NBER WORKING PAPER SERIES

ROSEPACK Document No. 2:

AUTOMATING STEM-AND-LEAF DISPLAYS

David C. Hoaglin*
Stanley S. Wasserman**

Working Paper No. 109

COMPUTER RESEARCH CENTER FOR ECONOMICS AND MANAGEMENT SCIENCE
National Bureau of Economic Research, Inc.
575 Technology Square
Cambridge, Massachusetts 02139

November 1975

Preliminary

*NBER Computer Research Center and Harvard University. Research supported in part by National Science Foundation Grants DCR 75-08802 to the National Bureau of Economic Research and SOC 72-05257 to Harvard University.

**NBER Computer Research Center and Harvard University. Research supported in part by National Science Foundation Grant DCR 70-03456 A04 to the National Bureau of Economic Research.

Abstract

The stem-and-leaf display is a natural semi-graphic technique to include in statistical computing systems. This paper discusses the choices involved in implementing both automated and flexible versions of the display, develops an algorithm for the automated version, examines various implementation considerations, and presents a set of semi-portable FORTRAN subroutines for producing stem-and-leaf displays.

Table of Contents

1. Introduction

The stem-and-leaf display has so many uses in everyday data analysis that it should be a natural component of any modern statistical computing system. In implementing it, however, one must make many decisions. For example, how many lines of output should the display occupy? How should it be scaled to give a pleasant and effective appearance? How many different leaf digits should fall on one line -- ten, five, or two? Some of the answers doubtless involve personal taste, so the system must have enough flexibility to produce the display the user really wants. Why, then, should one consider automation, which would seem to deprive the user of that control? There is actually a strong justification for trying to produce automatically a reasonable stem-and-leaf display for a given batch of data. If the system is interactive, the user will usually need to ask for only minor changes in order to produce the finished display. More extensive trial would be easy, but it would generally be unnecessary. In a non-interactive system the much longer delay in receiving the next output places a premium on getting close to the "right" stem-and-leaf display on the first try so that at most one further iteration is necessary. In what follows we discuss several basic rules which lead "automatically" to a reasonable stem-and-leaf display. An appendix presents a set of "semi-portable" FORTRAN subroutines and discusses some of the details of implementation.

2. Stem-and-Leaf Displays

For starting to look at a batch or sample of data the stem-and-leaf display, developed by John W. Tukey [7,8], provides a flexible and effective technique. The basic idea is to let the most significant digits of the data values themselves do most of the work of sorting the batch into numerical order and displaying it. In the simplest form one chooses a suitable pair of adjacent digits in the data, splits each data value between these two digits, allocates a separate line in the display for each possible string of leading digits (the stem), and writes down the first trailing digit (the leaf) of each data value on the line corresponding to its leading digits. (The name "stem-and-leaf" comes about by analogy to espaliered trees or shrubs, which are trained so that their trunks grow vertically against a wall and their branches grow horizontally along it.) An example readily shows how the process works.

Frohliger and Kane [2] report the pH values for 26 samples of precipitation collected at a location in Allegheny County, Pennsylvania, from December 1973 to June 1974. In chronological order the data values are

4.57, 5.62, 4.12, 5.29, 4.64, 4.31, 4.30, 4.39, 4.45,
5.67, 4.39, 4.52, 4.26, 4.26, 4.40, 5.78, 4.73, 4.56,
5.08, 4.41, 4.12, 5.51, 4.82, 4.63, 4.29, 4.60.

In this case it is reasonable to split between the second and third digits; for example, 4.57 yields 4.5|7 . The necessary lines (17 in all) run from 4.1 to 5.7, and writing down the leaves in chronological order gives the raw display on the left in Exhibit 1. In the finished display the decimal points have been dropped in favor of a reminder that all data values are in units of .01, an occasional asterisk indicates that the leaves are one-digit, and a column of depths (which we will shortly define) has been added to the left of the stems. In overall appearance the display is similar to a histogram with an interval width of .1; the leaves add numerical detail.

Exhibit 1

Stem-and-Leaf Displays for Precipitation pH Data

raw

4.1	22
4.2	669
4.3	1099
4.4	501
4.5	726
4.6	430
4.7	3
4.8	2
4.9	
5.0	8
5.1	
5.2	9
5.3	
5.4	
5.5	1
5.6	27
5.7	8

finished

(unit = .01)

2	41*	22
5	42	669
9	43	1099
12	44*	501
(3)	45	726
11	46	430
8	47*	3
7	48	2
	49	
6	50*	8
	51	
5	52	9
	53*	
	54	
4	55	1
3	56*	27
1	57	8

A data value can be assigned a rank by counting in from each end of the batch. The depth of the data value is the smaller of these two ranks. Since a number of summary values (such as the median and the quartiles or the hinges) can easily be defined in terms of their depths, it is helpful to present a set of depths with the display. Except for one middle line, the number in the depth column is the maximum depth associated with data values on that line. Thus the depth of 4.29 is 5. The "middle line" (absent when the batch size is even and the median falls between lines) includes the median, and the depth column shows in parentheses the number of leaves (in the example, 3) on this line. If the display has been prepared by hand, adding the count on the middle line and the depths on the two adjacent lines provides a simple check that no data values have been omitted. In the example, 12 + 3 + 11 = 26.

Either display in Exhibit 1 reveals quite a lot about the behavior of the precipitation pH data: a rather flat distribution of values from 4.1 to 4.7 with scattered values trailing off above that to 5.3 and a clump of four values from 5.51 to 5.78. It is worth remarking that Frohliger and Kane give the range and average of their data but do not comment on the distribution of the sample, which hardly lends itself to such a simple summary and may suggest a mixture of two populations.

3. How Many Lines?

Experience suggests that an effective choice of the number of lines involves the number of data values in the batch as well as the range to be covered. In view of the similarities between stem-and-leaf displays and histograms, we should be able to calculate the maximum number of lines according to a rule given by Dixon and Kronmal [1]:

$$L = [10 \times \log_{10} n] ,$$

where n is the number of data values and [x] is the largest integer not exceeding x. (Interestingly, Dixon and Kronmal based their histogram rule on a suggestion by Tukey!) This seems to give very reasonable values of L over the range $20 \leq n \leq 300$, where almost all applications fall. Values of n smaller than 20 may need special treatment. These cases are also more likely to arise when comparing several batches in parallel stem-and-leaf displays, a situation we would want to handle differently anyway. Batches of 300 or so are usually cumbersome in a stem-and-leaf display, but the rule should still cope with them reasonably well. There is nothing sacred about the constant factor 10 in the definition of L, and further experience may lead to a different value. As an alternative rule, Velleman [9] has suggested $L = [2\sqrt{n}]$.

4. What Scaling?

Using L as a rough limit on the number of lines in the display, we must now determine the interval of values corresponding to each line. The simple way to do this (implicit in the earliest form of stem-and-leaf display, as in Exhibit 1) is to arrange that the interval width be a power of 10. This we can easily accomplish by dividing R, the range of the batch, by L and rounding the quotient up (if necessary) to the nearest power of ten. A segment of the stems for such a display might look like this:

```
0|
1|
2|
3|
```

and each line would receive leaves 0 through 9. It soon became clear that the display was sometimes too crowded, having too many leaves per line. Tukey's response was to split the lines and repeat each stem:

```
0*|
0•|
1*|
1•|
2*|
2•|
```

putting leaves 0 through 4 on the * line and 5 through 9 on the • line. In such a display the interval width is 5 times a power of 10.

This change made a great improvement, but there were still some cases in which the result was too crowded even in the split-stem form and too straggly in the original form at the next lower power of 10. To cure these troubles, Tukey added the third form, five lines per stem:

```
0*|
 t|
 f|
 s|
0·|
```

with leaves 0 and 1 on the * line, 2 and 3 on the t line, 4 and 5 on the f line, 6 and 7 on s line, and 8 and 9 on the · line. (As a reminder in starting to place leaves it is convenient that the three lettered lines contain leaves whose words begin with that letter.) Here the interval width is 2 times a power of 10.

5. A Scaling Algorithm

Viewing the three forms together, all we need to do is divide R by L and round the quotient up to 1, 2, or 5 times a power of 10. This rule of thumb is useful when one is preparing a stem-and-leaf display by hand, but we must formulate it as a specific algorithm for a computer. This is straightforward if we adapt the algorithm of Dixon and Kronmal [1].

Their algorithm, intended for scaling axes in graphs and histograms, uses a set of "round" numbers $p_1,\ldots,p_m$ such that $1 \leq p_1 < p_2 < \ldots < p_m < 10$. For scaling stem-and-leaf displays we need only $p_1 = 1$, $p_2 = 2$, $p_3 = 5$. To fix notation, we let

A = smallest value in the batch,

B = largest value in the batch, and

S = scale factor (the interval width)

and recall that

L = number of lines in the display, and

R = B - A.

Thus we want $L \times S \geq R$ with S as small as possible and $S = p_i \times 10^k$ for some i and k. Like Dixon and Kronmal, we begin by taking $t = [\log_{10}(R/L)]$. If $(R/L)/10^t \leq p_3 = 5$, we let k = t and find the smallest p_i which is at least $(R/L)/10^k$. Otherwise, k = t+1 and the desired p_i is $p_1 = 1$.

Next we calculate the number of stems actually required for the display and determine whether S must be increased. We easily find that the number of lines actually used is $\mathrm{sign}(B) \times [|B/S|] - \mathrm{sign}(A) \times [|A/S|] + 1$ (plus 1 if needed for the -0 stem). This can exceed L, and if it does, we replace S with the next larger "round" number, increasing k if necessary. This completes the scaling.

6. Stems, Units, and Leaves

In producing the display we must calculate the stems and then separate each data value into a starting part and a leaf. To do this, we recall that all entries in a stem-and-leaf display may be regarded as integer multiples of a power of 10, referred to as the unit. If we denote this unit by U, we can very easily calculate it from S: When $S = 2 \times 10^k$ or $S = 5 \times 10^k$, $U = 10^k$; and when $S = 10^k$, $U = 10^{k-1}$.

In calculating the stems (or more precisely, the labels on the lines) we must take into account the special role of zero. We implicitly number the lines of the display in steps of S/U from $\text{sign}(A) \times [|A/S|] \times (S/U)$ to $\text{sign}(B) \times [|B/S|] \times (S/U)$, including -0 if it occurs. We drop the low-order digit of each such number, and when S is 2 or 5 times a power of ten, that digit indicates which of the special labels such as t and • to use for an intermediate line.

Finally we are ready to separate each data value into a starting part and a leaf and collect the leaves on their proper stems. We will generally cut each data value to a multiple of U by simply discarding low-order digits. (This makes it easy to match an entry in the display with its original data value whenever that value requires further attention.) In that integer multiple, then, the last digit is the leaf, and the other digits are the starting part. For example, with U = .01 we would cut 2.213 to 2.21 and separate it into a starting part of 22 and a leaf of 1.

7. Some Refinements

In a sense this discussion has removed the stem-and-leaf display from its natural context, exploratory data analysis. Since resistant analyses (which are little affected by changes in a small fraction of the data) are an important part of the exploratory mode, it is quite unwise for an automated stem-and-leaf display to depend so heavily on the extreme values, A and B. Instead, we should begin by setting aside any serious outliers and basing A and B on those values which remain.

Since the batch will already be sorted (or will be sorted as the first step in producing a stem-and-leaf display), it is easy to isolate outliers. One useful rule of thumb [7] finds the upper and lower hinges (approximate quartiles) and their difference dH and sets aside any data values further than $1.5 \times dH$ from the nearer hinge. In printing the display we can add the lines "hi" and "lo" beyond the set of stems and list those outlying values.

Another important refinement lets us handle parallel displays (with one common set of stems) for comparing several batches. Here the tentative rule (subject to further experience) is to regard the individual batches as combined into a single batch for the purposes of scaling and forming stems. This evidently provides a set of stems which covers the combined range, and using the total number of data values to determine the number of lines will accommodate moderate shifts from batch to batch. (Large shifts may lead us to line up the batches by subtracting at least a rough adjustment from each.)

8. Examples

A few examples should clarify the suggestions for automating stem-and-leaf displays. We present four which are representative of a broad range of possibilities.

In the first example (Exhibit 2) all steps are straightforward. We can easily round R/L up to give S=5 without using logarithms, and the line requirement (10) does not exceed L (14). The display seems quite satisfactory.

If we do not check for possible outliers, the new feature in the second example (Exhibit 3) is the need to increase S from .02 to .05 in order to keep the line requirement within L. The resulting display may seem rather bunched on the stems 23* and 23· . A closer look uncovers four possible stray values at the low end and leads to a somewhat more reasonable display (Exhibit 4). Since the four low values do not stray far, we might also use 15 lines and avoid separating the low values from the rest of the batch.

Exhibit 5 shows how to handle +0 and -0 when the batch contains both positive and negative values. In this case no data values need to be set aside, and it is a simple task to construct the display.

The fourth and last example (Exhibit 6) shows both parallel displays and an obvious outlier. After we set aside the one straying value, the display is reasonably effective, but the straggling behavior of the third batch has clearly taken a toll.

All in all, the present "automated" approach usually seems to produce the stem-and-leaf displays one might prefer after some trial and error.

Exhibit 2

Data (hardness of aluminum die castings [6, p. 42])

53.0, 70.2, 84.3, 55.3, 78.5, 63.5, 71.4, 53.4,
82.5, 67.3, 69.5, 73.0, 55.7, 85.8, 95.4, 51.1,
74.4, 54.1, 77.8, 52.4, 69.1, 53.5, 64.3, 82.7,
55.7, 70.5, 87.5, 50.7, 72.3, 59.5

Calculations

$n = 30$, $L = [10 \times \log_{10} 30] = [14.77] = 14$

$A = 50.7$, $B = 95.4$, $R = B - A = 44.7$

$R/L = 44.7/14 = 3.19$; $S = 5$, $U = 1$

lines required $= [95.4/5] - [50.7/5] + 1 = 10$

Display

```
                  ( UNIT =    0.1000E+01 )

  7               5   I 0123334
 11               5.  I 5559
 13               6   I 34
  3               6.  I 799
 14               7   I 001234
  8               7.  I 78
  6               8   I 224
  3               8.  I 57
  1               9   I
  1               9.  I 5
```

Exhibit 3

Data

2.346, 2.334, 2.365, 2.417, 2.399, 2.354,
2.339, 2.368, 2.257, 2.358, 2.326, 2.334,
2.313, 2.298, 2.371, 2.197, 2.378, 2.395,
2.335, 2.207, 2.398, 2.211, 2.273, 2.213,
2.330, 2.359, 2.468, 2.352, 2.463, 2.398

Calculations

$n = 30$, $L = [10 \times \log_{10} 30] = 14$

$A = 2.197$, $B = 2.468$; $R = 0.271$

$R/L = 0.271/14 = 0.0194$; $S = .02$, $U = .01$

lines required = $[2.468/.02] - [2.197/.02] + 1 = 15$

Since this exceeds L, S must be increased to .05.

Display (unit = .01)

1	21·	9
4	22*	011
7	22·	579
15	23*	12333334
15	23·	555566779999
3	24*	1
2	24·	66

Exhibit 4

<u>Data</u> (in Exhibit 3)

<u>Calculations</u>

hinges: 2.313 and 2.378, dH = .065

cutoff values: 2.2155 and 2.4755

Set aside 2,197, 2.207, 2.211, and 2.213 to get A = 2.257 and B = 2.468 with n = 26 and L = 14.

R/L = 0.211/14 = .015; S = .02, U = .01

Lines required = 12

<u>Display</u>

```
  4          LO I     2.1970    2.2070    2.2110    2.2130

           ( UNIT =    0.1000E-01 )

  5            F I 5
  6            S I 7
  7          22. I 9
  8          23  I 1
 14            T I 233333
  5            F I 45555
 11            S I 6677
  7          23. I 9999
  3          24  I 1
  2            T I
  2            F I
  2            S I 66
```

Exhibit 5

Data

-1.4, 1.2, 0.5, -0.3, -0.8, 0.4, -1.3, 0.5,
0.7, -0.2, 0.1, 2.3, 0.1, -0.8, -2.6, 0.7
-0.9, -0.4, 0.2, 1.3

Calculations

$n = 20$, $L = [10 \times \log_{10} 20] = 13$

$A = -2.6$, $B = 2.3$; $R = 4.9$

$R/L = 4.9/13 = .38$; $S = .5$, $U = .1$

lines required $= [2.3/.5] + [2.6/.5] + 2 = 11$

Display

```
                 ( UNIT =    0.1000E+00 )
 1                -2.  I 6
 1                -2   I
 1                -1.  I
 3                -1   I 43
 6                -0.  I 988
 9                -0   I 432
 4                 0   I 1124
 7                 0.  I 5577
 3                 1   I 23
 1                 1.  I
 1                 2   I 3
```

Exhibit 6

Data (counties, including independent cities, by region in U.S. states)

Northeast: 8, 16, 14, 10, 21, 62, 67, 5, 14

North Central: 102, 92, 99, 105, 83, 87, 115, 93, 53, 88, 67, 72

South: 67, 75, 3, 67, 159, 120, 64, 24, 82, 100, 77, 46, 95, 254, 130, 55

West: 29, 14, 58, 63, 5, 44, 56, 17, 32, 36, 29, 39, 33

Calculations (scale as one batch)

Hinges: 24 and 88, dH = 64

cutoff values: -72 and 184

Set aside 254 (Texas) to get A = 3 and B = 159 with n = 49 and L = 16

R/L = 156/16; S = 10, U = 1

Lines required = 16

Display (unit = 1 county)

0*	85		3	5
1	6404			47
2	1		4	99
3*				2693
4			6	4
5		3	5	86
6*	27	7	774	3
7		2	57	
8		378	2	
9*		293	5	
10		25	0	
11		5		
12*			0	
13			0	
14				
15*			9	
hi			254	

9. Acknowledgments

The present implementation of stem-and-leaf displays has evolved from earlier one-line-per-stem versions programmed by Michael D. Godfrey for the instructional computing package SNAP/IEDA on the IBM 7094 and refined by Hale F. Trotter when SNAP/IEDA was converted to the IBM 360. We are also indebted to Paul W. Holland, Paul F. Velleman, and Roy E. Welsch for articulating the needs of a variety of users, to Neil E. Kaden and Virginia Klema for valuable discussions of semi-portability, and to Stephen C. Peters for assistance in testing and debugging.

References

[1] Dixon, W.J., and Kronmal, R.A., 1965, "The Choice of Origin and Scale for Graphs," Journal of the Association for Computing Machinery 12, 259-261.

[2] Frohliger, J.O., and Kane, R., 1975, "Precipitation: Its Acid Nature," Science 189, 455-457.

[3] Hoaglin, D.C., and Welsch, R.E., 1974, "MIT-SNAP, An Interactive Data Analysis System," Sloan School of Management, Massachusetts Institute of Technology.

[4] Ryder, B.G., 1974, "The PFORT Verifier," Software -- Practice and Experience 4, 359-377.

[5] Ryder, B.G., and Hall, A.D., 1975, "The PFORT Verifier," Computing Science Technical Report #12, Bell Laboratories, Murray Hill, New Jersey.

[6] Shewhart, W.A., 1931, Economic Control of Quality of Manufactured Product, D. Van Nostrand, Inc., Princeton, New Jersey.

[7] Tukey, J.W., 1970, Exploratory Data Analysis (Limited Preliminary Edition), Volume 1, Addison-Wesley, Reading, Massachusetts.

[8] Tukey, J.W., 1972, "Some Graphic and Semigraphic Displays," Statistical Papers in Honor of George W. Snedecor (T.A. Bancroft, editor), Iowa State University Press, Ames, Iowa.

[9] Velleman, P.F., 1975, "Interactive Computing for Exploratory Data Analysis I: Display Algorithms," American Statistical Association: Proceedings of the Statistical Computing Section, 1975, American Statistical Association, Washington, D.C.

APPENDIX

A FORTRAN Implementation

To illustrate various aspects of automating stem-and-leaf displays, we have developed a set of "semi-portable" FORTRAN subroutines. ("Semi-portable" implies that it should be possible to compile the subroutines under many different versions of FORTRAN, but that a few machine-dependent details remain. We have used the PFORT Verifier [4,5] to check adherence to PFORT, a large, carefully defined, portable subset of American National Standard FORTRAN. Only one departure from PFORT remains: for clarity we have retained subscript expressions involving more than one integer variable, as in "N-I+1".)

In addition to presenting the FORTRAN listings for the subroutines, this appendix briefly describes the subroutines and their roles in producing the stem-and-leaf display, discusses machine-dependent numerical aspects of the algorithm, explains the calculations of the depths printed with the display, reports on how stem-and-leaf has been installed as a command in a statistical computing system, and reviews the points at which error checking is desirable.

A1. Organization of the Subroutines

In this implementation the process of producing a stem-and-leaf display for a single batch of data has been modularized into four components (SLDSPY, SLSCAL, SLLEAF, and SLPRNT) and two utility routines (SLSORT and IFLOOR). This particular arrangement provides the flexibility for pre-chosen scaling and for parallel displays using a common set of stems.

SLDSPY (sequenced SLAB) is the driver routine. It takes as input the batch of data and the necessary scratch storage, and it controls the succeeding steps in producing the display. After calling SLSORT to sort the data, it checks for possible outliers and withholds them from the display. (Outliers are identified according to a simple rule of thumb based on the "hinges" (approximate quartiles) of the batch: set aside any data value further than 1 step beyond the hinge, where a step is 1.5 times the difference between the hinges.) SLDSPY then calls SLSCAL to scale the display, calls SLLEAF to lay out the stems and calculate the leaves, prints a heading and the low outliers, calls SLPRNT to print the display, and finally prints the high outliers. The stem-and-leaf display is communicated from SLLEAF to SLPRNT in four arrays: ISTEMS and LABELS together determine the starting part or stem to be printed on each line, LEAVES contains the single-digit leaves, and ILFCNT gives the number of leaves on each line of the display. Using this structure, one could display several batches side-by-side on the same page by having a common set of stems (in ISTEMS and LABELS) and one set of leaves (as in LEAVES and ILFCNT) for each batch. In practice it usually suffices to use a common scaling for the batches and print the displays one after another, letting the user cut and paste to achieve the desired effect.

SLSCAL (sequenced SLAD) uses the Dixon/Kronmal scaling algorithm to set the interval of values corresponding to all lines in the display at 1, 2, or 5 times a power of 10.

SLLEAF (sequenced SLAF) handles the calculations involved in setting up the starting parts (in ISTEMS and LABELS), converting each data value into a leaf, and placing that leaf on the proper line. The statements from line SLAF1210 to line SLAF1540 may need some explanation. Basically, the statements involving stems implement the implicit numbering of lines in steps of S/U from $\text{sign}(A) \times [|A/S|] \times (S/U)$ to $\text{sign}(B) \times [|B/S|] \times (S/U)$, as described in the section "Stems, Units, and Leaves". The built-in

FORTRAN function INT(x) has the same effect as sign(x) × [|x|], and the variable IS has the value S/U. The statement "KS1 = KSF / 10" discards the low-order digit of KSF and puts the digits of the stem in KS1. That low-order digit is recovered and saved in LAB. If the lowest data value in the display is negative, each negative stem is shifted down by 1 to allow for the stem "-0", which is represented by the value -1 in ISTEMS. When the number of lines per stem (equal to 10/IS) is 2 or 5, this representation of negative stems actually introduces a jump in the implicit numbering scheme, and the test for KS = -10 at line SLAF1530 resets KS so that the next iteration will continue with the stem "+0". The role of the factor "ONE + EPS" in calculating stems and leaves is discussed in the next section.

SLPRNT (sequenced SLAH) prints the numerical unit used in the display and then prints the display, accompanied at the left by the column of depths. More details on these depths are given in a later section. The parameter LINWID (passed from SLDSPY) provides flexibility in producing output for various devices (including interactive terminals and line printers) with different line widths. The value of LINWID is the number of spaces in the output line; if it is less than 24, there will be no room for any leaves, and a value greater than 123 may produce incorrect output (because FORMAT statements provide for at most 100 leaves per line).

SLSORT (sequenced SLAJ) is a conveniently available and fairly straightforward sorting routine.

IFLOOR (sequenced SLAL) is simply the greatest integer or "floor" function required by SLSCAL. (If desired, it could readily be placed in-line in SLSCAL.)

A2. Numerical Aspects

In implementing a semi-graphic technique one might expect to have no difficulty with numerical details. As stem-and-leaf algorithms illustrate, matters are not quite so simple, but fortunately the numerical considerations are few and straightforward. In a naive algorithm the most likely indication of difficulty is the appearance of a data value as an incorrect leaf, possibly on the wrong line. For example, a data value of 6.2 might appear with a leaf of 1 in the computer-produced display. The reason is that most computers represent numbers in a "binary" form (the common bases are 2 and 16) with a fixed number of digits. Since .2, for example, has a non-terminating binary expansion, it must be rounded to fit into the fixed-length computer word, and the result is a number slightly less than .2. The scaling calculations will not necessarily correct for this, and the leaf may appear incorrectly as 1.

Since the floating-point representation error takes the form of a relative error and causes difficulty in calculating leaves only when the error reduces the magnitude of the data value, we use the factor "ONE + EPS" in SLSCAL and SLLEAF to compensate for it. The machine-dependent constant EPS is a small positive number chosen to allow for representation error and the effects of succeeding calculation. For single-precision arithmetic on the IBM 360 or 370 the value EPS = 10^{-6} is a conservative choice which allows for an error of 1 in the next-to-last of the 6 hexadecimal digits in the floating-point fraction ($16^{-5} = 2^{-20} \approx 10^{-6}$). For a double-precision implementation on the same computers the corresponding value would be EPS = 2.5×10^{-16}.

Another numerical problem can arise when all data values agree in their first several digits. In this case the digits which determine the leaves may be affected by representation and roundoff errors, and a jumbled stem-and-leaf display may result. This is entirely a consequence of the finite-precision floating-point representation, and the best solution is for the user to drop the leading digits common to all data values before entering the data into the computer (otherwise the damage may already have been done). A conservative test for this problem is (in the notation of Section 5) to ask whether $S/\max(|A|, |B|) < 10 \times EPS$. If so, it is desirable to give the user a warning.

A3. Calculating the Depths

The information necessary to produce the column of depths is readily available by summing the entries of the array ILFCNT, and the calculations are handled in lines SLAH0920 to SLAH0940 in SLPRNT. If we let k_i denote the cumulative count of data values up through line i and n_i denote the number of values on line i (that is, n_i = ILFCNT(i)), then the depth to be printed for line i is the smaller of $k_{i-1} + n_i$ and $n - k_{i-1}$. The exceptional "middle" line is defined as having $|(n - k_{i-1} - n_i) - k_{i-1}| < n_i$ (that is, the count above the line and the count below the line differ by less than n_i -- it is straightforward to show that this happens on at most one line), and the number printed is n_i. To avoid using additional FORMAT statements, it is convenient to print the "depth" value on the "middle" line without parentheses and also to print the value in the depth column when the line has no leaves.

A4. Installation As a Command

The subroutines we have developed are the basis for the STEM command in the instructional package MIT-SNAP [3]. In designing such a command it is important to offer a high degree of automation (so that beginning students need not grapple with a long list of options) as well as considerable flexibility (so that more experienced users can produce the displays they want). Thus in MIT-SNAP the simplest request for a stem-and-leaf display produces the automated version we have described, including lists of outlying values at each end of the batch. Optional parameters enable the user to control

- the maximum number of lines in the display,
- the basic unit (power of ten) for the data,
- the number of lines per stem (only 1,2, and 5 are permitted),
- the cut-off values at each end (data values outside these are listed separately),
- the forced inclusion of all data values in the display, and
- the use of a common scaling for several batches.

Velleman [9] has described some alternative choices and options as used in LEDA.

A5. Error Checking

In preparing a program for general use, one should check for input which would cause errors or lead to garbage as output. An "automated" stem-and-leaf-display routine can avoid most problems by leaving few decisions to the user, but a few checks remain to be made:

- Is the number of data values too small ($n < 4$)?
- Is the working storage (ILFCNT, ISTEMS, LABELS, and LEAVES) large enough ($M \geq$ MSTEMS)?

- Is the output line too narrow (LINWID < 33) or too wide (LINWID > 123)?

- Are the largest and smallest data values to be displayed different (XHI ≠ XLO)?

- Does the data have so many significant digits that the leaves are likely to be affected by roundoff error?

The parameter IERR, returned by the driver routine SLDSPY, indicates which (if any) of these difficulties has occurred.

Anyone who uses the component subroutines to get a display different from the automated one is expected to assume the responsibility of checking for errors. In MIT-SNAP the STEM command provides much greater flexibility and checks its optional parameters for validity.

A6. FORTRAN Listings

```
      SUBROUTINE SLDSPY (X, ILFCNT, ISTEMS, LABELS, LEAVES, LINWID, N,  SLAB0010
     *   M, IERR)                                                        SLAB0020
C     *****PARAMETERS:                                                   SLAB0030
      INTEGER  N, M, IERR, ILFCNT (M), ISTEMS (M), LABELS (M),           SLAB0040
     *   LEAVES (N), LINWID                                              SLAB0050
      REAL  X (N)                                                        SLAB0060
C     *****LOCAL VARIABLES:                                              SLAB0070
      INTEGER  I, IHI, ILOW, IOUNIT, IS, J, J1, J2, K, MSTEMS, NN,       SLAB0080
     *   NSTEMS, NUMVAL                                                  SLAB0090
      REAL  DH, FL, FU, HL, HU, SCALE, TEN, THREEH, TWO, UNIT, XHI, XLO  SLAB0100
C     *****FUNCTIONS:                                                    SLAB0110
      INTEGER  INT, MINO                                                 SLAB0120
      REAL  ALOG10, FLOAT                                                SLAB0130
C                                                                        SLAB0140
C     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAB0150
C                                                                        SLAB0160
C     *****PURPOSE:                                                      SLAB0170
C     THIS SUBROUTINE PREPARES AND PRINTS A STEM-AND-LEAF DISPLAY FOR    SLAB0180
C     A BATCH X OF SIZE N, USING THE ALGORITHM DISCUSSED IN (1).         SLAB0190
C     STEM-AND-LEAF DISPLAYS ARE DESCRIBED IN (2).                       SLAB0200
C                                                                        SLAB0210
C     *****PARAMETER DESCRIPTION:                                        SLAB0220
C                                                                        SLAB0230
C     ON INPUT:                                                          SLAB0240
C                                                                        SLAB0250
C        X CONTAINS THE ELEMENTS OF THE BATCH TO BE DISPLAYED IN A       SLAB0260
C             STEM-AND-LEAF DISPLAY.  ONLY THOSE DATA VALUES LYING       SLAB0270
C             BETWEEN THE UPPER AND LOWER FENCES OF THE BATCH WILL BE    SLAB0280
C             USED IN THE DISPLAY.  VALUES OUTSIDE THE FENCES ARE        SLAB0290
C             PRINTED SEPARATELY ON EITHER THE 'HI' OR 'LO' DISPLAY      SLAB0300
C             LINES.                                                     SLAB0310
C                                                                        SLAB0320
C        LINWID MUST BE SET TO THE PAGE WIDTH OF THE PRINTED OUTPUT.     SLAB0330
C             LINWID SHOULD BE BETWEEN 33 AND 123.                       SLAB0340
C                                                                        SLAB0350
C        N MUST BE SET TO THE NUMBER OF ELEMENTS IN X.                   SLAB0360
C             N SHOULD BE GREATER THAN 3.                                SLAB0370
C                                                                        SLAB0380
C        M MUST BE AT LEAST THE NUMBER OF LINES IN THE DISPLAY.          SLAB0390
C             AS A GOOD APPROXIMATION,  10 * ALOG10 (N)   MAY BE USED.   SLAB0400
C             M MUST BE GREATER THAN OR EQUAL TO NSTEMS. THE CALCULATED  SLAB0410
C             NUMBER OF LINES IN THE DISPLAY.                            SLAB0420
```

```
C
C     ON OUTPUT:
C
C        IERR CONTAINS AN INTEGER BETWEEN 0 AND 5 AND IS USED FOR
C             ERROR CHECKING.  THE FOLLOWING TABLE LISTS POSSIBLE
C             VALUES FOR IERR AND THE CORRESPONDING ERRORS.
C
C              IERR       ERROR
C               0      NONE. DISPLAY SUCCESSFUL.
C               1      N TOO SMALL.
C               2      M TOO SMALL.
C               3      LINWID TOO SMALL OR TOO LARGE.
C               4      ZERO RANGE FOR DATA IN DISPLAY.
C               5      LEAF DIGITS TOO NEAR MACHINE ROUND-OFF LEVEL.
C
C     TEMPORARY STORAGE:
C
C        ILFCNT IS AN ARRAY OF LENGTH M.  IT MUST BE DIMENSIONED
C             IN THE  CALLING PROGRAM.  ILFCNT IS USED TO STORE
C             THE NUMBER OF LEAVES ON EACH LINE OF THE DISPLAY.
C
C        ISTEMS IS AN ARRAY OF LENGTH M.  IT MUST BE DIMENSIONED
C             IN THE  CALLING PROGRAM.  ISTEMS IS THE ARRAY OF
C             STARTING PARTS OR STEMS FOR THE DISPLAY.
C
C        LABELS IS AN ARRAY OF LENGTH M.  IT MUST BE DIMENSIONED
C             IN THE  CALLING PROGRAM.  LABELS IS THE ARRAY OF
C             'LABELS' (0, 2, 4, 5, 6, OR 8) FOR THE DISPLAY.
C
C        LEAVES IS AN ARRAY OF LENGTH N.  IT MUST BE DIMENSIONED
C             IN THE CALLING PROGRAM.  LEAVES IS THE ARRAY OF
C             LEAVES FOR THE DISPLAY.
C
C
C     *****APPLICATION AND USAGE RESTRICTIONS:
C     SLDSPY CALLS A SEQUENCE OF SUBROUTINES.  THE FOLLOWING IS A BRIEF
C     DESCRIPTION OF EACH ROUTINE PRESENTED IN THE  ORDER CALLED
C     BY SLDSPY.
C        SLSORT SORTS THE DATA IN NONDECREASING ORDER.
C        SLSCAL DETERMINES THE SCALE FACTOR AND UNIT FOR THE DISPLAY.
C        SLLEAF DETERMINES THE STEMS AND LEAVES.
C        SLPRNT HANDLES DISPLAY  PRINTING.
```

```
C     SLDSPY CALCULATES THE 'LO' AND 'HI' DATA VALUES, PRINTS THESE        SLAB0850
C     VALUES, AND FINDS THE MAXIMUM NUMBER OF LINES IN THE DISPLAY.        SLAB0860
C                                                                          SLAB0870
C     *****ALGORITHM NOTES:                                                SLAB0880
C     LINES IN THE SUBROUTINE CONTAINING 'COP' IN THE FIRST THREE          SLAB0890
C     COLUMNS WILL HELP THE USER IN CONVERTING THE ROUTINE FROM SINGLE     SLAB0900
C     TO DOUBLE PRECISION. 'REAL' DEFINITION CARDS MUST BE REMOVED.        SLAB0910
C                                                                          SLAB0920
C     IN ADDITION, ALL CONSTANTS INITIALIZED IN THE SUBROUTINE             SLAB0930
C     SHOULD BE CHANGED FROM 'E' TO 'D' FOR DOUBLE-PRECISION               SLAB0940
C     CONVERSION.                                                          SLAB0950
C                                                                          SLAB0960
C     IOUNIT CONTAINS THE DATA SET REFERENCE NUMBER FOR THE WRITE          SLAB0970
C     STATEMENTS.  IOUNIT IS INITIALLY SET TO '6'  IN LINE                 SLAB0980
C     SLAB1470, BUT MAY BE CHANGED BY THE USER TO AN INTEGER BETWEEN       SLAB0990
C     1 AND 99 INCLUSIVE (THE UPPER LIMIT IS SPECIFIED BY THE              SLAB1000
C     INSTALLATION AND MAY BE LESS THAN 99).                               SLAB1010
C                                                                          SLAB1020
C     *****REFERENCES:                                                     SLAB1030
C     (1)    HOAGLIN, D.C. AND WASSERMAN, S.S., 'AUTOMATING                SLAB1040
C        STEM-AND-LEAF DISPLAYS'.  NBER WORKING PAPER SERIES.  WORKING     SLAB1050
C        PAPER NO. 109.  NOVEMBER 1975.                                    SLAB1060
C                                                                          SLAB1070
C     (2)    TUKEY, J.W., 'SOME GRAPHIC AND SEMIGRAPHIC DISPLAYS'.         SLAB1080
C        STATISTICAL PAPERS IN HONOR OF GEORGE W. SNEDECOR   (T.A.         SLAB1090
C        BANCROFT, EDITOR).  AMES, IOWA:  IOWA STATE UNIVERSITY            SLAB1100
C        PRESS, 1972.                                                      SLAB1110
C                                                                          SLAB1120
C     *****HISTORY:                                                        SLAB1130
C     WRITTEN BY STANLEY WASSERMAN AND DAVID HOAGLIN (NBER COMPUTER        SLAB1140
C     RESEARCH CENTER)  10 DECEMBER 1974.                                  SLAB1150
C                                                                          SLAB1160
C     DATE LAST MODIFIED:   4 DECEMBER 1975.                               SLAB1170
C                                                                          SLAB1180
C     *****GENERAL:                                                        SLAB1190
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO:                        SLAB1200
C        SUPPORT STAFF MANAGER                                             SLAB1210
C        COMPUTER RESEARCH CENTER FOR ECONOMICS AND MANAGEMENT SCIENCE     SLAB1220
C        NATIONAL BUREAU OF ECONOMIC RESEARCH                              SLAB1230
C        575 TECHNOLOGY SQUARE, NINTH FLOOR                                SLAB1240
C        CAMBRIDGE, MASS.    02139                                         SLAB1250
C                                                                          SLAB1260
```

```
C     RESEARCH SUPPORTED IN PART BY NSF GRANT NUMBER DCR70-03456 A04
C
C
C     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
CDP   DOUBLE PRECISION DH, DY, FL, FU, HL, HU, SCALE, TEN, THREEH, TWO,
CDP  *   UNIT, X (N), XHI, XLO
CDP   DOUBLE PRECISION ALOG10, FLOAT
C
CDP   ALOG10 (DY) = DLOG10 (DY)
CDP   INT (DY) = IDINT (DY)
CDP   FLOAT (I) = DFLOAT (I)
C
C     ::::::::  CONSTANT INITIALIZATION  ::::::::
      THREEH = 1.50E0
      TWO = 2.0E0
      TEN = 10.0E0
C
C     ::::::::  INITIALIZE IERR AND IOUNIT  ::::::::
      IERR = 0
      IOUNIT = 6
C
C     ::::::::  CHECK THAT N IS GREATER THAN 3  ::::::::
      IF ( N .GT. 3 )  GO TO 5
         IERR = 1
         RETURN
C
C     ::::::::  CHECK THAT LINWID IS BETWEEN 33 AND 123  ::::::::
    5 IF ( LINWID .GE. 33 .AND. LINWID .LE. 123 )  GO TO 10
         IERR = 3
         RETURN
C
   10 CALL  SLSORT (X, N)
C
C     ::::::::
C
C        DETERMINE THE LARGEST AND SMALLEST DATA VALUES FOR THE
C        DISPLAY.  X (ILOW) AND X (IHI) ARE THE FENCES, LOCATED
C        ONE 'STEP' BEYOND THE HINGES.
C
C     ::::::::
C
```

```
C     ::::::::  CALCULATE HINGES  ::::::::                                SLAB1690
      J = (N / 4) + 1                                                     SLAB1700
      K = N - J + 1                                                       SLAB1710
      HL = X (J)                                                          SLAB1720
      HU = X (K)                                                          SLAB1730
      IF ( (N + 1)  .NE.  (4 * J) )  GO TO 20                             SLAB1740
         HL = (X (J + 1) + HL) / TWO                                      SLAB1750
         HU = (X (K - 1) + HU) / TWO                                      SLAB1760
C                                                                         SLAB1770
C     ::::::::  DETERMINE H-SPREAD (DH) AND FENCES  ::::::::              SLAB1780
   20 DH = HU - HL                                                        SLAB1790
      FU = HU + THREEH * DH                                               SLAB1800
      FL = HL - THREEH * DH                                               SLAB1810
C                                                                         SLAB1820
C     ::::::::  FIND VALUE ADJACENT TO LOWER FENCE  ::::::::              SLAB1830
      DO  30  I = 1, J                                                    SLAB1840
         IF ( X (I) .GE. FL )  GO TO 40                                   SLAB1850
   30 CONTINUE                                                            SLAB1860
C                                                                         SLAB1870
   40 ILOW = I                                                            SLAB1880
      XLO = X (ILOW)                                                      SLAB1890
C                                                                         SLAB1900
C     ::::::::  FIND VALUE ADJACENT TO UPPER FENCE  ::::::::              SLAB1910
      DO  50  I = 1, J                                                    SLAB1920
         IF ( X (N - I + 1) .LE. FU )  GO TO 60                           SLAB1930
   50 CONTINUE                                                            SLAB1940
C                                                                         SLAB1950
   60 IHI = N - I + 1                                                     SLAB1960
      XHI = X (IHI)                                                       SLAB1970
C                                                                         SLAB1980
C     ::::::::  CHECK THAT DISPLAY DOES NOT HAVE ZERO RANGE  ::::::::     SLAB1990
      IF ( XHI .NE. XLO )  GO TO 70                                       SLAB2000
         IERR = 4                                                         SLAB2010
         RETURN                                                           SLAB2020
C                                                                         SLAB2030
C     ::::::::  NN IS THE NUMBER OF ENTRIES IN THE DISPLAY ::::::::       SLAB2040
C                                                                         SLAB2050
   70 NN = IHI - ILOW + 1                                                 SLAB2060
C                                                                         SLAB2070
C     ::::::::  CALCULATE MAXIMUM NUMBER OF LINES  ::::::::               SLAB2080
C                                                                         SLAB2090
      MSTEMS = INT (TEN * ALOG10 (FLOAT (NN)))                            SLAB2100
```

```
C                                                                       SLAB2110
      CALL  SLSCAL (XHI, XLO, MSTEMS, IERR, IS, NSTEMS, SCALE, UNIT)    SLAB2120
C                                                                       SLAB2130
C     ::::::::  CHECK FOR ABNORMAL EXIT FROM SLSCAL  ::::::::           SLAB2140
      IF ( IERR .EQ. 5 )  RETURN                                        SLAB2150
C                                                                       SLAB2160
      IF ( M .GE. NSTEMS ) GO TO 80                                     SLAB2170
         IERR = 2                                                       SLAB2180
         RETURN                                                         SLAB2190
C                                                                       SLAB2200
   80 CALL  SLLEAF (X, ILOW, IHI, N, NN, NSTEMS, IS, SCALE, UNIT,       SLAB2210
     *   ILFCNT, ISTEMS, LABELS, LEAVES)                                SLAB2220
C                                                                       SLAB2230
      WRITE (IOUNIT,500)  N                                             SLAB2240
C                                                                       SLAB2250
C     ::::::::  DETERMINE NUMVAL, THE NUMBER OF HIGH OR LOW VALUES      SLAB2260
C                   ALLOWED PER LINE  ::::::::                          SLAB2270
      NUMVAL = (LINWID - 23) / 10                                       SLAB2280
C                                                                       SLAB2290
C     :::::::::  PRINT LOW VALUES  ::::::::                             SLAB2300
C                                                                       SLAB2310
      K = ILOW - 1                                                      SLAB2320
      IF ( K .LT. 1 )  GO TO 100                                        SLAB2330
         J2 = 0                                                         SLAB2340
   90    IF ( J2 .EQ. K )  GO TO 100                                    SLAB2350
            J1 = J2 + 1                                                 SLAB2360
            J2 = J2 + MIN0 (NUMVAL, (K - J1 + 1))                       SLAB2370
            IF ( J1 .EQ. 1 )                                            SLAB2380
     *         WRITE (IOUNIT,600)  K, (X (I), I = J1, J2)               SLAB2390
            IF ( J1 .NE. 1 )                                            SLAB2400
     *         WRITE (IOUNIT,610)  (X (I), I = J1, J2)                  SLAB2410
            GO TO 90                                                    SLAB2420
C                                                                       SLAB2430
  100 WRITE (IOUNIT,700)                                                SLAB2440
      CALL  SLPRNT (ILFCNT, ILOW, IOUNIT, ISTEMS, LABELS, LEAVES,       SLAB2450
     *   LINWID, N, NN, NSTEMS, UNIT)                                   SLAB2460
C                                                                       SLAB2470
C     ::::::::  PRINT HIGH VALUES  ::::::::                             SLAB2480
C                                                                       SLAB2490
      K = IHI + 1                                                       SLAB2500
      J = N - IHI                                                       SLAB2510
      IF ( K .GT. N )  GO TO 120                                        SLAB2520
```

```
      J2 = K - 1
110   IF ( J2 .EQ. N )  GO TO 120
        J1 = J2 + 1
        J2 = J2 + MINO (NUMVAL, (N - J1 + 1))
        IF ( J1 .EQ. K )
     *     WRITE (IOUNIT,601)  J, (X (I), I = J1, J2)
        IF ( J1 .NE. K )
     *     WRITE (IOUNIT,610)  (X (I), I = J1, J2)
        GO TO 110
C
  120 RETURN
C
500   FORMAT (1X ///  10X,  27HSTEM-AND-LEAF DISPLAY,  N = , I6  ///)
600   FORMAT (1X, I5, 13X, 5HLO I , 10F10.4)
601   FORMAT (1X // 1X, I5, 13X,  5HHI I , 10F10.4)
610   FORMAT (24X, 10F10.4)
700   FORMAT (1H )
C
C     ::::::::   LAST LINE OF SLDSPY  ::::::::
      END
```

```
      SUBROUTINE  SLSCAL (XHI, XLO, MSTEMS, IERR, IS, NSTEMS, SCALE,
     *   UNIT)
C     *****PARAMETERS:
      INTEGER  MSTEMS, IERR, IS, NSTEMS
      REAL  XHI, XLO, SCALE, UNIT
C     *****LOCAL VARIABLES:
      INTEGER  I, IA, IB, K
      REAL  EPS, FIVE, ONE, RLOG2, RLOG5, T, TEN, TWO, Y, ZERO
C     *****FUNCTIONS:
      INTEGER IFLOOR, INT
      REAL  ABS, ALOG10, AMAX1, FLOAT
C
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     *****PURPOSE:
C     THIS SUBROUTINE CALCULATES THE 'DEFAULT' SCALE FACTOR AND
C     UNIT, THE EXACT NUMBER OF LINES, AND THE NUMBER OF LINES PER
C     STEM IN A STEM-AND-LEAF DISPLAY.
C
C     *****PARAMETER DESCRIPTION:
C
C     ON INPUT:
C
C        XHI CONTAINS THE LARGEST DATA VALUE USED IN THE DISPLAY.
C
C        XLO CONTAINS THE SMALLEST DATA VALUE USED IN THE DISPLAY.
C
C        MSTEMS CONTAINS THE MAXIMUM NUMBER OF LINES IN THE DISPLAY.
C             THIS VALUE IS CALCULATED IN THE SUBROUTINE SLDSPY.
C
C     ON OUTPUT:
C
C        IERR IS USED BY THE SUBROUTINE SLDSPY FOR ERROR CHECKING.
C             IF THE LEAF DIGITS ARE TOO CLOSE TO THE ROUND-OFF
C             LEVEL OF THE MACHINE, IERR IS SET TO 5 AND CONTROL IS
C             RETURNED TO THE CALLING SUBROUTINE.
C
C        IS, WHOSE INTEGER VALUE IS SCALE/UNIT, CONTAINS EITHER 10, 5,
C              OR 2.  THE NUMBER OF LINES PER STEM IS (10 / IS).
C
C        NSTEMS CONTAINS THE EXACT NUMBER OF LINES.
C
```

```
C         SCALE CONTAINS THE SCALE FACTOR, OR NUMERICAL VALUE OF ONE       SLAD0430
C              LINE, IN THE DISPLAY.                                        SLAD0440
C                                                                           SLAD0450
C         UNIT CONTAINS THE APPROPRIATE POWER OF 10 FOR THE DISPLAY.        SLAD0460
C                                                                           SLAD0470
C      *****APPLICATION AND USAGE RESTRICTIONS:                             SLAD0480
C      SLSCAL IS CALLED FROM THE SUBROUTINE SLDSPY.  THE CALCULATIONS       SLAD0490
C      PERFORMED HERE ARE NECESSARY IN THE ACTUAL COMPUTING OF THE STEMS SLAD0500
C      AND LEAVES FOR A STEM-AND-LEAF DISPLAY.                              SLAD0510
C                                                                           SLAD0520
C      THE ALGORITHM USED IN FINDING THE SCALE FACTOR IS FOUND IN (1).      SLAD0530
C                                                                           SLAD0540
C      XHI SHOULD BE DISTINCT FROM XLO, AS THE SUBROUTINE COMPUTES THE      SLAD0550
C      BASE 10 LOGARITHM OF THEIR DIFFERENCE.                               SLAD0560
C                                                                           SLAD0570
C      *****ALGORITHM NOTES:                                                SLAD0580
C      LINES IN THE SUBROUTINE CONTAINING 'CDP' IN THE FIRST THREE          SLAD0590
C      COLUMNS WILL HELP THE USER IN CONVERTING THE ROUTINE FROM SINGLE     SLAD0600
C      TO DOUBLE-PRECISION. 'REAL' DEFINITION CARDS MUST BE REMOVED.        SLAD0610
C                                                                           SLAD0620
C      IN ADDITION, ALL CONSTANTS INITIALIZED IN THE SUBROUTINE             SLAD0630
C      SHOULD BE CHANGED FROM 'E' TO 'D' FOR DOUBLE-PRECISION               SLAD0640
C      CONVERSION.                                                          SLAD0650
C                                                                           SLAD0660
C      EPS IS A MACHINE-DEPENDENT TOLERANCE WHICH ALLOWS FOR ERRORS         SLAD0670
C      INHERENT IN REPRESENTING DECIMAL NUMBERS IN 'BINARY' BASES.          SLAD0680
C      (IT ALLOWS ROUGHLY THE LAST FOUR BITS OF THE FLOATING POINT          SLAD0690
C      FRACTION TO BE AFFECTED BY 'REPRESENTATION ERROR'.)                  SLAD0700
C                                                                           SLAD0710
C      FOR SUCCESSFUL CALCULATION OF THE LEAVES,                            SLAD0720
C      (SCALE / MAX ( ABS (XHI), ABS (XLO)) SHOULD BE GREATER               SLAD0730
C      THAN 10 * EPS.                                                       SLAD0740
C                                                                           SLAD0750
C      *****REFERENCES:                                                     SLAD0760
C      (1)    DIXON, W.J. AND KRONMAL, R.A., 'THE CHOICE OF ORIGIN AND      SLAD0770
C         SCALE FOR GRAPHS', JOURNAL OF THE ACM, VOL. 12 (1965),            SLAD0780
C         PAGE 259.                                                         SLAD0790
C                                                                           SLAD0800
C      :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAD0810
C                                                                           SLAD0820
CDP    DOUBLE PRECISION  DY, EPS, FIVE, ONE, RLOG2, RLOG5,                  SLAD0830
CDP   *   SCALE, T, TEN, TWO, UNIT, XHI, XLO, Y, ZERO                       SLAD0840
```

```
CDP   DOUBLE PRECISION  ABS, ALOG10, AMAX1, FLOAT
C
CDP   ABS (DY) = DABS (DY)
CDP   ALOG10 (DY) = DLOG10 (DY)
CDP   AMAX1 (DY) = DMAX1 (DY)
CDP   INT (DY) = IDINT (DY)
CDP   FLOAT (I) = DFLOAT (I)
C
C     ::::::::  CONSTANT INITIALIZATION  ::::::::
      TEN = 10.0E0
      FIVE = 5.0E0
      TWO = 2.0E0
      ONE = 1.0E0
      ZERO = 0.0E0
C     ::::::::  SET EPS, MACHINE-DEPENDENT TOLERANCE  ::::::::
      EPS = 1.0E-6
C
      RLOG2 = ALOG10 (TWO)
      RLOG5 = ALOG10 (FIVE)
C
C     ::::::::  THE FOLLOWING LINES DETERMINE K, THE CORRECT EXPONENT
C                   OF 10 FOR UNIT.  ::::::::
C
      T = ALOG10 ((XHI - XLO) / FLOAT (MSTEMS))
C     ::::::::   IFLOOR SIMULATES  FLOOR FUNCTION  ::::::::
      K = IFLOOR (T)
      Y = T - FLOAT (K)
C
C     ::::::::
C
C        NEXT CALCULATE SCALE, UNIT, AND IS, THE CORRECT
C        MULTIPLE OF UNIT TO DETERMINE THE VALUE OF EACH LINE
C        IN THE DISPLAY.
C
C     ::::::::
C
      IF ( Y .LE. RLOG5 )  GO TO 10
         IS = 10
         GO TO 20
   10 IS = 5
      IF ( Y .LE. RLOG2 )  IS = 2
      IF ( Y .GT. ZERO )  GO TO 20
```

```
         IS = 10
         K = K - 1
   20 UNIT = TEN ** K
      SCALE = FLOAT (IS) * UNIT
C
C     ::::::::
C
C        NSTEMS IS DETERMINED NEXT.  IF NSTEMS EXCEEDS MSTEMS,
C        SCALE MUST BE INCREASED TO THE NEXT  'ROUND'
C        NUMBER.
C
C     ::::::::
C
      IA = INT ((XLO * (ONE + EPS)) / SCALE)
      IB = INT ((XHI * (ONE + EPS)) / SCALE)
      NSTEMS = IB - IA + 1
      IF ( (XHI * XLO) .LT. ZERO  .OR.  XHI .EQ. ZERO )
     *   NSTEMS = NSTEMS + 1
      IF ( NSTEMS .LE. MSTEMS )  GO TO 50
C        ::::::::  INCREASE SCALE  ::::::::
         IF ( IS .NE. 2 )  GO TO 30
            IS = 5
            SCALE = FIVE * UNIT
            GO TO 40
   30    IF ( IS .EQ. 10 )  GO TO 35
            IS = 10
            SCALE = TEN * UNIT
            GO TO 40
   35       IS = 2
            UNIT = TEN * UNIT
            SCALE = TWO * UNIT
C           ::::::::  RECALCULATE NSTEMS WITH NEW SCALE  ::::::::
   40       IA = INT ((XLO * (ONE + EPS)) / SCALE)
            IB = INT ((XHI * (ONE + EPS)) / SCALE)
            NSTEMS = IB - IA + 1
            IF ( (XHI * XLO) .LT. ZERO  .OR.  XHI .EQ. ZERO )
     *         NSTEMS = NSTEMS + 1
C
C     ::::::::  CHECK THAT LEAF DIGITS ARE NOT TOO CLOSE TO
C                  ROUND-OFF LEVEL  ::::::::
C
   50 IF ( (SCALE / AMAX1 (ABS (XHI), ABS (XLO))) .LT. TEN * EPS )
```

```
     *    IERR = 5                                                     SLAD1690
C                                                                      SLAD1700
      RETURN                                                           SLAD1710
C     :::::::: LAST LINE OF SLSCAL  ::::::::                           SLAD1720
      END                                                              SLAD1730
```

```
      SUBROUTINE  SLLEAF (X, ILOW, IHI, N, NN, NSTEMS, IS, SCALE,
     *   UNIT, ILFCNT, ISTEMS, LABELS, LEAVES)
C     *****PARAMETERS:
      INTEGER  ILOW, IHI, N, NN, NSTEMS, IS, ILFCNT (NSTEMS),
     *   ISTEMS (NSTEMS), LABELS (NSTEMS), LEAVES (NN)
      REAL  X (N), SCALE, UNIT
C     *****LOCAL VARIABLES:
      INTEGER  I, J, KL, KL1, KS, KSF, KSL, KS1, LAB, N1
      REAL  EPS, ONE, ZERO
C     *****FUNCTIONS:
      INTEGER  INT, IABS
C
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     *****PURPOSE:
C     THIS SUBROUTINE DETERMINES THE STEMS AND LEAVES FOR A
C     STEM-AND-LEAF DISPLAY.  DATA VALUES ARE 'CUT' TO DETERMINE THE
C     LEAVES.
C
C     *****PARAMETER DESCRIPTION:
C
C     ON INPUT:
C
C        X CONTAINS THE SORTED ELEMENTS OF THE BATCH FOR THE STEM-
C             AND-LEAF DISPLAY. THE SORTING IS DONE BY THE
C             SUBROUTINE SLSORT.
C
C        ILOW CONTAINS THE INDEX FOR THE SMALLEST DATA VALUE IN
C             X TO BE USED IN THE DISPLAY.
C
C        IHI CONTAINS THE INDEX FOR THE LARGEST DATA VALUE IN
C             X TO BE USED IN THE DISPLAY.
C
C        N MUST BE SET TO THE NUMBER OF ELEMENTS IN X.
C
C        NN MUST BE SET TO THE NUMBER OF ELEMENTS IN THE DISPLAY.
C             I.E. NN = IHI - ILOW + 1.  NN IS CALCULATED IN SLDSPY.
C
C        NSTEMS CONTAINS THE NUMBER OF LINES FOR THE DISPLAY, AS
C             COMPUTED IN THE SUBROUTINE SLSCAL.
C
C        IS CONTAINS THE INTEGER VALUE OF SCALE/UNIT,  10, 5, OR 2.
```

```
C        THE NUMBER OF LINES PER STEM IS 10/IS.  IS  IS COMPUTED     SLAF0430
C        IN SLSCAL.                                                  SLAF0440
C                                                                    SLAF0450
C     SCALE CONTAINS THE SCALE FACTOR, OR NUMERICAL VALUE OF EACH    SLAF0460
C        LINE.  SCALE IS ALSO COMPUTED IN SLSCAL.                    SLAF0470
C                                                                    SLAF0480
C     UNIT CONTAINS THE APPROPRIATE POWER OF 10 FOR THE DISPLAY.     SLAF0490
C        UNIT IS COMPUTED IN SLSCAL.                                 SLAF0500
C                                                                    SLAF0510
C  ON OUTPUT:                                                        SLAF0520
C                                                                    SLAF0530
C     ILFCNT IS AN ARRAY OF LENGTH NSTEMS.  IT CONTAINS THE NUMBER   SLAF0540
C        OF LEAVES ON EACH LINE OF THE DISPLAY.  ILFCNT IS USED      SLAF0550
C        BY THE SUBROUTINE SLPRNT IN DETERMINING HOW THE LEAVES      SLAF0560
C        ARE TO BE PLACED ON THE STEMS.                              SLAF0570
C                                                                    SLAF0580
C     ISTEMS IS AN ARRAY OF LENGTH NSTEMS, CONTAINING THE 'STEM'     SLAF0590
C        PART OF EACH LINE IN THE DISPLAY.                           SLAF0600
C                                                                    SLAF0610
C     LABELS IS ALSO AN ARRAY OF LENGTH NSTEMS.  STORED IN           SLAF0620
C        LABELS ARE THE VALUES 0, 2, 4, 5, 6, OR 8 WHICH ARE         SLAF0630
C        USED BY SLPRNT TO DETERMINE THE 'TAG' (T, F, OR S) TO       SLAF0640
C        PLACED ON EACH LINE.                                        SLAF0650
C                                                                    SLAF0660
C     LEAVES IS AN ARRAY OF LENGTH NN CONTAINING THE 'LEAF'          SLAF0670
C        PART OF EACH DATA VALUE IN THE DISPLAY.                     SLAF0680
C                                                                    SLAF0690
C  *****APPLICATION AND USAGE RESTRICTIONS:                          SLAF0700
C  SLLEAF IS CALLED BY THE SUBROUTINE SLDSPY.  IT USES THE BATCH OF  SLAF0710
C  DATA SORTED BY SLSORT AND THE OUTPUT OF SLSCAL (NSTEMS, IS,       SLAF0720
C  SCALE, AND UNIT) TO PRODUCE THE STEMS, LABELS, AND                SLAF0730
C  LEAVES FOR THE DISPLAY.  THESE ARRAYS ARE PASSED TO SLPRNT FOR    SLAF0740
C  PRINTING THE STEM-AND-LEAF DISPLAY.                               SLAF0750
C                                                                    SLAF0760
C  *****ALGORITHM NOTES:                                             SLAF0770
C  LINES IN THE SUBROUTINE CONTAINING 'CDP' IN THE FIRST THREE       SLAF0780
C  COLUMNS WILL HELP THE USER IN CONVERTING THE ROUTINE FROM SINGLE  SLAF0790
C  TO DOUBLE-PRECISION. 'REAL' DEFINITION CARDS MUST BE REMOVED.     SLAF0800
C                                                                    SLAF0810
C  IN ADDITION, ALL CONSTANTS INITIALIZED IN THE SUBROUTINE          SLAF0820
C  SHOULD BE CHANGED FROM 'E' TO 'D' FOR DOUBLE-PRECISION            SLAF0830
C  CONVERSION.                                                       SLAF0840
```

```
C                                                                         SLAF0850
C     EPS IS A MACHINE-DEPENDENT TOLERANCE WHICH ALLOWS FOR ERRORS        SLAF0860
C     INHERENT IN REPRESENTING DECIMAL NUMBERS IN 'BINARY' BASES.         SLAF0870
C     (IT ALLOWS ROUGHLY THE LAST FOUR BITS OF THE FLOATING-POINT         SLAF0880
C     FRACTION TO BE AFFECTED BY 'REPRESENTATION ERROR'.)                 SLAF0890
C                                                                         SLAF0900
C                                                                         SLAF0910
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAF0920
C                                                                         SLAF0930
C                                                                         SLAF0940
CDP   DOUBLE PRECISION DY, EPS, ONE, SCALE, UNIT. X (N). ZERO             SLAF0950
C                                                                         SLAF0960
CDP   IABS (DY) = IDABS (DY)                                              SLAF0970
CDP   INT (DY) = IDINT (DY)                                               SLAF0980
C                                                                         SLAF0990
C     :::::::: CONSTANT INITIALIZATION  ::::::::                          SLAF1000
      ONE = 1.0E0                                                         SLAF1010
      ZERO = 0.0E0                                                        SLAF1020
C     ::::::::  SET EPS, MACHINE-DEPENDENT TOLERANCE  ::::::::            SLAF1030
      EPS = 1.0E-6                                                        SLAF1040
C                                                                         SLAF1050
C     ::::::::                                                            SLAF1060
C                                                                         SLAF1070
C        THE FIRST LINE (KS1) AND THE FIRST LEAF (KL1) ARE                SLAF1080
C        DETERMINED.                                                      SLAF1090
C                                                                         SLAF1100
C     ::::::::                                                            SLAF1110
C                                                                         SLAF1120
      IF ( X (ILOW)  .NE.  ZERO )  GO TO 10                               SLAF1130
         KSF = 0                                                          SLAF1140
         KS = 0                                                           SLAF1150
         KS1 = 0                                                          SLAF1160
         KL1 = 0                                                          SLAF1170
         LAB = 0                                                          SLAF1180
         GO TO 20                                                         SLAF1190
C     ::::::::  STEM CALCULATION  ::::::::                                SLAF1200
   10 KSF = INT ((X (ILOW) * (ONE + EPS)) / SCALE) * IS                   SLAF1210
      KS = KSF                                                            SLAF1220
      KS1 = KSF / 10                                                      SLAF1230
C     ::::::::  LEAF CALCULATION  ::::::::                                SLAF1240
      KL1 = IABS (INT ((X (ILOW) * (ONE + EPS)) / UNIT) - KS1 * 10)       SLAF1250
C     ::::::::  DETERMINE 'LABEL' FOR THE LINE  ::::::::                  SLAF1260
```

```
      LAB = IABS (KSF - KS1 * 10)
C     :::::::::  SUBTRACT 1 TO ALLOW FOR -0 STEM  ::::::::
      IF ( X (ILOW) .GE. ZERO )  GO TO 20
         KS1 = KS1 - 1
         KS = KS - 10
C
   20 ISTEMS (1) = KS1
      LEAVES (1)  = KL1
      LABELS (1) = LAB
      ILFCNT (1) = 1
C
C     ::::::::
C
C        THE STEMS AND THE LABELS ARE DETERMINED FOR THE REMAINING
C        ENTRIES IN THE ARRAYS ISTEMS AND LABELS, BY INCREMENTING
C        THE FIRST STEM AND LABEL.
C
C     ::::::::
C
      DO  30  I = 2, NSTEMS
         KS = KS +  IS
         ISTEMS (I) = KS / 10
         LABELS (I) = IABS (KS - ISTEMS (I) * 10)
         ILFCNT (I) = 0
C        :::::::::  ADJUST KS TO ALIGN LABELS OF POSITIVE AND
C                   NEGATIVE STEMS  ::::::::
         IF ( KS  .EQ.  (-10) )  KS = -IS
   30 CONTINUE
C
C     ::::::::
C
C        THE NEXT SECTION CALCULATES THE REMAINING LEAVES AND
C        PLACES THEM ON THE CORRECT LINE OF THE DISPLAY BY
C        ADJUSTING ILFCNT.
C
C     ::::::::
C
      N1 = ILOW + 1
C
      DO  60  I = N1, IHI
         IF ( X (I)  .NE.  ZERO )  GO TO 40
            KS = 0
```

```
      KL = 0
      KSL = 0
      GO TO 50
C     ::::::::  STEM CALCULATION  ::::::::
   40 KSL = INT ((X (I) * (ONE + EPS)) / SCALE) * IS
      KS = KSL / 10
C     ::::::::  LEAF CALCULATION  ::::::::
      KL = IABS (INT ((X (I) * (ONE + EPS)) / UNIT) - KS * 10)
      IF ( X (I) .LT. ZERO )  KS = KS - 1
C
   50 LEAVES (I - ILOW + 1) = KL
C
C     ::::::::  PLACE LEAVES ON LINES  ::::::::
C
      J = 1 + ((KSL - KSF) / IS)
      IF ( KS1 .LT. 0  .AND.  KS .GE. 0 )  J = J + 1
      ILFCNT (J) = ILFCNT (J) + 1
   60 CONTINUE
C
      RETURN
C     ::::::::  LAST LINE OF SLLEAF  ::::::::
      END
```

```
      SUBROUTINE  SLPRNT (ILFCNT, ILOW, IOUNIT, ISTEMS, LABELS, LEAVES, SLAH0010
     *   LINWID, N, NN, NSTEMS, UNIT)                                   SLAH0020
C     *****PARAMETERS:                                                  SLAH0030
      INTEGER N, NN, NSTEMS, ILFCNT (NSTEMS), IOUNIT, ISTEMS (NSTEMS),  SLAH0040
     *   LABELS (NSTEMS), LEAVES (NN), LINWID                           SLAH0050
      REAL  UNIT                                                        SLAH0060
C     *****LOCAL VARIABLES:                                             SLAH0070
      INTEGER I, ID, IDEPTH, ISTEM, J, KLEND, KL1, KL2, KS, LEWID       SLAH0080
C     *****FUNCTIONS:                                                   SLAH0090
      INTEGER  IABS, MINO                                               SLAH0100
C                                                                       SLAH0110
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAH0120
C                                                                       SLAH0130
C                                                                       SLAH0140
C     *****PURPOSE:                                                     SLAH0150
C     THIS SUBROUTINE PRINTS A STEM-AND-LEAF DISPLAY, PRECEDED BY THE   SLAH0160
C     APPROPRIATE UNIT FOR THE DISPLAY.                                 SLAH0170
C                                                                       SLAH0180
C     *****PARAMETER DESCRIPTION:                                       SLAH0190
C                                                                       SLAH0200
C     ON INPUT:                                                         SLAH0210
C                                                                       SLAH0220
C        ILFCNT IS AN ARRAY OF LENGTH NSTEMS.  IT CONTAINS THE NUMBER   SLAH0230
C             OF LEAVES ON EACH LINE OF THE DISPLAY.  ILFCNT IS USED    SLAH0240
C             BY THE SUBROUTINE IN DETERMINING WHICH LEAVES             SLAH0250
C             ARE TO BE PLACED ON EACH LINE.                            SLAH0260
C                                                                       SLAH0270
C        ILOW IS THE POSITION IN THE SORTED BATCH OF THE FIRST ELEMENT  SLAH0280
C             TO BE PRINTED IN THE DISPLAY.                             SLAH0290
C                                                                       SLAH0300
C        IOUNIT CONTAINS THE DATA SET REFERENCE NUMBER FOR THE WRITE    SLAH0310
C             STATEMENTS.  IOUNIT IS INITIALLY SET TO '6' BY THE        SLAH0320
C             SUBROUTINE SLDSPY.                                        SLAH0330
C                                                                       SLAH0340
C        ISTEMS IS AN ARRAY OF LENGTH NSTEMS, CONTAINING THE 'STEM'     SLAH0350
C             PART OF EACH LINE IN THE DISPLAY.                         SLAH0360
C                                                                       SLAH0370
C        LABELS IS ALSO AN ARRAY OF LENGTH NSTEMS.  STORED IN           SLAH0380
C             LABELS ARE THE VALUES 0, 2, 4, 5, 6, OR 8 WHICH ARE       SLAH0390
C             USED BY SLPRNT TO DETERMINE THE 'TAG' (T, F, S, OR .)     SLAH0400
C             TO BE PLACED ON EACH LINE.                                SLAH0410
C                                                                       SLAH0420
C        LEAVES IS AN ARRAY OF LENGTH NN CONTAINING THE 'LEAF'
```

```
C             DIGIT FOR EACH DATA VALUE IN THE DISPLAY.
C
C        LINWID MUST BE SET TO THE PAGE WIDTH OF THE PRINTED OUTPUT.
C
C        N MUST BE SET TO THE NUMBER OF DATA VALUES IN THE BATCH.
C
C        NN MUST BE SET TO THE NUMBER OF ELEMENTS IN THE DISPLAY.
C             NN IS CALCULATED IN SLOSPY.
C
C        NSTEMS CONTAINS THE NUMBER OF LINES FOR THE DISPLAY, AS
C             COMPUTED IN THE SUBROUTINE SLSCAL.
C
C        UNIT CONTAINS THE APPROPRIATE POWER OF 10 FOR THE DISPLAY.
C             UNIT IS COMPUTED IN SLSCAL.
C
C
C     *****APPLICATION AND USAGE RESTRICTIONS:
C     SLPRNT IS CALLED BY THE SUBROUTINE SLOSPY.  IT USES THE OUTPUT
C     OF SLSCAL (UNIT, AND NSTEMS) AND THE STEMS, LEAVES, AND
C     LABELS PRODUCED BY SLLEAF TO PRINT THE STEM-AND-LEAF DISPLAY.
C
C     *****ALGORITHM NOTES:
C     LINES IN THE SUBROUTINE CONTAINING 'CDP' IN THE FIRST THREE
C     COLUMNS WILL HELP THE USER IN CONVERTING THE ROUTINE FROM SINGLE
C     TO DOUBLE-PRECISION. 'REAL' DEFINITION CARDS MUST BE REMOVED.
C
C
C     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
CDP   DOUBLE PRECISION DY, UNIT
C
CDP   IABS (DY) = IDABS (DY)
C
      WRITE (IOUNIT,400)  UNIT
      KL2 = 0
      KLEND = 0
      IDEPTH = ILOW - 1
C     ::::::::  DETERMINE LEWID, THE MAXIMUM NUMBER OF LEAVES FOR THE
C                 PRINTED PAGE  ::::::::
      LEWID = LINWID - 23
C
C     ::::::::  THE DO LOOP FOR THE DISPLAY  ::::::::
```

```
C                                                                         SLAH0850
      DO  50  I = 1, NSTEMS                                               SLAH0860
C         ::::::::  ISTEM IS THE CURRENT STEM  ::::::::                   SLAH0870
          ISTEM = ISTEMS (I)                                              SLAH0880
C         ::::::::  KS SPECIFIES THE CURRENT LINE IN THE CYCLE  :::::::: SLAH0890
          KS = LABELS (I)                                                 SLAH0900
C         ::::::::  ID IS THE DEPTH OF THE CURRENT LINE  ::::::::         SLAH0910
          ID = MINO (IDEPTH + ILFCNT (I), N - IDEPTH)                     SLAH0920
          IF (IABS (N - (IDEPTH + ILFCNT (I)) - IDEPTH) .LT. ILFCNT (I)) SLAH0930
     *        ID = ILFCNT (I)                                             SLAH0940
          IF ( ILFCNT (I) .EQ. 0 )  GO TO 20                              SLAH0950
          KL1 = KL2 + 1                                                   SLAH0960
          KL2 = KL2 + ILFCNT (I)                                          SLAH0970
C         ::::::::  CHECK THAT LEAF COUNT IS NOT GT LINE WIDTH  :::::::: SLAH0980
          KLEND = KL2                                                     SLAH0990
          IF ( ILFCNT (I) .GT. LFWID )  KL2 = KL1 + LFWID - 1             SLAH1000
          IF ( ISTEM .EQ. (-1) )  GO TO 10                                SLAH1010
          IF ( ISTEM .LT. 0 )  ISTEM = ISTEM + 1                          SLAH1020
C                                                                         SLAH1030
C         ::::::::  PRINTING OF LINES WITH LEAVES ON THEM  ::::::::       SLAH1040
C                                                                         SLAH1050
          IF ( KS .EQ. 0 )  WRITE (IOUNIT,500)  ID, ISTEM,                SLAH1060
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1070
          IF ( KS .EQ. 5  .OR.  KS .EQ. 8 )  WRITE (IOUNIT,505)  ID,      SLAH1080
     *          ISTEM,  (LEAVES (J), J = KL1, KL2)                        SLAH1090
          IF ( KS .EQ. 2 )  WRITE (IOUNIT,502) ID,                        SLAH1100
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1110
          IF ( KS .EQ. 4 )  WRITE (IOUNIT,503) ID,                        SLAH1120
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1130
          IF ( KS .EQ. 6 )  WRITE (IOUNIT,504) ID,                        SLAH1140
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1150
          GO TO 40 .                                                      SLAH1160
C                                                                         SLAH1170
C         ::::::::  SPECIAL ARRANGEMENTS FOR THE -0 STEM  ::::::::        SLAH1180
C                                                                         SLAH1190
   10     IF ( KS .EQ. 0 )  WRITE (IOUNIT,510) ID,                        SLAH1200
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1210
          IF ( KS .EQ. 5  .OR.  KS .EQ. 8 )  WRITE (IOUNIT,515)  ID,      SLAH1220
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1230
          IF ( KS .EQ. 2 )  WRITE (IOUNIT,512) ID,                        SLAH1240
     *          (LEAVES (J), J = KL1, KL2)                                SLAH1250
          IF ( KS .EQ. 4 )  WRITE (IOUNIT,513) ID,                        SLAH1260
```

```
     *        (LEAVES (J), J = KL1, KL2)
           IF ( KS .EQ. 6 )  WRITE (IOUNIT,514) ID,
     *        (LEAVES (J), J = KL1, KL2)
           GO TO 40
C
C          ::::::::  PRINTING OF LINES WITH NO LEAVES ON THEM  ::::::::
C
   20      IF ( ISTEM .EQ. (-1) )  GO TO 30
           IF ( ISTEM .LT. 0)  ISTEM = ISTEM + 1
           IF ( KS .EQ. 0 )  WRITE (IOUNIT,500)  ID, ISTEM
           IF ( KS .EQ. 5 .OR. KS .EQ. 8 )  WRITE (IOUNIT,505)  ID, ISTEM
           IF ( KS .EQ. 2 )  WRITE (IOUNIT,502)  ID
           IF ( KS .EQ. 4 )  WRITE (IOUNIT,503)  ID
           IF ( KS .EQ. 6 )  WRITE (IOUNIT,504)  ID
           GO TO 40
C
C          ::::::::  PRINTING THE -0 STEM IF IT HAS NO LEAVES  ::::::::
C
   30      IF ( KS .EQ. 0 )  WRITE (IOUNIT,510)  ID
           IF ( KS .EQ. 5  .OR.  KS .EQ. 8 )  WRITE (IOUNIT,515)  ID
           IF ( KS .EQ. 2 )  WRITE (IOUNIT,512)  ID
           IF ( KS .EQ. 4 )  WRITE (IOUNIT,513)  ID
           IF ( KS .EQ. 6 )  WRITE (IOUNIT,514)  ID
C
C       ::::::::  PRINT THE LINE OVERFLOW  ::::::::
   40 IF ( KLEND .EQ. KL2 )  GO TO 50
           KL1 = KL2 + 1
           KL2 = KL2 + MIN0 (LEWID, (KLEND - KL1 + 1))
           WRITE (IOUNIT,600)  (LEAVES (J), J = KL1, KL2)
           GO TO 40
C
C       ::::::::  UPDATE IDEPTH  ::::::::
   50 IDEPTH = IDEPTH + ILECNT (I)
C
        RETURN
C
400     FORMAT (1X / 15X, 9H( UNIT = , E12.4, 2H ) /)
500     FORMAT (1X, I5, 5X, I9, 4H  I , 100I1)
502     FORMAT (1X, I5, 14X, 4HT I , 100I1)
503     FORMAT (1X, I5, 14X, 4HF I , 100I1)
504     FORMAT (1X, I5, 14X, 4HS I , 100I1)
505     FORMAT (1X, I5, 5X, I9, 4H. I , 100I1)
```

```
C                                                                       SLAH1690
510    FORMAT (1X, I5, 12X, 6H-O   I , 100I1)                           SLAH1700
512    FORMAT (1X, I5, 12X, 6H   T I , 100I1)                           SLAH1710
513    FORMAT (1X, I5, 12X, 6H   F I , 100I1)                           SLAH1720
514    FORMAT (1X, I5, 12X, 6H   S I , 100I1)                           SLAH1730
515    FORMAT (1X, I5, 12X, 6H-O. I , 100I1)                            SLAH1740
C                                                                       SLAH1750
600    FORMAT (24X, 100I1)                                              SLAH1760
C                                                                       SLAH1770
C      ::::::::  LAST LINE OF SLPRNT  ::::::::                          SLAH1780
       END                                                              SLAH1790
```

```
      SUBROUTINE  SLSORT (V1, N)                                        SLAJ0010
C     *****PARAMETERS:                                                  SLAJ0020
      INTEGER  N                                                        SLAJ0030
      REAL  V1 (N)                                                      SLAJ0040
C     *****LOCAL VARIABLES:                                             SLAJ0050
      INTEGER  I, J, K, L, M                                            SLAJ0060
      REAL  X                                                           SLAJ0070
C     *****FUNCTIONS:                                                   SLAJ0080
C     NONE                                                              SLAJ0090
C                                                                       SLAJ0100
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAJ0110
C                                                                       SLAJ0120
C     *****PURPOSE:                                                     SLAJ0130
C     THE SUBROUTINE PERFORMS A SHELL SORT.  THIS ALGORITHM IS          SLAJ0140
C     TAKEN FROM (1).                                                   SLAJ0150
C                                                                       SLAJ0160
C     *****PARAMETER DESCRIPTION:                                       SLAJ0170
C                                                                       SLAJ0180
C     ON INPUT:                                                         SLAJ0190
C                                                                       SLAJ0200
C        V1 CONTAINS THE ELEMENTS OF THE BATCH TO BE SORTED. AS         SLAJ0210
C             THIS ALGORITHM SORTS DESTRUCTIVELY, THE ORIGINAL ORDER    SLAJ0220
C             OF THE ELEMENTS IS LOST.                                  SLAJ0230
C                                                                       SLAJ0240
C        N MUST BE SET TO THE NUMBER OF ELEMENTS IN V1.                 SLAJ0250
C                                                                       SLAJ0260
C     ON OUTPUT:                                                        SLAJ0270
C                                                                       SLAJ0280
C        V1 CONTAINS THE SORTED ELEMENTS OF THE BATCH.                  SLAJ0290
C                                                                       SLAJ0300
C     *****APPLICATION AND USAGE RESTRICTIONS:                          SLAJ0310
C     SLSORT IS CALLED FROM THE SUBROUTINE SLDSPY.  THE SORTING         SLAJ0320
C     PERFORMED HERE IS NECESSARY IN DETERMINING THE STEM-AND-LEAF      SLAJ0330
C     DISPLAY.                                                          SLAJ0340
C                                                                       SLAJ0350
C     *****ALGORITHM NOTES:                                             SLAJ0360
C     LINES IN THE SUBROUTINE CONTAINING 'COP' IN THE FIRST THREE       SLAJ0370
C     COLUMNS WILL HELP THE USER IN CONVERTING THE ROUTINE FROM SINGLE  SLAJ0380
C     TO DOUBLE-PRECISION. 'REAL' DEFINITION CARDS MUST BE REMOVED.     SLAJ0390
C                                                                       SLAJ0400
C     *****REFERENCES:                                                  SLAJ0410
C     (1)    BOOTHROYD, J.,  ALGORITHM 201, COMMUNICATIONS OF THE ACM,  SLAJ0420
```

```
C          VOLUME 6  (1963), PAGE 445.                                   SLAJ0430
C                                                                        SLAJ0440
C     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAJ0450
C                                                                        SLAJ0460
CDP   DOUBLE PRECISION  V1 (N), X                                        SLAJ0470
C                                                                        SLAJ0480
      I = 1                                                              SLAJ0490
    1 I = I + I                                                          SLAJ0500
      IF ( I .LE. N )  GO TO 1                                           SLAJ0510
      M = I - 1                                                          SLAJ0520
    2 M = M / 2                                                          SLAJ0530
      IF ( M .EQ. 0 )   RETURN                                           SLAJ0540
      K = N - M                                                          SLAJ0550
C                                                                        SLAJ0560
      DO  4  J = 1, K                                                    SLAJ0570
         L = J                                                           SLAJ0580
    3    IF ( L .LT. 1 )  GO TO 4                                        SLAJ0590
         IF ( V1 (L + M) .GE. V1 (L) )   GO TO 4                         SLAJ0600
         X = V1 (L + M)                                                  SLAJ0610
         V1 (L + M) = V1 (L)                                             SLAJ0620
         V1 (L) = X                                                      SLAJ0630
         L = L - M                                                       SLAJ0640
         GO TO 3                                                         SLAJ0650
    4 CONTINUE                                                           SLAJ0660
C                                                                        SLAJ0670
      GO TO 2                                                            SLAJ0680
C     ::::::::  LAST LINE OF SLSORT  ::::::::                            SLAJ0690
      END                                                                SLAJ0700
```

```
      FUNCTION IFLOOR (Y)                                               SLAL0010
C     *****PARAMETER:                                                   SLAL0020
      REAL  Y                                                           SLAL0030
C     *****LOCAL VARIABLES:                                             SLAL0040
      INTEGER  I                                                        SLAL0050
      REAL  ZERO                                                        SLAL0060
C     *****FUNCTIONS:                                                   SLAL0070
      INTEGER  INT                                                      SLAL0080
      REAL  FLOAT                                                       SLAL0090
C                                                                       SLAL0100
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAL0110
C                                                                       SLAL0120
C     *****PURPOSE:                                                     SLAL0130
C     THIS FUNCTION SIMULATES THE FLOOR FUNCTION.                       SLAL0140
C                                                                       SLAL0150
C     *****PARAMETER DESCRIPTION:                                       SLAL0160
C                                                                       SLAL0170
C     ON INPUT:                                                         SLAL0180
C                                                                       SLAL0190
C        Y CONTAINS A REAL NUMBER.  IFLOOR RETURNS THE LARGEST INTEGER  SLAL0200
C             LESS THAN OR EQUAL TO Y.                                  SLAL0210
C                                                                       SLAL0220
C     *****APPLICATION AND USAGE RESTRICTIONS:                          SLAL0230
C     NONE                                                              SLAL0240
C                                                                       SLAL0250
C     *****ALGORITHM NOTES:                                             SLAL0260
C     LINES IN THE FUNCTION CONTAINING 'CDP' IN THE FIRST THREE         SLAL0270
C     COLUMNS WILL HELP THE USER IN CONVERTING THE FUNCTION FROM SINGLE SLAL0280
C     TO DOUBLE-PRECISION. 'REAL' DEFINITION CARDS MUST BE REMOVED.     SLAL0290
C                                                                       SLAL0300
C     IN ADDITION, ALL CONSTANTS INITIALIZED IN THE SUBROUTINE          SLAL0310
C     SHOULD BE CHANGED FROM 'E' TO 'D' FOR DOUBLE-PRECISION            SLAL0320
C     CONVERSION.                                                       SLAL0330
C                                                                       SLAL0340
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::SLAL0350
C                                                                       SLAL0360
CDP   DOUBLE PRECISION DY, Y, ZERO                                      SLAL0370
CDP   DOUBLE PRECISION FLOAT                                            SLAL0380
C                                                                       SLAL0390
CDP   INT (DY) = IDINT (DY)                                             SLAL0400
CDP   FLOAT (I) = DFLOAT (I)                                            SLAL0410
C                                                                       SLAL0420
```

```
C     ::::::::  CONSTANT INITIALIZATION  ::::::::
      ZERO = 0.0E0
C
      IFLOOR = INT (Y)
      IF ( Y .LT. ZERO  .AND.  Y .NE. FLOAT (IFLOOR) )
     *   IFLOOR = IFLOOR - 1
      RETURN
C     ::::::::   LAST LINE OF FUNCTION IFLOOR  ::::::::
      END
```