

# A Consistently Well-Behaved Method of Interpolation

Russell W. Stineman

*We're not an academic journal, but once in a while something serious and original comes in. This is apparently a real solution to a problem graphics guys know well; this paper gives a smooth slope-matching interpolating curve, without ever going wild.*

## Abstract

When a curve is represented in a digital computer by a table of points, many methods of interpolation have difficulty near an abrupt change in the slope of the original curve. The problem is evidenced by the interpolating curve having more inflection points than the actual function that it is intended to approximate.

## Polynomial interpolation often gives wild results near an abrupt change of slope.

This paper presents a method of interpolation which generates a curve that will never have more inflection points than are clearly required by the given set of points. The interpolating curve passes through the tabulated points and exactly matches the given slopes at those points (except for one unlikely degenerate case which has a slope discontinuity at one of the given points).

When used to approximate a sine function, the method of this paper was found to be more accurate than spline interpolation. The amount of computation required to find an interpolated point is approximately the same as to evaluate a sixth-degree polynomial.

An appendix presents a suggested way to compute slopes at the given points when only the points are known. However, exact slopes should be used whenever possible.

## Introduction

In digital computer computations, it is common to represent a curve by a set of points, with interpolation between points. However, as is well

known, polynomial interpolation often gives wild results near an abrupt change of slope. Spline interpolation can also give unreasonable results.

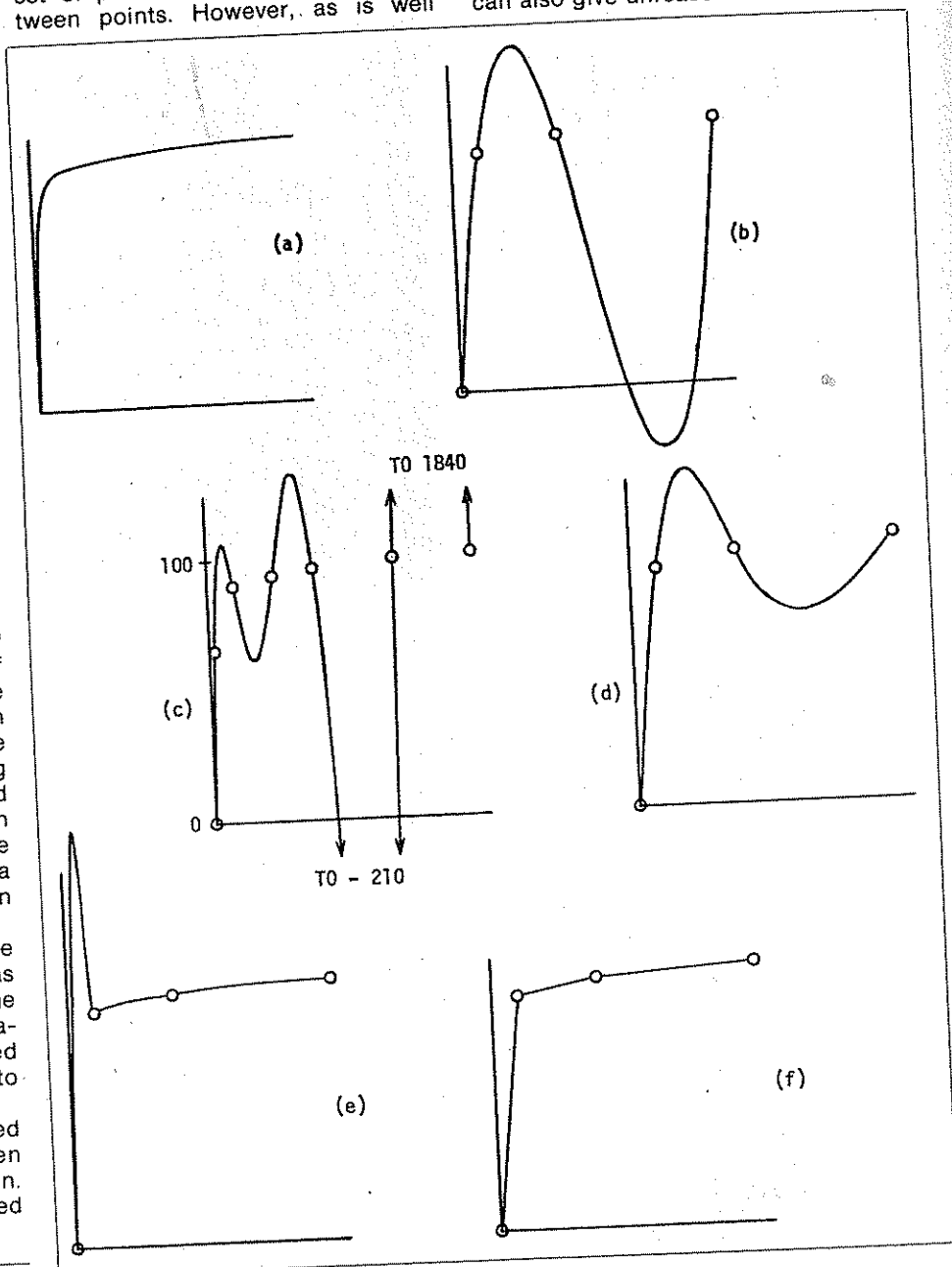


Figure 1

The problem is illustrated in Figure 1. Curve (a) shows a monotonically increasing curve with an abrupt decrease in slope. In (b), four points on curve (a) have been fitted by a cubic polynomial, with completely unsatisfactory results. In (c), three additional points have been taken from (a) and all seven points have been fitted by a 6th-degree polynomial, still with unsatisfactory results. In (d), the same four points as in (b) have been fitted by three piecewise cubic polynomials, chosen to preserve continuity of first and second derivatives at the interior points ("spline" interpolation), again unsatisfactory. Piecewise cubics are also used in (e), but the slopes at the given points are made equal to the slopes in (a), also unsatisfactory. In desperation, many analysts have used linear interpolation as in (f), accepting the need for a relatively large number of points to achieve a given accuracy.

**The complete assurance that the procedure will never generate "wild" points makes it attractive as a general-purpose procedure.**

What is needed is an interpolation procedure with the following properties:

a. If values of the ordinates of the specified points change monotonically, and the slopes of the line segments joining the points change monotonically, then the interpolating curve and its slope will change monotonically.

b. If the slopes of the line segments joining the specified points change monotonically, then the slope of the interpolating curve will change monotonically.

c. Suppose that the conditions in (a) or (b) are satisfied by a set of points, but a small change in the ordinate or slope at one of the points will result in conditions (a) or (b) being no longer satisfied. Then making this small change in the ordinate or slope at a point will cause no more than a small change in the interpolating curve.

An interpolation procedure that has the above properties is given in the next section of this article. The last of the three properties is discussed later in the paper in terms of an example.

**Interpolation Procedure**

In the following discussion, it is assumed that  $x_j, y_j, y'_j, j = 1, 2, \dots, n$  are given where

$x_j, y_j$  = rectangular coordinates of  $j$ th point on curve.

$y'_j$  = slope of the curve at  $j$ th point.  
 $x_j < x_{j+1}$  for  $j = 1, 2, \dots, n-1$

If the slopes are not initially known, they may be calculated by the method described in the Appendix. Slopes thus calculated are consistent with achieving the objectives stated in the Introduction.

Given  $x$  such that  $x_j \leq x \leq x_{j+1}$ , the procedure for calculating  $y$  (the corresponding interpolated value) is the following. The slope of the line segment joining the two points is

$$s_j = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} \quad (1)$$

Values of  $s_j$  may be precomputed and stored along with the given points and slopes. On the line segment, the ordinate corresponding to  $x$  is

$$y_0 = y_j + s_j (x - x_j) \quad (2)$$

Next,

$$\Delta y_j = y_j + y'_j (x - x_j) - y_0 \quad (3)$$

where  $\Delta y_j$  is the vertical distance from the point  $(x, y_0)$  to a line through  $(x_j, y_j)$  with slope  $y'_j$ , as shown in Figure 2. Similarly,

$$\Delta y_{j+1} = y_{j+1} + y'_{j+1} (x - x_{j+1}) - y_0 \quad (4)$$

is the vertical distance from the point  $(x, y_0)$  to a line thru  $(x_{j+1}, y_{j+1})$  with slope  $y'_{j+1}$ , also shown in Figure 2. The product  $\Delta y_j \Delta y_{j+1}$  is then calculated and tested.

If  $y'_j = s_j$ , then the line through point  $(x_j, y_j)$  with slope  $y'_j$  will coincide with the line segment joining points  $(x_j, y_j)$  and  $(x_{j+1}, y_{j+1})$ , and  $\Delta y_j = 0$ . Similarly, if  $y'_{j+1} = s_{j+1}$ , then  $\Delta y_{j+1} = 0$ . If either or both  $\Delta y_j$  and  $\Delta y_{j+1}$  are zero, then the product  $\Delta y_j \Delta y_{j+1} = 0$ , and  $y$  is simply

$$y = y_0 \quad (5)$$

If  $\Delta y_j \Delta y_{j+1} \neq 0$ , but  $\Delta y_j$  and  $\Delta y_{j+1}$

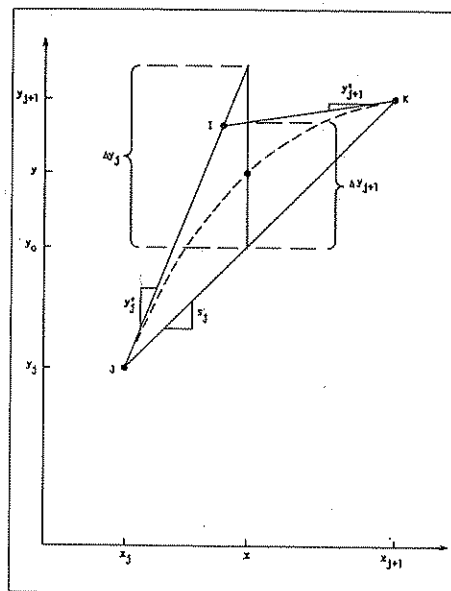


Figure 2

are not both zero, then the interpolating curve will have a slope discontinuity. For example, if  $\Delta y_j \neq 0$ , then the slope for  $x_j < x < x_{j+1}$  will be  $y' = s_j = y'_{j+1}$ . But as  $x \rightarrow x_j$  from the left,  $y' \rightarrow y'_j \neq s_j$ . This degenerate case is the only way a slope discontinuity can occur.

If  $\Delta y_j \Delta y_{j+1} > 0$ , then (as in Figure 2)  $\Delta y_j$  and  $\Delta y_{j+1}$  have the same sign, and

$$y = y_0 + \frac{\Delta y_j \Delta y_{j+1}}{\Delta y_j + \Delta y_{j+1}} \quad (6)$$

Equation (6) always determines the point  $(x, y)$  inside the triangle IJK of Figure 2. The slope of the interpolating curve matches the given slopes at the given points. The slope changes monotonically between the given points, so the interpolating curve is always concave toward the line segment joining the given points.

If  $\Delta y_j \Delta y_{j+1} < 0$ , then the geometry is like Figure 3, and there must be an inflection point between  $x_j$  and  $x_{j+1}$ . In this case,

$$y = y_0 + \frac{\Delta y_j \Delta y_{j+1} (x - x_j + x - x_{j+1})}{(\Delta y_j - \Delta y_{j+1}) (x_{j+1} - x_j)} \quad (7)$$

Equation (7) always determines the point  $(x, y)$  inside the quadrilateral JIKL of Figure 3, where the vertical distance LO equals the vertical distance OI. The slope of the interpolating curve matches the given slopes at the given points. The interpolating curve intersects line segment JK at its midpoint.

The rationale for equation (7) may be understood by considering the case where  $y'_j$  is significantly greater than  $s_j$ , the slope of line segment JK, but  $y'_{j+1}$  is nearly equal to  $s_j$ , Figure 2 or 3. Regardless of whether  $y'_{j+1}$  is greater or less than  $s_j$ , points I and L will be very close to point J and the interpolating curve will be very close to line segment JK. Thus, a change of  $y'_{j+1}$  from slightly more than  $s_j$  to slightly less than  $s_j$  will cause only a slight change in the interpolating curve. This example illustrates the third requirement given in the Introduction.

Equations (6) and (7) fall in the general area of rational interpolation. However, the desirable properties of this method of interpolation stem from the particular form of (6) and (7). In general, rational interpolation does not have such properties.

The curve in Figure 1(a) was calculated by the above method of interpolation, given the four points shown in Figure 1(b), and with slopes calculated by the method given in the Appendix.

**Accuracy**

The accuracy of the given interpolation procedure may be illustrated

## Interpolation, cont'd...

by fitting the function

$$y = \sin x \quad (8)$$

No attempt is made to get an optimum fit. Rather, the values chosen for  $x_j$  are 0, 45, and 90 degrees, and the corresponding values of  $y_j$  and  $y'_j$  are computed exactly using equation (8). The resulting interpolated curve deviates from  $\sin x$  by a maximum of .00333, at  $x = 24$  degrees.

By contrast, the maximum error using linear interpolation is .0704, at 68 degrees. From another standpoint, to achieve a maximum error of no more than .00333, linear interpolation requires that  $x$  be given at intervals of 9 degrees.

An example given in reference [1] considers one full cycle of the function given in equation (8). The points  $x_j$  are selected at 45-degree intervals (that is,  $x_j = 0, 45, 90, \dots, 360$  degrees), and the values of  $y_j$  are calculated exactly by (8). Interpolation is then done with a fifth-degree spline. That is, piecewise fifth-degree polynomials are found such that the interpolating curve and its first four derivatives are continuous. The interpolating curve also exactly matches the first and third derivatives of equation (8) at  $x = 0$  and at  $x = 360$  degrees. The interpolating curve deviates from  $\sin x$  by a maximum of .0372, at 25 and at 335 degrees.

The same 9 points as in the above example from reference (1) were fitted by the method of this paper, using slopes calculated by the method given in the Appendix, rather than exact slopes. In this case, the interpolating curve deviates from  $\sin x$  by a maximum of .0766, at 18 and at 342 degrees. Using exact slopes, the maximum error is .00333, at 24, 156, 204, and 336 degrees. This shows the importance of using accurate slopes at the given

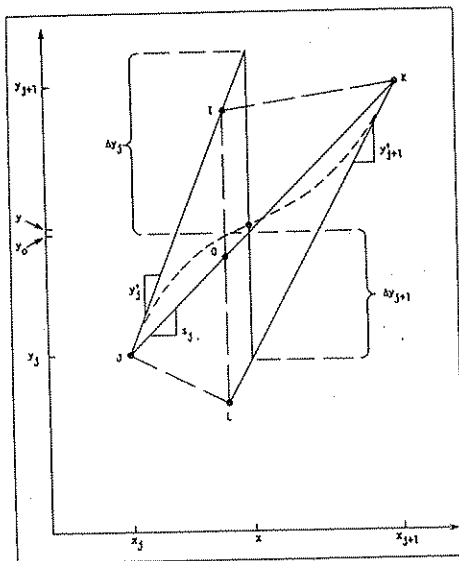


Figure 3

points, if the slopes are known.

## Conclusions

A procedure has been presented for interpolating between tabulated points. This procedure completely avoids the problems which various forms of polynomial interpolation, including spline interpolation, have near an abrupt change of slope. The procedure is especially recommended for such applications as the voltage-current curve of a semiconductor. However, the complete assurance that the procedure will never generate "wild" points makes it attractive as a general-purpose procedure.

The procedure uses only ordinary arithmetic operations (that is, no

**In desperation, many analysts have used linear interpolation, accepting the need for a relatively large number of points to achieve a given accuracy.**

trigonometric, exponential, or similar functions need be evaluated. The number of operations is approximately equivalent to the evaluation of a sixth-degree polynomial. Memory required for data is four words per point.

When fitting a known function, the procedure given in this paper is at least as accurate as spline interpolation, provided that accurate values are available for the slopes of the function at the given points. For a given accuracy, the spacing of tabulated points may be significantly greater than for linear interpolation.

## Appendix. Calculation of Slopes

Given the points  $x_j, y_j, j = 1, 2, \dots, n$ , the problem addressed in this Appendix is to compute slopes,  $y'_j$ , consistent with the requirements stated in the Introduction. It is assumed that the interpolating procedure given in this paper will be used.

In Figure 4, let I, J, and K be any three consecutive points. Point J may be above or below the line segment joining I and K, as shown in Figures 4(a) and 4(b), respectively. The requirements of the Introduction are satisfied if  $y'_j$  has a value between the slopes of the line segments IJ and JK. That is, for Figure 4(a), it is necessary that

$$\text{slope (IJ)} > y'_j > \text{slope (JK)} \quad (9a)$$

while for Figure 4(b),

$$\text{slope (IJ)} < y'_j < \text{slope (JK)} \quad (9b)$$

Another point is that if, for example, line segment IJ is much shorter than JK, it may easily be seen that a smoother overall interpolating curve will result if  $y'_j$  is nearly equal to the slope of IJ.

All of the above considerations are satisfied by setting  $y'_j$  equal to the slope at point J of a circle through points I, J, and K. Thus

$$y'_j = \frac{(y_j - y_i)((x_k - x_j)^2 + (y_k - y_j)^2) + (y_k - y_j)((x_j - x_i)^2 + (y_j - y_i)^2)}{(x_k - x_i)((x_k - x_j)^2 + (y_k - y_j)^2) + (x_j - x_i)((x_j - x_i)^2 + (y_j - y_i)^2)} \quad (10)$$

Equation (10) takes care of all interior points. The end points require special attention. From the requirements of the Introduction, it may be seen that the slope at an end point must have the same sign as the line segment from the end point to the next point. In line with the use of equation (10), one might try drawing a circle through the first (or last) three points. However, if either point I or K of Figure 4 is an end point, the particular arrangement of the points causes the slope of the circle at I or K to have the wrong sign. The sign of the slope is not necessarily wrong at the end point, but it may be wrong.

The problem divides into two cases. To simplify the notation, let M and subscript m designate either point I or point K, whichever is an end point, and let  $s$  be the slope of the line segment joining points J and M. The first case occurs when  $s$  is "steeper" than  $y'_j$ . In this case, a parabola through

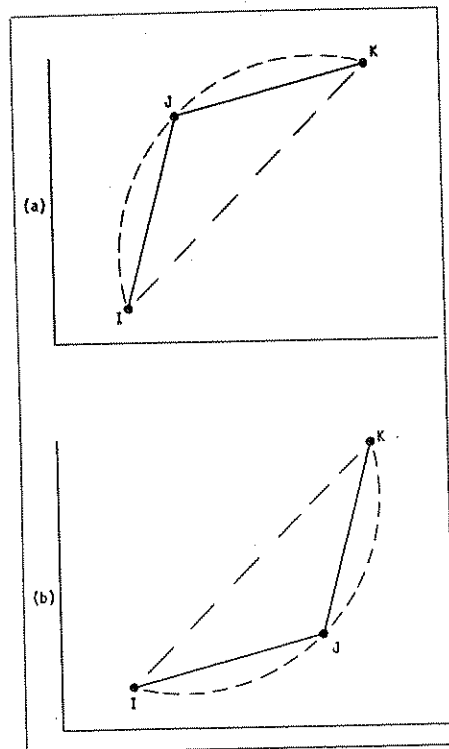


Figure 4

J and M with slope  $y'_j$  at J has a slope at M which meets the requirements and is reasonable. Stated more precisely, if  $s > 0$  and  $s > y'_j$ , or if  $s < 0$  and  $s < y'_j$ , then

$$y'_m = s + (s - y'_j) = 2s - y'_j \quad (11a)$$

The second case occurs if neither condition for the first case is satisfied.

**The procedure uses only ordinary arithmetic operations. The number of operations is approximately equivalent to the evaluation of a sixth-degree polynomial. Memory required for data is four words per point.**

In this case, the term in parenthesis in equation (11a) is multiplied by a factor between zero and one which assures that  $y'_m$  is always the same sign as  $s$ . The result is

$$y'_m = s + \frac{|s| (s - y'_j)}{|s| + |s - y'_j|} \quad (11b)$$

It should be understood that the slopes calculated by equations (10) and (11) are not independent of the scaling of the variables. For best results,  $x$  and  $y$  should be scaled to have roughly equal ranges, before calculating slopes. □

#### References

A. C. Ahlin, *On Smooth Interpolation by Continuously Connected piecewise Polynomials*, Rendiconti del Circolo Matematico di Palermo, Serie II, Tomo XX, 1971, pp 229-253.



"This system is not very friendly."

© Creative Computing

## BUSINESS - PROFESSIONAL - GAME SOFTWARE FOR APPLE AND TRS-80

### HOME FINANCE PAK I: Complete package \$49.95 Apple, TRS-80

- BUDGET:** The heart of a comprehensive home finance system. Allows user to define up to 20 budget items. Actual expense input can be by keyboard or by automatic reading of CHECKBOOK II files. Costs are automatically sorted and compared with budget. BUDGET produces both monthly actual/budget/variance report and a year-to-date by month summary of actual costs. Color graphics display of expenses. . . \$24.95
- CHECKBOOK II:** This extensive program keeps complete records of each check/deposit. Unique check entry system allows user to set up common check purpose and recipient categories. Upon entry you select from this pre-defined menu to minimize keying in a lot of data. Unique names can also be stored for completeness. Rapid access to check files. Check register display scrolls for ease of review. 40 column print-out. Up to 100 checks per month storage. Files accessible by BUDGET program. . . \$19.95
- SAVINGS:** Allows user to keep track of deposits/withdrawals for up to 10 savings accounts. Complete records shown via screen or 40 column printer. . . \$14.95
- CREDIT CARD:** Keep control of your cards with this program. Organizes, stores and displays purchases, payments and service charges. Screen or 40 column printer display. Up to 10 separate cards. . . \$14.95

### THE UNIVERSAL COMPUTING MACHINE: \$39.95 Apple, TRS-80

A user programmable computing system structured around a 20 row x 20 column table. User defines row and column names and equations forming a unique computing machine. Table elements can be multiplied, divided, subtracted or added to any other element. User can define repeated functions common to a row or column greatly simplifying table setup. Hundreds of unique computing machines can be defined, used, stored and recalled, with or without old data, for later use. Excellent for sales forecasts, engineering design analysis, budgets, inventory lists, income statements, production planning, project cost estimates-in short for any planning, analysis or reporting problem that can be solved with a table. Unique cursor commands allow you to move to any element, change its value and immediately see the effect on other table values. Entire table can be printed by machine pages (user-defined 3-5 columns) on a 40 column printer. Transform your computer into a UNIVERSAL COMPUTING MACHINE.

### COLOR CALENDAR: HI-RES color graphics display of your personal calendar. Automatic multiple entry of repetitive events. Review at a glance important dates, appointments, anniversaries, birthdays, action dates, etc. over a 5 year period. Graphic calendar marks dates. Printer and screen display a summary report by month of your full text describing each day's action item or event. Ideal for anyone with a busy calendar. . . (Apple Only) . . . \$19.95

### BUSINESS SOFTWARE SERIES: Entire package \$239.95 Apple, TRS-80

- MICROACCOUNTANT:** The ideal system for the small cash business. Based on classic T-accounts and double-entry bookkeeping, this efficient program records and produces reports on account balances, general ledger journals, revenue and expenses. Screen or 40 column printer reports. Handles up to 500 journal entries per period, up to 100 accounts. Instructions include a short primer in Financial Accounting. \$49.95
- UNIVERSAL BUSINESS MACHINE:** This program is designed to SIMPLIFY and SAVE TIME for the serious businessman who must periodically Analyze, Plan and Estimate. The program was created using our Universal Computing Machine and it is programmed to provide the following planning and forecasting tools:  
CASH FLOW ANALYSIS      PROFORMA BALANCE SHEET      SOURCE AND USE OF FUNDS  
PROFORMA PROFIT & LOSS      SALES FORECASTER      JOB COST ESTIMATOR

Price, including documentation and a copy of the base program. Universal Computing Machine. . . \$89.95

- INVOICE:** Throw away your pens. Use the ELECTRONIC INVOICE facsimile displayed on your CRT. The program prompts and you fill in the data. Includes 3 address fields (yours, Bill to and Ship to), Invoice No., Account No., Order No., Salesman, Terms, Ship Code, FOB Pt. and Date. Up to 10 items per sheet with these descriptions: Item No., No. of units, Unit Price, Product Code, Product Description, Total Dollar amount per item and invoice total dollar amount. Generates, at your option, hard copy invoices, shipping memos, mailing labels, audit copies and disc updates to master A/R files. Compatible A/R module available Aug. - 1980 (48K) . . . \$49.95

- MAILING LIST:** The best mailing list program available, bar none. Options for keyboard or file input, alphanumeric sort on any of 7 fields, including zip code sort for bulk mailing. Multiple addresses per line, vertical space adjust. Up to 500 records per disc (48K) . . . \$49.95
- BUSINESS CHECK REGISTER:** Expanded version of the Checkbook II program. Handles up to 500 checks per month with complete record keeping. (48K) . . . \$29.95
- BUSINESS BUDGET:** As described above and companion program to Business Check Register. Handles 500 transactions per month, up to 20 cost categories. Accesses BCR files for actual costs. (48K) . . . \$29.95

### ELECTRICAL ENGINEERING SERIES: Both programs \$159.95 Apple only

- LOGIC SIMULATOR: SAVE TIME AND MONEY.** Simulate your digital logic circuits before you build them. CMOS, TTL, or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time by bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including NANDS, NORs, INVERTERS, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS and user-defined MACROS. Edge triggered or 2 phase clocks. Up to 40 user-defined, random, or binary input patterns. Simulation results displayed on CRT or printer. Accepts network descriptions from keyboard or from LOGIC DESIGNER for simulation. Specify 1000 gate version (48K required) or 500 gate version (32K required) . . . \$89.95

- LOGIC DESIGNER:** Interactive HI-RES Graphics program for designing digital logic systems. A menu driven series of keyboard commands allows you to draw directly on the screen up to 15 different gate types, including 10 gate shape patterns supplied with the program and 5 reserved for user specification. Standard patterns supplied are NAND, NOR, INVERTER, EX-OR, T-FLOP, JK-FLOP, D-FLOP, RS-FLOP, 4 Bit COUNTER and 8-BIT SHIFT REGISTER. User interconnects gates just as you would normally draw using line graphics commands. Network descriptions for LOGIC SIMULATOR generated simultaneously with the CRT diagram being drawn. Drawing is done in pages of up to 20 gates. Up to 50 pages (10 per disc) can be drawn, saved and recalled. Specify 1000 gate (48K) or 500 gate (32K) system . . . \$89.95

### MATHEMATICS SERIES: Complete Package \$49.95 Apple only

- NUMERICAL ANALYSIS: HI-RES 2-Dimensional plot of any function. Automatic scaling. At your option, the program will plot the function, plot the INTEGRAL, plot the DERIVATIVE, determine the ROOTS, find the MAXIMA and MINIMA and list the INTEGRAL VALUE. For 16K . . . \$19.95**
- MATRIX:** A general purpose, menu driven program for determining the INVERSE and DETERMINANT of any matrix, as well as the SOLUTION to any set of SIMULTANEOUS LINEAR EQUATIONS. Disk I/O for data save. Specify 55 eqn. set (48K) or 35 eqn. (32K) . . . \$19.95
- 3-D SURFACE PLOTTER:** Explore the ELEGANCE and BEAUTY of MATHEMATICS by creating HI-RES PLOTS of 3-dimensional surfaces from any 3-variable equation. Disc save and recall routines for plots. Menu driven to vary surface parameters. Demos include BLACK HOLE gravitational curvature equations. . . \$19.95

### ACTION ADVENTURE GAMES SERIES: Entire series \$49.95 Apple only

- RED BARON:** Can you outfly the RED BARON? This fast action game simulates a machine-gun DOG-FIGHT between your WORLD WAR I BI-PLANE and the baron's. You can LOOP, DIVE, BANK or CLIMB in any one of 8 directions - and so can the BARON. In HI-RES graphics (16K) . . . \$19.95
- BATTLE OF MIDWAY:** You are in command of the U.S.S. HORNETS' DIVE-BOMBER squadron. Your targets are the Japanese carriers, Akagi, Soryu and Kaga. You must fly your way through ZEROs and AA FIRE to make your DIVE-BOMB run. In HI-RES graphics (16K) . . . \$19.95
- SUB ATTACK:** It's April, 1943. The enemy convoy is headed for the CORAL SEA. Your sub, the MORAY, has just sighted the CARRIERS and BATTLESHIPS. Easy pickings. But watch out for the DESTROYERS - they're fast and deadly. In HI-RES graphics (16K) . . . \$19.95

**FREE CATALOG-**All programs run on Apple II w/Disc II and Applesoft ROM Card & TRS-80 Level II and require 32K RAM unless otherwise noted. Detailed instructions included. Orders shipped within 48 hours. Card users include card number. Add \$1.50 postage and handling with each order. California residents add add 6 1/2% sales tax. Make checks payable to:



**SPECTRUM SOFTWARE**

DEALER INQUIRIES      P.O. BOX 2084 - SUNNYVALE, CALIFORNIA 94087  
INVITED      FOR PHONE ORDERS - 408-245-1415