unbound ligand is the measured quantity we again get a quadratic (Eq. [8]). Note that this equation may

$$\hat{y}^2 + \hat{y}(K + N - x) - Kx = 0 \qquad [8]$$

be obtained from Eq. [7] by interchanging K and N and reversing their signs and these same substitutions may be used to obtain the equations required for median methods. The median–unbiased combinations are, as before, $1/N$ and K/N.

The usefulness of bisquare weighting was

[3] These are not the correct weights for the error distribution that is being simulated. These incorrect weights are used so that the results could be compared with those of Nimmo and Atkins (11).

[4] The iteration limit may be changed by modifying line 1730. This has often been found to be necessary when the bisquare weighting option is selected.

assessed using some simulated data for the case where free ligand is measured and the results obtained are shown in Table 1. Without bisquare weighting, the fit is very sensitive to the presence of an outlier with K changing by 80% from its "true" value for a change of only 20% in the value of 1 of 10 data points. With bisquare weighting introduced, the fitted value of K responds to small deviations in the aberrant data point but larger deviations are essentially ignored. A similar but less pronounced effect is seen for the values of N. The relative insensitivity of N to the presence of an outlier is ascribed to the fact that the spurious data point occurs at a moderately small x value. Fitting (without bisquare weighting) to a data set with an outlier at a high x value gives rise to changes in N which are much larger than those seen in Table 1.

DISCUSSION

The analysis of experimental data in biochemistry, as in other quantitative sciences, frequently requires that the data be compared with a mathematical equation which describes an underlying theoretical model. In many instances this equation is nonlinear in the parameters and the appropriate method of analysis will involve fitting the equation to the data by nonlinear regression. The frequently used alternative of transforming the data into a linear form retains its popularity because nonlinear regression computer programs have not been designed with a laboratory environment in mind.

This paper presents a nonlinear regression computer program with a number of desirable features. It is written in BASIC, a language which is available on most computers and which is one of the simplest languages for the novice to understand. Those who are proficient in FORTRAN should be able to learn the elements of BASIC in a few hours. The program has been deliberately kept short to encourage implementation and to facilitate use on mini and microcomputers for which program size can impose limitations. The equation to be fitted to the data is specified in a single statement so changing to other equations is extremely easy. Partial derivatives are calculated by a numerical method to relieve the user of the sometimes onerous task of deriving these functions algebraically. A variety of weighting options are available which should cover most commonly encountered situations and others can be added easily. Finally, a bisquare weighting option is available which detects and reduces the effects of observations which deviate markedly from the fitted equation.

The output from the program consists of best-fit values of the parameters, standard errors of these values, and a comparison of the experimental data with the fitted equation. For a nonlinear equation these standard errors are only an approximate guide to the precision of the parameters and should be interpreted with this in mind. More reliable methods for estimating precision have been described (10) but these cannot be incorporated into the present program without a substantial increase in complexity. The aim was not to produce a program which could cope with any contingency, but rather to produce one which would be useful in many situations, and which is short and simple to use. Restricting the size of the program makes it inevitable that there will be limitations of capability but these are not excessive. The main limitation (and this is not fundamental) is that the program will only fit equations in which there are two parameters to be estimated and one independent variable. Clearly there will be some models for which the program cannot be used without substantial modification. The second limitation is that the program uses the Gauss–Newton method of nonlinear regression which is known to be ineffective when initial estimates of the parameters are not reasonably close to the best fit values. In spite of this, the program was found to perform as well as a much more sophisticated program on a test problem (11).

Results have been presented from fitting

three equations but these do not represent the limits of applicability. Other models which have been successfully fitted include the analysis of $C_d J$ curves (12) and the three compartment model

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C.$$

Further applications could include the determination of pK's and stability constants, the analysis of ultracentrifugation data and of the effects of temperature on enzymatic and chemical reactions. This list is by no means exhaustive; the only limit is the imagination of the user.

REFERENCES

1. Wilkinson, G. N. (1961) *Biochem. J.* **80,** 324–332.
2. Wahrendorf, J. (1979) *Int. J. Bio-Med. Comput.* **10,** 75–87.
3. Mosteller, F., and Tukey, J. W. (1977) *in* Data Analysis and Regression, pp. 353–365, Addison–Wesley, Reading, Mass.
4. Cornish-Bowden, A., and Eisenthal, R. (1974) *Biochem. J.* **139,** 721–730.
5. Cornish-Bowden, A., and Eisenthal, R. (1978) *Biochim. Biophys. Acta* **523,** 268–272.
6. Porter, W. R., and Trager, W. F. (1977) *Biochem. J.* **161,** 293–302.
7. Duggleby, R. G. (1979) *J. Theor. Biol.* **81,** 671–684.
8. Duggleby, R. G. (1980) *in* Design and Analysis of Enzyme and Pharmacokinetic Experiments (Endrenyi, L., ed.), Plenum Press. in press.
9. Duggleby, R. G., Sneddon, M. K., and Morrison, J. F. (1978) *Biochemistry* **17,** 1548–1554.
10. Duggleby, R. G. (1980) *Eur. J. Biochem.* **109,** 93–96.
11. Nimmo, I. A., and Atkins, G. L. (1979) *Anal. Biochem.* **94,** 270–273.
12. Britten, R. J., and Kohne, D. E. (1968) *Science* **161,** 529–540.