Review of MLAB: A Mathematical Modeling Laboratory

Sidney H. Young

Department of Chemistry, University of South Alabama, Mobile, Alabama 36688

Received March 3, 1995

INTRODUCTION

MLAB is a numeric mathematics program useful for laboratories and classrooms in which a large amount of numerical analysis must be performed. The software allows the development of mathematical models (a collection of mathematical functions or differential equations) which can be used for evaluating experimental data or for observing predictions involving simulated data which may be modified at will.

The software was originally developed by Gary Knott and Douglas Reece at the National Institute of Health and is now available on IBM-PCs, X-Windows, Macintosh II computers, and other machines as well as from Civilized Software, 7735 Old Georgetown Rd., Suite 410, Bethesda, MD 20814. The current price of the DOS version is $1495.

HARDWARE

The IBM-PC version used by the reviewer required an Intel 80386 or 80486 CPU chip and a 80387 or 80487 math coprocessor chip. (An 80486DX is also "math coprocessor ready".) The MLAB installation diskettes required a 3.5" 1.44M drive. At least 3Mbytes of space was needed on the hard disk. The monitor system must have EGA or VGA hardware-compatible graphics and associated monitor (colored or monochrome). It needs up to 640K of user memory to operate efficiently.

FEATURES

The program is written as a group of procedures which performs tasks involving data manipulation and mathematical calculations. Included in the program are features to perform the following mathematical operations (plus many more):

1. Curve fitting to data—This algorithm uses the Levenberg–Marquardt algorithm to fit a set of data to a function (linear or nonlinear) or a first-order differential equation and generates useful statistical analyses (such as residuals) to assist in interpretation of data. This algorithm can be used for simple straight-line fits as well.

2. Solutions of differential equations—The INTEGRATE function will give numerical solutions to a series of coupled linear first-order differential equations using the Gear or Adams methods, which are presently the recommended methods for solving a series of "stiff" differential equations.

3. Graphics—The program will produce plots either on the screen or from a plotter (PostScript or LaserJet II only!). The pictures can be pulled up at will. The program will adjust the number of pixels to fill the viewing screen at installation time, so very little black border remains at the screen. Multidimensional graphs are also available.

4. A very large number of built-in mathematical, statistical, probability, and signal-processing functions are available as well as vector and matrix functions. For advanced statistics, cluster and survival analysis functions have also been included.

5. The ability to write programs involving some of the above routines is facilitated by the addition of looping and "if-then-else" routines.

Most of the above procedures have practical applications in both laboratory and academic settings.

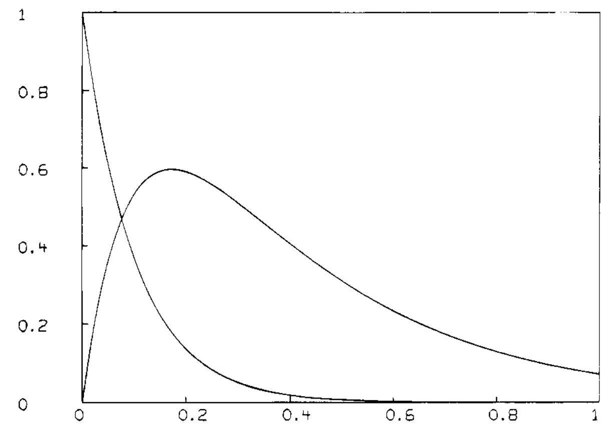

Figure 1. MLAB solution to the intermediate problem.

A COMPARATIVE EXAMPLE

To illustrate the difference between MLAB and other numerical programs, the following problem was posed for solution by both MLAB and MATHCAD.

Suppose you wanted to graph the concentration of the chemical species B as a function of time, where B is an intermediate in a process A → B → C, for various values of k1 and k2, the rate constants. The following simple procedure in MLAB will perform the calculations and prepare the graph, with a time range of from 0 to 1 s and with k1=10 and k2=3, for the case where there is only initially A.

- FCT A'T(T)=-K1*A
- FCT B'T(T)=K1*A-K2*B
- K1=10
- K2=3
- INIT A(0)=1
- INIT B(0)=0
- TIME=0:1!101
- M=INTEGRATE(A'T,B'T,TIME)
- DRAW M COL(1,2)
- DRAW M COL(1,4)
- VIEW

Two consecutive reactions.

$$K1 := 10 \qquad K2 := 3$$

$$F1(x1, x2, t) := -K1 \cdot x1$$

$$F2(x1, x2, t) := K1 \cdot x1 - K2 \cdot x2$$

$$h := 0.010$$

$$x1_0 := 1$$

$$x2_0 := 0$$

$$t_0 := 0$$

$$k11_0 := h \cdot F1(x1_0, x2_0, t_0) \qquad k12_0 := h \cdot F2(x1_0, x2_0, t_0)$$

$$k21_0 := h \cdot F1\left(x1_0 + \frac{k11_0}{2}, x2_0 + \frac{k12_0}{2}, t_0 + \frac{h}{2}\right)$$

$$k22_0 := h \cdot F2\left(x1_0 + \frac{k11_0}{2}, x2_0 + \frac{k12_0}{2}, t_0 + \frac{h}{2}\right)$$

$$k31_0 := h \cdot F1\left(x1_0 + \frac{k21_0}{2}, x2_0 + \frac{k22_0}{2}, t_0 + \frac{h}{2}\right)$$

$$k32_0 := h \cdot F2\left(x1_0 + \frac{k21_0}{2}, x2_0 + \frac{k22_0}{2}, t_0 + \frac{h}{2}\right)$$

$$k41_0 := h \cdot F1(x1_0 + k31_0, x2_0 + k32_0, t_0 + h)$$

$$k42_0 := h \cdot F2(x1_0 + k31_0, x2_0 + k32_0, t_0 + h)$$

$$i := 1 .. 100$$

$$\begin{bmatrix} x1_i \\ x2_i \\ t_i \\ k11_i \\ k12_i \\ k21_i \\ k22_i \\ k31_i \\ k32_i \\ k41_i \\ k42_i \end{bmatrix} := \begin{bmatrix} x1_{i-1} + \frac{k11_{i-1}}{6} + \frac{k21_{i-1}}{3} + \frac{k31_{i-1}}{3} + \frac{k41_{i-1}}{6} \\ x2_{i-1} + \frac{k12_{i-1}}{6} + \frac{k22_{i-1}}{3} + \frac{k32_{i-1}}{3} + \frac{k42_{i-1}}{6} \\ t_{i-1} + h \\ h \cdot F1(x1_{i-1}, x2_{i-1}, t_{i-1}) \\ h \cdot F2(x1_{i-1}, x2_{i-1}, t_{i-1}) \\ h \cdot F1\left(x1_{i-1} + \frac{k11_{i-1}}{2}, x2_{i-1} + \frac{k12_{i-1}}{2}, t_{i-1} + \frac{h}{2}\right) \\ h \cdot F2\left(x1_{i-1} + \frac{k11_{i-1}}{2}, x2_{i-1} + \frac{k12_{i-1}}{2}, t_{i-1} + \frac{h}{2}\right) \\ h \cdot F1\left(x1_{i-1} + \frac{k21_{i-1}}{2}, x2_{i-1} + \frac{k22_{i-1}}{2}, t_{i-1} + \frac{h}{2}\right) \\ h \cdot F2\left(x1_{i-1} + \frac{k21_{i-1}}{2}, x2_{i-1} + \frac{k22_{i-1}}{2}, t_{i-1} + \frac{h}{2}\right) \\ h \cdot F1(x1_{i-1} + k31_{i-1}, x2_{i-1} + k32_{i-1}, t_{i-1} + h) \\ h \cdot F2(x1_{i-1} + k31_{i-1}, x2_{i-1} + k32_{i-1}, t_{i-1} + h) \end{bmatrix}$$

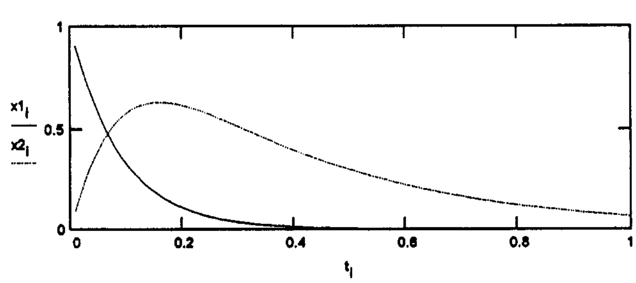

Figure 2. MATHCAD solution to the intermediate problem.

That is the entire program! A sophomore could easily use this simple language to perform complex procedures. Figure 1 gives the output plot of MLAB, printed on a HP-LaserJet II printer.

The same problem was solved by MATHCAD, and the results are given in Figure 2. Note that in terms of time and effort, the MLAB procedure was faster (the entire calculation took less than 2 s of computer time), and the routine was simpler than the MATHCAD procedure, which took over 1 min to run.

AUDIENCE

There are two groups of users for which this software could be very valuable. One group would be freshman or sophomore students in a lower-level engineering or science course, in which the results of the calculation or simulation is more important than presentation or study of the actual algorithm used. A second group of users would be students in advanced science, engineering or mathematics courses, or laboratory users, where complex problems are solved with state-of-the-art algorithms.

The software is probably not useful in situations in which the algorithm used is to be taught. For example, the actual algorithm used for the "INTEGRATE" command is hidden from the user, so it could not be taught directly from the software. In this situation, MATHCAD might be more useful as a teaching tool. There is also little ability to compare various algorithms in terms of ability or limitations of the algorithm; for example, there might be interest in comparison between the Runge–Kutta and the Gear methods for solution of first-order differential equations. But MLAB does not have a built-in function for the Runge–Kutta method, and there is no easy way to encode the algorithm with the available MLAB functions.

CONCLUSION

MLAB is a strong program useful for many mathematics applications to science and engineering. The language is simple to master, and with the assistance of a good reference manual and a series of well-written examples, the user can incorporate the software into whatever analysis is needed quickly and efficiently.

CI950348R