AN EFFICIENT ALGORITHM FOR AUTOMATIC SCALING OF AXES

by Timothy Eckert*

Often data generated by laboratory experiments are more easily analyzed when plotted on a graph. In fact, plotting programs are commonly included in computer programs which process data. A necessary part of such programs is the scaling of the axes to provide convenient minima, maxima and step sizes. Some plotting programs require the user to manually provide the desired minima, maxima and step sizes for the axes, a bothersome task. Other programs automatically scale axes, but either use the largest and smallest coordinates of the data points for the maxima and minima or else always set the minimum at zero. The former technique leads to axes scaled with awkward numbers, while the latter precludes graphing data points with negative coordinates. Moreover even if all the points have positive coordinates but are clustered relatively far from zero, an axis minimum at zero prevents differentiation of the points.

The algorithm in my program, SCALING, automatically sets the minima and maxima of the axes at positive or negative values not only to accomodate points with negative coordinates but also to allow greater distinction among points clustered far from zero. To demonstrate the versatility of this program the table below shows the scaling output for six types of data ranges (for simplicity, in one dimension only).

data minimum	data maximum	axis minimum	axis maximum	step size
0	0.9	0	1	0.1
1.2	14	0	16	2
900	1001	860	1020	20
-100	9	-140	20	20
-9	-6	-9.5	-4.5	0.5
-3	0	-3.5	0	0.5

SCALING is written in BASIC and is listed below. Its algorithm is not designed to stand alone, but instead to be readily incorporated into the user's own plotting program. In its present form it receives the input of the data minimum and maximum and outputs the appropriate axis minimum, maximum and step size.

Of course no algorithm can provide ideal, automatic scaling for all purposes. For example, if one wants to extrapolate a line far from the data points, perhaps to locate a distant axis intercept, this program which focuses on data points fails. An option for manual scaling axes should always accompany any algorithm for automatic scaling.

PROGRAM

```
  5  REM ALGORITHM FOR SCALING AXES
 10  PRINT "DATA MINIMUM, DATA MAXIMUM"
 20  INPUT N,X
 30  If X<>0 THEN 50
 40  X = -.1 ^9
 50  M = (1+SGN(X))*N/2+(SGN(X)-1)*X/2
 60  W = (1+SGN(X))*X/2+(SGN(X)-1)*N/2
 70  D = W*(1-(0.5^NT(LOG(W/(W-M))/LOG(2))))
 80  G = 10^(INT((LOG((W-D)/25))/LOG(10))
 90  S = G*5*2^INT(LOG((W-D)/(25*G))/LOG(2))
100  B = S*(INT(W/S)+1)
110  E = B+S*INT((D-B)/S)
120  F = (1+SGN(X))*E/2+(SGN(X)-1)*B/2
130  C = (1+SGN(X))*B/2+(SGN(X)-1)*E/2
140  PRINT "AXIS MINIMUM";F
150  PRINT "AXIS MAXIMUM";C
160  PRINT "STEP SIZE";S
170  END
```

DOCUMENTATION

This listing is the fully general form of the algorithm.

If the data maximum cannot equal 0, lines 30-40 may be deleted.

Lines 50, 60, 120, 130 invert parameters if the data maximum is negative. But if the data minimum cannot be negative, these lines may be deleted, and M, W, E, B may be replaced by N, X, F, C, respectively, throughout the program.

Lines 80-90 establish step size as the product of 1, 2, 4 or 5 and a power of 10 to insure 5-10 axis divisions. If more divisions are desired, the number 25 in lines 80-90 may be increased.

Lines 100-110 establish the axis maximum and minimum.

Some versions of BASIC may lead to an occasional deviation in output associated with arithmetic round-off errors in statements like line 90.

*Chemistry Department
Fredonia State University College
Fredonia, NY 14063