original

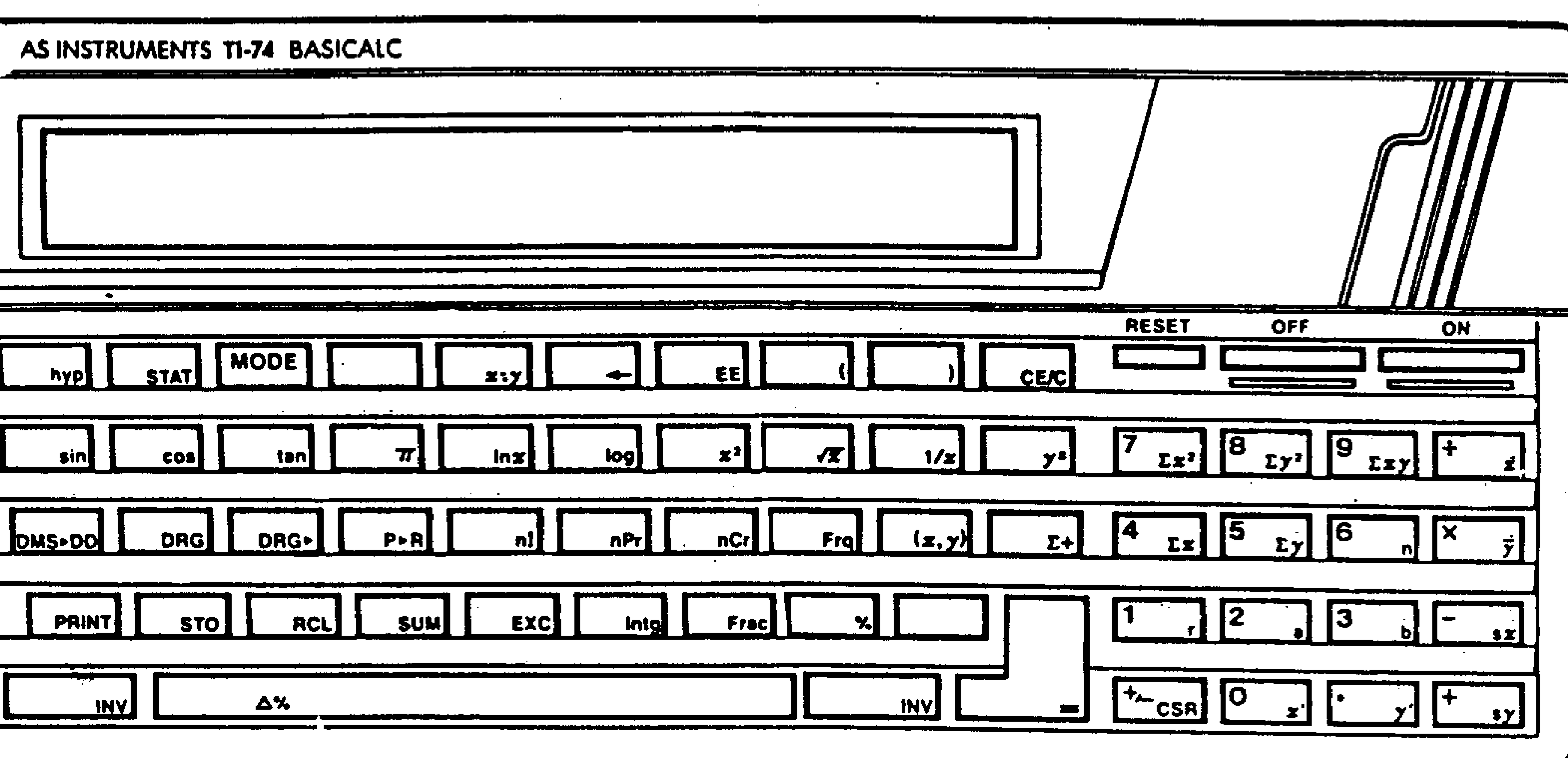
A Collection of Information on The CC-40 and TI-74 Computers

by

Palmer O. Hanson, Jr.
Editor - TI PPC Notes

May 1987

This collection is a compilation of articles on the CC-40 and TI-74 and peripherals from the Volume 11 (1986) issues of TI PPC Notes.



The full-size illustration above was extracted from pages ii and iii of the TI-74 User's Guide. The key nomenclature for the calculator mode is shown. Different key assignments are used for the BASIC mode; many of those functions are discussed on pages 4 ff. Generally, the key assignments in BASIC are very similar to those on the CC-40. The alphabet portion of the BASIC key assignment is pseudo-QWERTY. Note that the keys are not staggered from row to row.

Quirks in the QuickDisk for the CC40

It was a great leap forward when the QuickDisk became available for the CC40 because it solves the off-line storage problem. Unfortunately, the QuickDisk has certain quirks which make it difficult to use.

1. Buffer is 16K. Any record which is read from the QuickDisk or written to the QuickDisk is read out of a single buffer in a one-write situation. That buffer has an upper limit of 16.3 Kilobytes. In Data applications, that is small. Whenever you have a Data File that is greater than 16 Kilobytes, you have to break it into several different files and access them individually. This is a problem because of the One-File-Open Rule.

2. One-File-Open Rule. Most disk operating systems will let you open at least 3, up to 8, and some I have seen as many as 32, disk files simultaneously. The QuickDisk lets you open 1 (one) disk file. In order to read another file, you must close the first file, and re-open the second file. This follows because it reads the entire file into the buffer memory, and since the buffer memory is only 16 Kilobytes, it has to clear it to free it for the next record. (problem in Data applications.)

3. Opening Modes. When a file is opened in the Input mode, a copy is not re-written to the diskette when it is closed. This is a very handy option when several files are to be scanned. However, when a file is opened in Output or Append or Update mode, it is essentially written sequentially and it will be rewritten into fresh space. The most practical of course, is the Update, and when you open in Update, the entire file is read from the disk into the buffer, changes are made while it is in the buffer, and the entire file is rewritten to the diskette in fresh space.

4. Pseudo-magnetic ROM. Clearing the disk resets it to all clear. As files are written, they are written sequentially, into unused space. When a file is opened in Update or Append or Output mode, it is read into the buffer. Even if no change is made, closing will make the QuickDisk write the file again in unused space, letting the original one stand also. So we have two copies of the same file on the diskette. The operating software of the QuickDisk will sort thru these and only read in the second one when it is re-opened. And again, if it is opened in Update, Append or Output, it will write a third copy of the file when it is closed.

5. Record Separators. In sequential mode, the QuickDisk seems to insert its own separators, although this is not a constant finding. However, in Random mode, the QuickDisk does not insert its own separators, such as returns or commas, and you have to insert these. If you do not insert these, you will get an error when you try to read the file. (It cannot find the end and continues to read.)

6. Creation of a Record. This QuickDisk basically thinks and acts like a sequential output device, that can provide, in some cases, a limited amount of random access. If a file is opened in the Random mode, and records 1, 2, 3, and 4 are written, and then you choose to write in record 10, you will get an error, because you have to write the records sequentially, even though you can access already written records randomly, you cannot write records in a portions of the file which has not been initialized in a sequential step-wise fashion. Therefore, this is a Pseudorandom Access.

7. Record Zero. When a file is opened, whether it be sequential or random modes, the index pointer in a QuickDisk is at the Zero Record position. You can write into this record, and in sequential files, this is perfectly usable. However, in random files, the record zero cannot be reaccessed. If the file is initially written into Record Space 0, 1, and 2, and then you move the Index Pointer with a Restore Record Command to 0, you would think it would point the index back to record 0, but it doesn't. It increments and points it to 3. It interprets a restore 0 as an increment to the next available record. Therefore the machine thinks like a sequential, although limited random access is available in an already created file.

8. Closing Error. If a CLOSE #N is executed, but there isn't enough room on the disk to save the file, an error message is returned. The QuickDisk does not save the file, and considers the file to stay open. However, the FILE BLOCK in the CC40 is destroyed, and the CC40 considers the file closed. This is unfortunate because future disk requests are blocked because it thinks it has a file open. When this occurs, the solution is to unplug the QuickDisk, wait a few seconds and plug it in, or to send a CALL IO (8,255).

With these problems in mind, the QuickDisk is a very practical storage device for small files, and these files can be concatinated thru software to use the entire disk.

Factorials on the CC-40 - V8N5P18 noted that factorials could be found on the CC-40 with the Mathematics module by recognizing that N! = Gamma(N+1). The final line stated that the CC-40 could then find factorials up to 85!. That should have been factorials up to 84! = 3.31424E+126.

SOCIAL SECURITY NUMBER PUZZLE

In V10N4P12 Maurice Swinnen proposed the puzzle of finding a nine digit number for which the first two digits on the left are divisible by two, the first three numbers are divisible by three, etc., up to the entire number is divisible by nine. The number does not contain zeroes, and no digit is repeated. Maurice proposed a very slow computer solution. Members were asked to write a faster solution.

The correct answer is 381654729. Larry Leeds and Robert Prins obtained the solution without the use of a computer program. Maurice Swinnen submitted a program from a friend written for a Radio Shack computer. I converted it for use on the CC-40 (listing at the right); it obtains the solution in about six minutes. George Thomson submitted a longer program which obtains the solution in only 32 seconds on his Panasonic Senior Partner running in ZBASIC.

All of the solutions relied on the idea that the fifth digit must be a five, and the even digits must be even, and proceed to shuffle and test the digits according to some algorithm. More on the methods in the next issue.

```
100 S(5)=5
2000 FOR A=2 TO 8 STEP 2
2020 FOR B=2 TO 8 STEP 2
2030 IF B=A THEN 2140
2040 FOR C=2 TO 8 STEP 2
2050 IF C=A OR C=B THEN 2120
2060 FOR D=2 TO 8 STEP 2
2070 IF D=A OR D=B OR D=C THEN 2100
2080 S(2)=A:S(4)=B:S(6)=C:S(8)=D
2090 GOTO 3000
2100 NEXT D
2120 NEXT C
2140 NEXT B
2160 NEXT A
3000 FOR W=1 TO 9 STEP 2
3010 IF W=5 THEN 3560
3020 FOR X=1 TO 9 STEP 2
3030 IF X=W OR X=5 THEN 3540
3040 FOR Y=1 TO 9 STEP 2
3050 IF Y=W OR Y=X OR Y=5 THEN 3520
3060 FOR Z=1 TO 9 STEP 2
3070 IF Z=W OR Z=X OR Z=Y OR Z=5 THE
N 3500
3075 J=J+1:IF J>570 THEN END
3080 S(1)=W:S(3)=X:S(7)=Y:S(9)=Z
3090 S$="":FOR Q=1 TO 9:S$=S$&STR$(S
(Q))
3100 IF Q<3 OR Q=5 THEN 3400
3120 SS=VAL(S$)
3140 IF (SS/Q-INT(SS/Q))>.01 THEN 35
00
3400 NEXT Q
3420 PRINT "The number is ":SS
3450 PAUSE
3500 NEXT Z
3520 NEXT Y
3540 NEXT X
3560 NEXT W
3600 GOTO 2100
```

QUIRKS IN THE QUICKDISK FOR THE CC-40 - Louis Krumpelman. Earlier issues have noted the availability of a disk drive for use with the CC-40. There are indications that the Disk drive may also be useable with the TI-74. The following page presents the experience of one user of the device.

CC-40 MATERIAL FOR SALE - Maurice Swinnen has extra CC-40 material. He has divided the equipment into two systems, and will sell by system, not by individual components.

System 1:	CC40 with 18K Memory (new)	$ 375
	HX-1010 80 Column Thermal printer (new), with 1000 sheets of thermal paper, 500 sheets of glossy plain paper, 8 rolls of 8.5" x 100 ft thermal roll paper, and 6 boxes of thermal offset ribbons.	200
	HX-1000 Four Color Plotter-Printer (new)	200
	Mathematic Solid State Cartridge	60
	Electronics Engineering Solid State Cartridge	60
	Statistics Solid State Cartridge	60
	Mechatronic Disk Drive	295
	Ten Diskettes	50

	New Price	$1300
	Sale Price	$850

System 2:	Engineering prototype CC-40 with 18K Memory HX-1010 80 Column Thermal Printer + mod to Use Roll Paper HX-3000 RS-232 Interface	
	Sale Price	$400

PC-100C Printer (new) with 6 Rolls of Thermal Paper $150

Write to Maurice Swinnen, 9213 Lanham Severn Road, Lanham MD 20706, or call during working hours (Eastern time) at 301-427-5125. This distribution of material in the two systems is tentative. He is open to suggestions. He will ship by UPS or other carrier per your instructions. You pay the shipping in addition to the prices quoted above.

THE MATHEMATICS LIBRARY CARTRIDGE FOR THE TI-74 - The program package is very similar to that which was available with the Mathematics module for the CC-40. The two discrepancies with the CC-40 module have been corrected, namely the prompting anomaly in the AU routine (V8N5P18) and the wrong sign for the determinant for some matrices (V10N1P7).

THE TI-74 BASICALC - Maurice Swinnen reports that the title page of a manual for the Mechatronic Quick Disk (see V10N4P4) indicates that the device can be used with both the CC-40 and the TI-74 BASICALC. This seems to confirm the notation that appeared in the Executive Photo Catalog (see V10N4P3), but as yet there is nothing from TI's Consumer Press Relations organization.

THE TI-74 - Palmer Hanson. In mid-July the TI-74 became available from Elek-Tek. Page 18 of their Volume 12 catalog shows the following items and prices:

417627	TI-74 BASICALC	$94.00
417635	8K Constant Memory Cartridge	34.00
417643	Printer	69.00
417651	Cassette Interface	22.00
417669	Learn Pascal Software Cartridge	29.00
417677	Statistics Software Cartridge	29.00
417685	Mathematics Software Cartridge	29.00

Those prices are substantially lower than those quoted in V10N4P3 from the Executive Photo Catalog. You can call Elek-Tek toll-free at 800-621-1269 and order by credit card if you like. There is a shipping charge of $4.00 for the first item and an additional $1.00 for each additional item. As of mid-August only the TI-74 and the software cartridges were in stock.

A full size illustration of the TI-74 appears on the front page. It is somewhat smaller than the CC-40, and comes with a snugly fitting plastic carrying case. It will have to be removed from the carrying case to connect to the printer and cassette interface. The advertisement in the Elek-Tek catalog calls the device "An advanced scientific calculator that is BASIC programmable". The CALC mode provides most of the functions which we have come to expect from a scientific calculator from TI including statistics, linear regression, hyperbolic functions, permutations and combinations. The BASIC mode provides many of the functions of the CC-40.

The CALC and BASIC mode both display ten digits and seem to use the same internal mathematics carrying 13 or 14 digits depending upon the number. Thus, in BASIC mode

e = EXP(1) = 2.718281828458 which includes 13 digits, and

e^3 = (EXP(1)^3) = 20.085536923165 which includes 14 digits.

That is generally consistent with Larry Leeds' discussion of the arithmetic of the TI-99/4 and CC-40 in V9N5P6. The "Itsy-bit of Paranoia" test from the February 1985 issue of BYTE (see V10N2P16 for a listing which will run on the TI-74 if each "Print #1" is changed to "Print") yields exactly the same results as the CC-40 with radix = 100, precision = 7, fpwidth = 1.E+14, ulpone = 1.E-14, and the existence of a guard digit for add/subtract.

One of the major deficiencies of the CC-40 keyboard has been removed; the TI-74 has a shift key at the right of the space bar. There are some other keyboard features which are inconvenient. The keyboard is "typewriter-like" for the alphabet and the punctuation symbols, but is entirely too small to be used like a typewriter by any but the smallest hands. To accomodate both the the calculator functions and the Basic functions the upper row of a normal typewriter keyboard (numbers and symbols) has been replaced by a functions row, and numbers and symbols can only be entered from the calculator keypad at the right. One result is that the arrow keys used for editing in Basic, which were directly above the calculator keypad at the right side of the keyboard on the CC-40, have been moved toward the center of the keyboard.

The TI-74 - (cont)

That change is particularly inconvenient for me since the two computers that I used most in the past, the CC-40 and the Model 100, had those keys in very similar positions on the keyboard. However, I have become resigned to the idea that calculator/computer manufacturers change keystrokes and key positions with total disregard for the habits of users. For example, any serious TI-58/59 user had developed a habit of thinking of the Op command as 2nd-Op, the Label command and 2nd-Lbl, etc. as a "crutch" for remembering the keyin sequence for rapid entry of program commands. The TI-66 changed all that, for no apparent good reason so far as I can see.

I tested the TI-74 arithmetic with some of the same benchmarks which we have discussed in earlier issues and obtained mixed results:

1. e x π was equal to π x e indicating that multiplication was commutative. The non-commutative multiply on the TI-59 was discussed in V9N2P15.

2. sin(45) was not equal to cos(45), and in general sin(X) was not equal to cos(90 - X). Page A-33 of the TI-74 Programming Reference Guide makes the helpful suggestion that:

> "A useful technique is to test whether two values are sufficiently close together rather than absolutely equal ...
> ... Instead of IF X=Y THEN ... use
> IF ABS(X-Y) < 1E-11 THEN ... "

That is a safer technique than the idea of doing EE-INV-EE before a comparison as proposed in Personal Programming for the TI-59. Of course the size of the test value depends upon the problem.

3. The square root-squared test: V8N3P13/14 described this test which is a derivative of the $(\sqrt{2})^2$ test by Brian Hayes on page 136 of the January 1981 issue of BYTE. For our test we start with an integer, take the square root five times, take the square five times and compare the result to the original number. I tested selected integers from 2 through 17. The display returned the starting integer in each case. The actual values before truncation to the display were:

2	1.99999 99999 83	12	12.00000 00001 26
3	3.00000 00000 04	13	12.99999 99998 13
5	4.99999 99999 70	15	15.00000 00002 91
7	7.00000 00000 71	17	17.00000 00000 69

where all of the answers are better than the TI-59. Again, note that some solutions show 13 digits and other solutions show 14 digits.

4. 1.0000001 squared 27 times: V9N2P11 described this test from the "Computer Recreations" column of the April 1984 issue of Scientific American, where there are different methods of calculation:

Exact	674530.4707410 84559 ...
Mode A (Repeated A^2)	674530.3180422 5
Mode B (Repeated A*A)	674530.3180422 5
Mode C (A^134217728)	674530.4707401 0

where in tests of other devices only the Model 100 yielded a better answer than the Mode C solution from the TI-74.

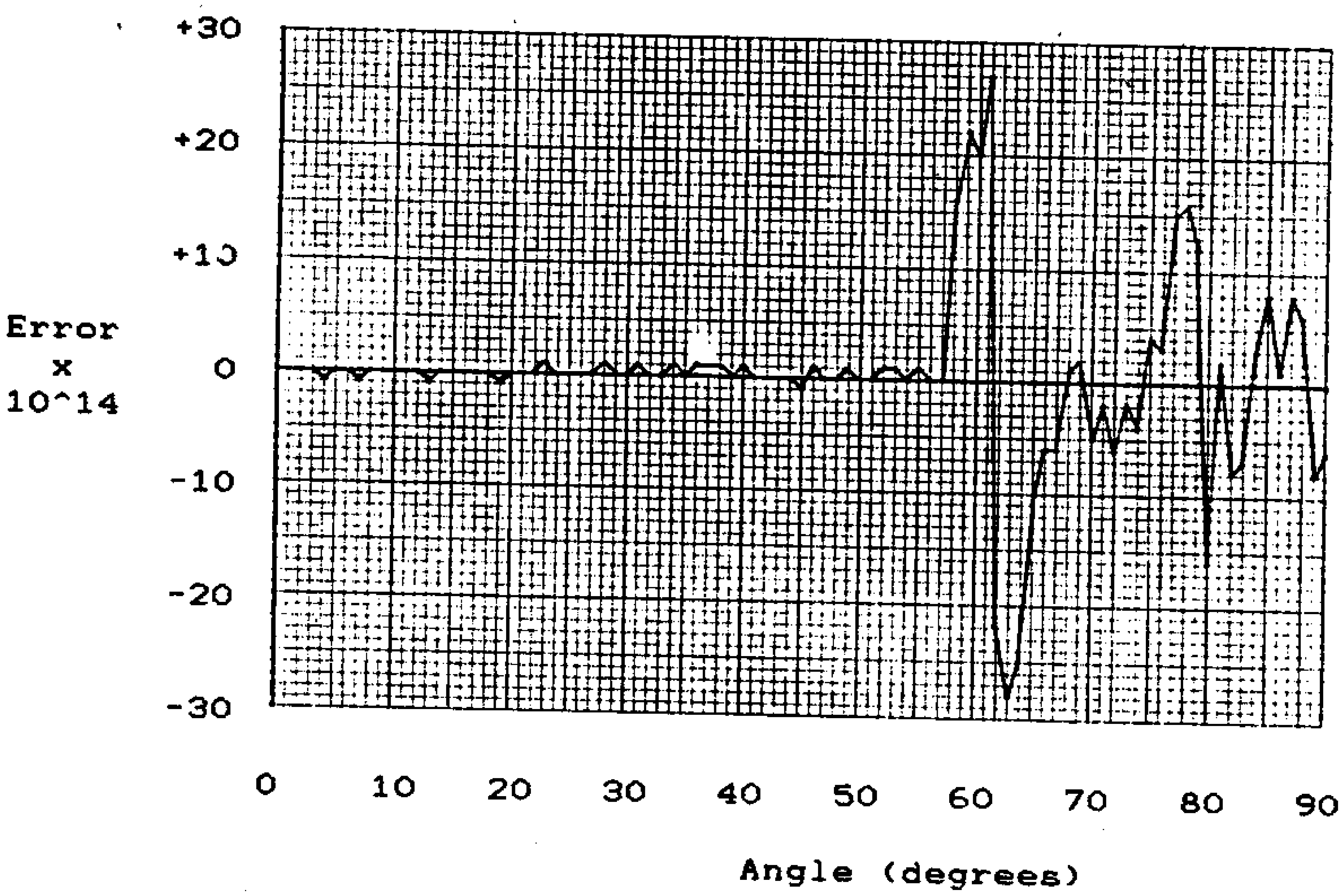
The TI-74 - (cont)

5. The Bob Fruit benchmark: Bob proposed a compound interest problem as another benchmark (see V8N4P4). The appropriate equation is that for the sum of a geometric series S = [(1 + i)^n -1]/i . An annual interest rate of ten per cent (i = .10/12) and compounding monthly for thirty years (n = 360) yields:

Exact	2260.48792 47960 86067 ...
TI-74 using the y^x function	2260.48792 45128

where that answer is better than the CC-40 or TI-99/4, but not as good as the TI-59 or TI-66.

6. Accuracy of the sine function:
The errors in the output of the sine function from the TI-74 are illustrated in the following figure:



For input angles less than one radian (57.2957... degrees) the error is never greater than 1E-14; but for input angles greater than one radian the errors become erratic. For one degree increments over the 0 to 90 degree range the RMS errors was 8E-14 and the peak error was 28E-14. Those values are substantially better than the TI-59, TI-66, or CC-40 and only slightly worse than the Radio Shack Model 100.

It occurred to me that one might be able to extend the very good accuracy for the sine function by using cos(90 - X) for some angles, say for angles above one radian. Examination of the cosine function showed that is not the case. In fact, for the range from 58 degrees through 90 degrees the error in the cos(90 - X) function was exactly the same as the error in the sinX function for 29 of the 33 angles tested. The cosine function is also substantially less accurate than the sine function for angles below one radian with a peak error of 56E-14 and an RMS error of 23E-14 over that range. This suggests that for the most accurate results the user might want to use sin(90 - X) or sqr(1 -sin(X)^2) instead of cos(X) wherever possible.

THE TI-74 RANDOM NUMBER GENERATOR - The RND function provides access to a series of predetermined uniformly distributed pseudo-random numbers with values in the range from 0 to 1. The same sequence of values will be returned each time a program is run. Using a little tallying program to determine the distribution of the RND outputs in ten equal width "buckets" the results for 1000, 10,000 and 100,000 random numbers were

	1000	10,000	100,000
D(1)	113	1,017	9,921
D(2)	95	973	9,999
D(3)	75	958	10,092
D(4)	95	1,013	9,892
D(5)	104	1,008	10,185
D(6)	98	971	9,943
D(7)	105	974	10,044
D(8)	95	1,047	10,031
D(9)	112	1,028	9,994
D(10)	108	1,011	9,899

```
10 FOR I=1 TO 1000
20 A=10*RND+.5
30 D(A)=D(A)+1
40 NEXT I
50 FOR I=1 TO 10
60 PRINT D(I)
70 PAUSE:NEXT I
99 END
```

where the distribution of 1000 numbers is identical to that presented for the CC-40 on V10N1P25. Tests show that the number sequence delivered by the RND function in the TI-74 is identical to that obtained from the CC-40.

The tallying program uses the indirect address sorting technique described in V10N1P24 and V10N3P13. Lines 20 and 30 rely on the characteristic of the TI-74 which selects the address for a non-integer subscript by rounding to the nearest integer. That is another TI-74 feature which is common with the CC-40. As written the program counts into D(1) through D(10). An alternative would have been to write line 20 as A = INT(10*RND) which would count into D(0) through (9). In either case no dimensioning statement is needed since BASIC provides automatic dimensioning of any variable up to 10.

WHAT'S MISSING IN THE TI-74? - The preceding discussion has noted the similarity of the features of the TI-74 to those of the CC-40. There are some attractive features of the CC-40 which were dropped:

* The BEEP which allowed the user to sound a tone as an attention getter, say for erroneous inputs, the end of calculations, etc.

* The CHAR command which allowed the user to define his own set of characters on the CC-40.

* An external power supply to minimize battery useage. Page 1-5 of the TI-74 Users Guide cautions:.

  "The Constant Memory (TM) feature retains stored information for a short time after the batteries are removed. As a precaution, however, you may want to save any important programs and data on a storage device (such as a cassette) before replacing batteries."

* An RS-232 capability.

COMBINATORIAL ANALYSIS ON THE TI-74 - The CALC mode of the TI-74 provides built-in functions for factorials, permutations and combinations. The larger numerical range of the TI-74 (10^127 as opposed to 10^99 with the TI-59, another feature which is common with the CC-40), permits the calculation of factorials up to 84! = 3.314240135E+126. The solution is very fast; the answer seems to appear in the display as soon as the factorial key is pressed. V11N1P5 also reported execution times for benchmark permutation and combination problems. The TI-74 returns the permutations of 100 items taken 50 at a time in about a second, and the combinations of 328 items taken 164 at a time in about 3 seconds. That makes it the fastest hand-held/lap-top device tested so far.

THE STATISTICS LIBRARY CARTRIDGE FOR THE TI-74 - The program package is very similar to that which was available with the Statistics module for the CC-40. Even the examples in the documentation are the same. The major differences are in the responses to prompts. For example, the first prompt with the Means and Moments program asks "Use Printer?", where the acceptable answers are "Y" or "y" for yes and "N" or "n" for no. With the CC-40 the user entered one of those letters and pressed ENTER to proceed to the next step. With the TI-74 the machine moves immediately to the next step on entry of an apropriate response. That should provide some increase in speed of operation; however, at present I am so conditioned to the CC-40 sequence that I press ENTER anyway. The CC-40 sounds a tone when the response to a prompt is not acceptable. The TI-74 can't do that since it does not have a BEEP command.

Page 2-18 of the manual defines the geometric mean as

$$Xg = (X_1^{f_1} \times X_2^{f_2} \times \ldots\ldots \times X_K^{f_K})^{1/N} \qquad \text{where } N = \sum f_K.$$

That is often written as $Xg = [\prod_{i=1}^{n} X_i^{f_i}]^{1/N}$

The actual Means and Moments routine is probably not mechanized in that way. If it were, you would cause an overflow condition as soon as the product equals or exceeds 10E127. Test cases will show that such overflow does not occur. It seems likely that the geometric mean routine is mechanized using a sum of log(X) formulation as indicated on page 23 of the manual for the Statistics cartridge for the CC-40.

$$Xg = 10\hat{}[(\sum_{i=1}^{n} f_i * \log(x_i))/N]$$

V10N2P22 noted that there was an error in the formula for finding the geometric mean on page 23 of the manual for the Statistics cartridge of the CC-40 and gave a correct formula as shown above. V10N2P3 noted that overflow problems with the calculation of the geometric mean were delayed in the CC-40 because of the log formulation.

MORE SUBPROGRAMS IN THE STATISTICS MODULE FOR THE CC-40 AND TI-74

V8N6P20 reported that the Statistics cartridge for the CC-40 had subprograms which were not discussed in the manual. One subprogram would input and edit a two-dimensional array in a manner very similar to the MI subprogram in the CC-40 Mathematics cartridge. That capability is also available in the Statistics cartridge for the TI-74, and again there is no mention in the manual. The call is the same as that defined on page 95 of the manual for the Mathematics cartridge for the CC-40:

```
CALL MI(PROMPT$,ARRAY(,),FIRST,LAST ROW,LAST COLUMN,PRINTER)
```

An example will illustrate the use of this call:

```
10  CALL MI("A",A(,),1,2,3,0)
```

The first prompt which will appear is "Enter A(1,1):". As you enter each element of the array the prompts will proceed through "Enter A(1,2):", "Enter A(1,3):", "Enter A(2,1):", etc. If the last variable in the call is a one then each entry is printed together with the accompanying notation. When the last array element has been entered the subprogram enters an edit mode. The prompts and responses for the TI-74 Statistics cartridge are the same as those described for the Mathematics cartridge of the CC-40. The CC-40 Statistics cartridge implementation differs in that it exits the subprogram at the end of an edit of all the input.

Subprogram ME uses the same arguments as subprogram AU but starts execution at the edit routine.

V8N6P20 also noted that there also appeared to be an AU subprogram for input and edit of two one-dimensional arrays in the Statistics cartridge of the CC-40, but the use of the AU call defined on page 87 of the Mathematics cartridge manual yielded a response "Illegal Syntax". After some additional experimentation I have found that a longer call will work for the Statistics cartridges in both the CC-40 and the TI-74:

```
CALL AU(PROMPT1$,PROMPT2$,ARRAY1(),ARRAY2(),FIRST,LAST,
     PRINTER,EXTRA1,EXTRA2)
```

where the EXTRA1 and EXTRA2 elements must be added for use with the Statistics cartridges. An example will illustrate the use of this call:

```
10 N=3
20 CALL UP("Demo of AU in Statistics Module",PN)
30 CALL AU("X","Y",X(),Y(),1,N,PN,0,0)
```

Line 20 is the printer useage subprogram. If the user responds "N" to the prompt "Use Printer?" then PN is set to zero and the computer proceeds to the next line of the program. If the user responds to the prompt "Use Printer?" with "Y" then PN is set to one, and the next prompt asks "Enter File Name:". The user enters the device number. The text inside the quotation marks in the call is printed and the computer proceeds to the next line of the program.

Line 30 is the AU call. This call illustrates that the FIRST, LAST, PRINTER, EXTRA1, and EXTRA2 elements of the call can be constants or variables. The first prompt is "Enter X(1)". When that element has been entered the next prompt is "Enter Y(1)". As each subsequent element is entered the prompt alternates between requests for X or Y elements. When all of the elements of the two arrays have been entered the subprogram enters an editing routine.

More Subprograms for the Statistics Modules - (cont)

The reasons for the elements EXTRA1 and EXTRA2 in the call for the AU subprogram are not clear. Experimentation shows that if EXTRA2 is not zero, then the value for EXTRA1 is added to the value for FIRST to define the subscript for the first prompt for data entry. For example if in line 30 above EXTRA1 is two and EXTRA2 is one, then the first prompt will be "Enter X(3):", the second prompt will be "Enter Y(3):", etc. up through "Enter Y(5):". But if you test the contents of the X and Y arrays you will find that the entered values are in locations with subscripts 1 through 3 not 3 through 5. Thus the EXTRA1/EXTRA2 elements of the call seem to affect the subscripts of the prompt, but not the locations into which the entries are stored! I am at a loss to explain how this is useful. Until we have a better understanding of the effects of the EXTRA1 and EXTRA2 elements of the AU call I recommend that users set them to zero.

Subprogram AE uses the same arguments as subprogram AU but starts execution at the edit routine.

Subprogram RZ checks to see if an input variable is greater than zero. The argument for the call is the same as for the IZ subprogram:

```
CALL RZ(Variable, Flag)
```

If the variable is greater than zero the flag is set to 0. If the variable is less than or equal to zero the flag is set to -1 and the message "VALUE MUST BE > 0" is displayed. However, the message is only displayed for a short period of time. To see the message s user should include a PAUSE statement immediately following the call.

Subprogram IC checks to see that an input value is an integer between two predetermined limits. The call is:

```
CALL IC(Lower, Upper, Variable, Flag)
```

If the variable is not an integer the flag is set to -1 and the message "VALUE MUST BE AN INTEGER" is displayed. If the variable is between the limits the flag is set to zero. If the variable is outside the limits the flag is set to -1 and a message defining which limit was exceeded is displayed. An example will illustrate the use of this call:

```
10 A = -3.5
20 B = 10
30 INPUT N
40 CALL (A,B,N,Q)
50 IF Q=0 THEN PAUSE
60 PRINT "Flag = ";Q:PAUSE
70 GOTO 30
```

Note that the limits do not need to be integers. As with the IZ, RZ and PC subprograms the conditional PAUSE at line 50 permits the user to view the error message. Line 60 allows the user to check the flag response. An input of 5 yields only the message "Flag = 0". An input of yields the message "VALUE MUST BE >=-3.5." followed by the message "Flag = -1". An input of 11 yields the message "VALUE MUST BE <=10." followed by the message "Flag = -1". If the values for A and B are reversed so that the first element of the call is greater than the second element of the call then the flag is set to -1 for all input, and some of the messages are non-sensical. If the two limits are set to the same integer, then only that integer as an input will set the flag to 0.

<u>More Subprograms for the Statistics Modules</u> - (cont)

Subprogram PC checks to see than an input value is greater than zero and less than or equal to one. The argument for the call is the same as for the IZ and RZ subprograms:

CALL PC(Variable, Flag)

If the variable is inside the defined range then the flag is set to zero. If the variable is outside the defined range then the flag is set to -1 and the message "VALUE MUST BE >0 And <=1" is displayed for a short period of time. Again, a following PAUSE statement should be included.

There is one additional subprogram with a two letter call which is not listed in the Statistics manuals, the TI subprogram. I know a subprogram is there since a call does not yield a "Not Found" response, but I have not yet identified proper arguments for the call. There may also be three letter calls which are not listed in the manuals. Hopefully, we can find a subprogram which will provide for input and edit of a one-dimensional array in a manner similar to the AK subprogram in the Mathematics cartridge.

<u>AREA FINDING</u> - The September 1986 issue of Computer Shopper includes an article entitled "The Universal Area Calculator" by Frank Tymon. A BASIC program was provided for finding the area and perimeter of a polygon defined by its vertices. The method of solution for the area of the polygon is the same as that used in Henrik Klein's polygon program for the TI-58/59 in V8N2P7. Tymon notes that the area calculation will yield a positive answer if the vertices are taken in a counter-clockwise order, and a negative answer if the vertices are taken in a clockwise order. He fails to recognize the capability to traverse the outer polygon and use a "cut" to go from an outer polygon to an inner polygon, followed by a clockwise traverse of the inner polygon to find the area between two polygons. Perhaps he avoided that idea because of the difficulty keeping the "cut" from being included in the perimeter. A CC-40 or TI-74 program which uses the subprograms in the Statistics cartridge to solve for the area is at the right.

```
10 DIM X(31),Y(31)
15 CALL UP("Area Finder",PN)
20 INPUT "Number of Vertices? ";N
25 CALL IC(3,30,N,Q)
30 IF Q=-1 THEN PAUSE:GOTO 20
35 CALL AU("X","Y",X(),Y(),1,N,PN,0,
0)
40 X(N+1)=X(1):Y(N+1)=Y(1)
45 S=0
50 FOR I=1 TO N
55 S=S+X(I)*Y(I+1)-X(I+1)*Y(I)
60 NEXT I
65 PRINT #PN,"Area = ";S/2
70 PAUSE
75 END
```

The prompts provide the instructions for running the program. When making a "cut" it is necessary to enter the transition points between the inner and outer curves both going and returning.

Line 25 uses the IC subprogram to ensure that the number of vertices is at least three and not more than 30. Line 30 illustrates the use of the flag set by the IC routine to branch back to the input if the entered number of vertices was not acceptable.

This program will also operate with the Mathematics cartridge and the CC-40 by deleting line 25 and the last two zeroes in the argument in line 35.

MANY DIGITS OF LN(2) AND LN(3) IN BASIC - One of the programs in Robert Prins' treatise on extended precision calculations for the TI-59 (see page 25) can calculate natural logarithms up to 1188 digits. The problem was to find an independent listing to verify that the program was correct. I was able to find fifty digits of ln(2), ln(3) and ln(10) on page A-1 of the CRC Handbook of Chemistry and Physics, but was unable to find any longer listing. Robert and I decided to independently calculate 1200 digits using other programs and compare our results. My program used the well known relationship

$$\ln\frac{1+x}{1-x} = 2\left[x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \ldots\right]$$

which is also the basis for Robert's program for the TI-59. For $x = 1/3$ that series reduces to

$$\ln(2) = 2\left[\frac{1}{3} + \frac{1}{3\cdot 3^3} + \frac{1}{5\cdot 3^5} + \frac{1}{7\cdot 3^7} + \frac{1}{9\cdot 3^9} \ldots\right]$$

I used a brute force approach since memory limitations weren't a problem with my CC-40. The listing for the program that I used on the CC-40 is at the right.

```
100 DIM A(120),B(120),C(120)
110 IMAGE ##########
120 CALL UP("Logarithm of 2",Z)
130 S=1.E+10
140 INPUT "Number of 10 Digit Blocks
 ? ":N
150 FOR I=1 TO N
160 A(I)=6666666666
170 NEXT I
300 I=1
400 R=0
410 FOR J=1 TO N
420 B(J)=A(J)+R*S
430 R=B(J)-I*INT(B(J)/I)
440 B(J)=INT(B(J)/I)
450 NEXT J
500 M=(I+1)/4
510 R=2*(M-INT(M))
520 FOR J=N TO 1 STEP -1
530 C(J)=C(J)+B(J)+R
540 R=INT(C(J)/S)
550 C(J)=C(J)-R*S
560 NEXT J
600 R=0
610 FOR J=1 TO N
620 T=INT((A(J)+R*S)/9)
630 R=A(J)+R*S-9*T
640 A(J)=T
650 NEXT J
700 IF B(N)<>0 THEN I=I+2:GOTO 400
800 PRINT #Z,10*N;" Digits"
810 PRINT #Z
820 FOR J=1 TO N
830 PRINT #Z,USING 110,C(J)
840 IF Z=0 THEN PAUSE
850 NEXT J
900 IF Z=1 THEN CLOSE #1
999 END
```

Line 100 sets up 3 extended precision registers (arrays) where ten digits to the left of the decimal point are maintained in each of the 120 elements of each array for a total of 1200 digits. The digits to the right of the decimal point are used to carry from one element to the next.

Line 120 assumes the user has one of the cartridges installed.

Line 140 allows the user to select how many digits to calculate. Using only the number of digits needed saves calculation time.

Lines 150-170 load the A array with the value 2/3, the first term of the series. Lines 400-450 divides the A array by 2n-1 and accumulates the result in the B array.

Lines 500-510 provide automatic compensation for truncation effects by adding 1 at the least significant digit every other cycle, for an average rate of 0.5 of the least significant digit per cycle which will be approximately correct if the truncation errors are uniformly distributed. Without this correction a series with all positive terms will generate an error equal to one-half of the number of terms times the least significant digit.

Lines 520-560 complete the addition of array B to array C to accumulate the sum.

Many Digits of Ln(2) and Ln(3) - (cont)

Lines 600-650 divide the contents of the A array by nine (3 squared) to prepare for the next iteration.

Line 700 keeps the routine iterating until there are no more corrections to array C. This is easier than trying to figure out how many cycles to use to get a given number of digits.

Lines 800-840 provide printout of the result once the calculations are complete.

To calculate ln(3) you only need to note that x should be 1/2 in the beginning formula. In the program you change line 160 to 9999999999 (ten nines). You do not start with a 1 and ten zeroes in A(1) and zeroes in the rest of the elements of A as you might think to avoid termination after the first term by line 700.

The listings at the right are for 1000 digits of ln(2) and ln(3). Robert and I agree on ln(2) but he has not yet completed ln(3). The first 50 digits of ln(3) agree with the CRC values.

MORE PERIPHERALS FOR THE TI-74? - P. Hanson

Last week I was contacted by a telephone survey performed for TI. Purchasers of the TI-74 and the TI-95 were being contacted. The questions suggest that several new peripherals may become available for those devices:

* An interface which would allow the connection of the AC-9201 without the use of the PC-324.
* A Centronics interface for use with printers and plotters.
* An RS-232 interface.
* A combined Centronics and RS-232 interface.

One of the purposes of the survey was to determine what would be acceptable prices. The caller asked what I thought might be a price at which I would probably buy, and then asked for the likelihood that I would buy at various prices. We will have to wait and see.

Logarithm of 2	Logarithm of 3
1000 Digits	1000 Digits
6931471805	986122886
5994530941	6810969139
7232121458	5245236922
1765680755	5257046474
13436025	9055782274
5254120680	9451734694
94933936	3336374942
2196969471	9321860896
5605863326	6873615754
9964186875	8137320887
4200148102	8797002906
570685733	5957865742
6855202357	3680042259
5813055703	3051982105
2670751635	2801870767
759619307	2774100831
2757082837	6270918338
1435190307	1307179373
386238916	0988443609
7347112335	5988374257
115364497	316795911
9552391204	5211455919
7517268157	1775867134
4932065155	7054940160
5247341395	7755002222
2588295045	317025294
3007095326	0897500690
3666426541	1005215856
423915781	4200813003
4952043740	6317373298
4303855008	5777023009
194417064	9105479213
1671518644	1010149020
7128399681	301030236
7178454695	3012224065
7026271631	2748190225
645461502	9918374524
5720740248	9889045805
1637773389	3407000045
6385506952	9058857484
6066834113	4411901885
7273873722	7087047494
9289564935	0070790130
4702576265	0582941100
2098859693	2100121184
2019650585	14090255
5476470330	1439194070
6793654432	8093079849
5476327449	4302255731
5125040606	5353290853
9438147104	4529525145
6899465062	9213070494
2016772042	0059325027
4524529612	9441053094
6879465461	1570272310
9316517468	3551000001
1392672504	211040989
1038025462	439943063
5965686914	1302552857
4192871608	3040080282
2938031727	4908130022
1436778265	8000341439
4877566485	1078089325
856740776	1450437510
4845146443	2944510275
9940461422	6153497398
6031930967	2110941585
3540257444	7490353017
6070308096	5890897512
850474866	2233797730
3852313818	9090877543
1676751438	5475513571
6674766478	2902177817
9088143714	5812421223
1985494231	5148581016
5199735488	3272465588
375165861	9372495049
2753529166	1910524290
1000710535	790084234
5824987941	0470093772
4729509293	3725205508
1138971559	2032078333
9820565439	9290558928
2871700072	5314087309
1808576102	5132086458
5230689213	3891043974
2449713893	9082223832
2037843935	5705407533
3088774825	3110230196
9701715591	4827525759
708823683	9132217851
6275898425	3533902374
8918535302	8290433958
4363421436	2540874245
7061189236	8243340008
7891923723	0012100143
1467232172	0520505429
534016492	5427070185
5687274778	547779542
2344535353	2933973343

USE OF THE AC-9201 WITH THE TI-74 AND PC-324 - V11N2P17 noted that the Educalc catalog listed the AC-9201 as TI-74 hardware, but that my TI-74 did not have an adapter socket. A statement on page 5 of the manual for the PC-324 provides an explanation:

> "You can use the optional AC9201 adapter to operate the printer on standard line voltage. When connected to the printer, the adapter provides power for both the printer and the calculator."

Tests also show that a TI-74 without batteries will run when connected to a PC-324 with batteries.

If you remove the batteries from a standalone TI-74 for about a minute, and then reinstall them you will receive the message "W30 Initialized" at turnon. Among other things, that message means that you lost any programs which may have been in memory. Page 1-5 of the TI-74 User's Guide warns against that condition with the statement:

> "The Constant Memory (TM) feature retains stored information for a short time after the batteries are removed. As a precaution, however, you may want to save any important progrms and data on a storage device (such as a cassette) before replacing batteries."

The availability of power for the TI-74 through the printer connection, whether from the batteries in the printer or the AC-9201, allows the user to circumvent the potential problem of loss of memory when changing batteries in the TI-74. All you need to do is have the printer connected, with batteries installed or the AC-9201 connected, during the battery change in the TI-74.

DEVICE CODE FOR THE PC-324 - The instructions for accessing the printer which appear in the manuals for the Statistics and Mathematics cartridges for the TI-74 state that you can find the identification number (device code) in the printer operating manual. I couldn't find the number in the manual for the PC-324. I experimented and found that the number 12 would work. Later, I found that page 3-19 of the TI-74 Programming Reference Guide indicates that the device code for the printer is indeed 12.

DISCUSSION OF THE TI-59 FIRMWARE - Page 7 of this issue discusses the firmware mechanization of the DMS function in the TI-59., Newcomers who are interested in more details on the firmware in the TI-59 should send four dollars for a twenty page discussion on subjects such as how to view the firmware from the keyboard, the mechanizations for all of the statistics and conversions functions, the HIR 20 function, and identification of the constants used in the log and trig functions.

TI- 74 AVAILABILITY - TI-74's became available at the local Service Merchandise outlets in October 1986. Only the computer was available at a price of $99.97 plus tax. No peripherals so far.

Extra-precision Combinations on the TI-59 - (cont)

Steps 5 through 29 provide for initialization and data entry. The GTO 472 at 031-033 picks a jump address for fast mode entry at step 001 for the 13 digit calculation. Steps 60 to 100 are the 13 digit calculation which determines if the answer is less than 38 digits long. The GTO 469 at 036-038 picks up a different jump address

$$8*(WXY) + Z + 1 = 8*12 + 4 + 1 = 101$$

where W through Z are the ninth through twelfth digits. Steps 101 through 248 are the extra-precision calculation. Steps 39 through 59 display the result of the precision calculation.

For N = 100 and R = 70 the 13 digit answer 2.937234e25 will be returned in about 13 seconds. The extra precision solution requires about 340 seconds, and is read out by Mode E as 0 0 29372339 821610944 823963760.

EXTRA-PRECISION COMBINATIONS IN BASIC - L. Leeds

Larry also provided a Model 100 BASIC language equivalent of the TI-59 program on the preceding page. The program at the right is the editor's conversion for use on the CC-40 or TI-74.

```
100 Z=1.E+10
110 CALL UP("nCr = N!/(R!(N-R)!)",PN
)
120 IMAGE ##########
130 INPUT "Number of Things? ";N
135 PRINT #PN,"n = ";N
140 INPUT "How many at a time? ";R
145 PRINT #PN,"r = ";R
150 B=N:C=1
160 IF R>N/2 THEN R=N-R
170 FOR K=1 TO R
180 C=C*B/K:B=B-1:NEXT K
190 PRINT #PN,"nCr = ";C
195 IF PN=0 THEN PAUSE ELSE PRINT #P
N
200 INPUT "Need more precision [Y/N]
? ";A$
230 IF A$="n"OR A$="N"THEN 130
235 FOR I=0 TO 4:A(I)=0:NEXT I:A(4)=
1
240 FOR H=1 TO R
250 FOR J=0 TO 4
260 A(J)=A(J)*N:NEXT J
270 FOR J=4 TO 1 STEP -1
280 T=A(J)/Z:A(J-1)=A(J-1)+INT(T)
290 A(J)=(T-INT(T))*Z:NEXT J
300 FOR J=0 TO 4
310 B=A(J)/H:Q(J)=INT(B)
320 A(J+1)=A(J+1)+(A(J)-Q(J)*H)*Z
330 NEXT J
340 FOR J=0 TO 4
350 A(J)=Q(J):NEXT J
360 N=N-1:NEXT H
370 FOR I=0 TO 4
380 PRINT #PN,USING 120;Q(I)
390 IF PN=0 THEN PAUSE
400 NEXT I
410 PRINT #PN
420 GOTO 130
430 END
```

Line 110 presumes that a software module is installed so that the UP subprogram can be used to print a title, set a branching value (PN) to select use of a printer or the display for output, and open a file for access to the printer if needed.

Use of the ten digit IMAGE definition at line 120 provides right justification of the output for the extra-precision answer.

Lines 150 through 195 are the single-precision solution.

Lines 200 through 360 are the extended-precison solution.

Lines 370 through 410 provide output of the extra-precision solution.

The program capability is 41 digits. The printout illustrates the solution for two problems. The printout and listing were made with a CC-40/HX-1000 combination in the 36 character per line mode and magnified for readability.

```
n =  100
r =  70
nCr =  2.937234E+25

         0
         0
    293723
3982161094
4823963760
```

```
n =  140
r =  70
nCr =  9.382097E+40

         9
3820909097
8490412047
8589458058
6297000000
```

A listing and printout using a TI-74 with a PC-324 appears on page 18. That program also illustrates the mechanization of an option between output to the printer or to the display with out using the Call UP idea.

EXTRA-PRECISION COMBINATIONS ON THE TI-74 & PC-324

Page 5 presented a conversion of a BASIC program by Larry Leeds for use with the CC-40. The program assumed that a software module was installed so that the UP subprogram could be used to print a title, to set a branching value to select use of either the printer or the display for output, and to open a file for access to the printer when needed. That program would also run on a TI-74 if one of the software modules was installed.

The program at the right does not assume that a software module is installed. Thus the functions which might have been provided by the UP subprogram must be provided in the user's program. The changes required are in lines 110 through 117 where:

Line 110 provides a prompt for selection of the printer or the display for output.

Line 113 processes the user response to set the PN flag as required.

Line 115 provides a prompt for entry of the device number if the printer option was selected. For the TI-74 operating with the PC-324 the user response is 12. For the CC-40 operating with the HX-1000 the response may be either a 10 or a 11 depending upon the setting of the switch in the HX-1000.

Line 117 opens a file for access to the printer if the printer option was selected.

The program listing was made using the normal printing mode of the PC-324, or 24 characters per line. Note the substantially improved legibility relative to the listing on page 5 which was made using the 36 character per line mode of the HX-1000 and then magnified.

Sample printouts appear below. Again, note the improved legibility over that on page 5.

```
n =  140
r =  70
nCr =  9.382097E+40

         9
3820969697
8400412047
8589458050
6297666600
```

```
n =  100
r =  70
nCr =  2.937234E+25

         0
         0
    293723
3982161094
4823963760
```

```
100 Z=1.E+10
110 INPUT "Use Printer <Y/N>? ";A$
113 IF A$="Y"OR A$="y"THEN PN=1
115 IF PN=1 THEN INPUT "Enter device name: ";F$
117 IF PN=1 THEN OPEN #PN,F$,OUTPUT
120 IMAGE ##########
130 INPUT "Number of Things? ";N
135 PRINT #PN,"n = ";N
140 INPUT "How many at a time? ";R
145 PRINT #PN,"r = ";R
150 B=N:C=1
160 IF R>N/2 THEN R=N-R
170 FOR K=1 TO R
180 C=C*B/K:B=B-1:NEXT K
190 PRINT #PN,"nCr = ";C
195 IF PN=0 THEN PAUSE ELSE PRINT #PN
200 INPUT "Need more precision <Y/N>? ";A$
230 IF A$="n"OR A$="N"THEN 130
235 FOR I=0 TO 4:A(I)=0:NEXT I:A(4)=1
240 FOR H=1 TO R
250 FOR J=0 TO 4
260 A(J)=A(J)*N:NEXT J
270 FOR J=4 TO 1 STEP -1
280 T=A(J)/Z:A(J-1)=A(J-1)+INT(T)
290 A(J)=(T-INT(T))*Z:NEXT J
300 FOR J=0 TO 4
310 B=A(J)/H:Q(J)=INT(B)
320 A(J+1)=A(J+1)+(A(J)-Q(J)*H)*Z
330 NEXT J
340 FOR J=0 TO 4
350 A(J)=Q(J):NEXT J
360 N=N-1:NEXT H
370 FOR I=0 TO 4
380 PRINT #PN,USING 120;Q(I)
390 IF PN=0 THEN PAUSE
400 NEXT I
410 PRINT #PN
420 GOTO 130
430 END
```

LOAN SCHEDULE WITH THE FINANCE MODULE OF THE CC-40

Earlier issues have presented programs for loan schedules for the TI-59; for example, see V9N1P20 and V11N2P14. V10N1P9 also presented a loan schedule for the CC-40 used with the RS-232 interface and a full size printer. During search of the Finance module of the CC-40 for non-published subprograms I found that the Money Evaluator program can print out a loan schedule as well. A sample printer output appears at the right. To obtain this printout install the Finance module in the CC-40, connect the HX-1000 Printer/Plotter, enter RUN "MEVAL" in the display and press ENTER. The following sequence shows the displayed prompts followed by the responses in brackets:

1. Use Printer? <y>
2. Enter Device Name: <10.s=0>. The 36 character per line printer option is selected and the heading "MONEY EVALUATOR" is printed.
3. Nominal Interest? <y>. The heading "Nominal Interest is printed.
4. Enter Compounding Prds/Yr: <12> . The prompt and entry are printed.
5. Enter # Pmts/Yr: <12> . The prompt and entry are printed, followed by a menu of options.
6. Enter Compute Option: <1> , where we select the Payment option (1) to calculate the payment per period.
7. Enter # Payments: <9> . With Matteson's TI-59 program on V11N2P14 we had used 0.75 years. "# Payments= 9" is printed.
8. Enter %Interest: <12.5> , and "%Interest= 12.5" is printed.
9. Enter Pres Val: <1000> , and "Pres Val= 1000" is printed.
10. Enter Future Val: <0> , and "Future Val= 0" is printed.
11. End of Period Payments: <y> , and the prompt is printed. If the user response had been <n> then the prompt "Beginning of Period Payments" would have been displayed to permit use of an alternative method of payments.
12. Discount Backward? <y> , and the prompt is printed. I don't really understand this; perhaps some member with more background in accounting can enlighten me.
13. Edit? <n> , where if we had answered <y> we would have an opportunity to check and change if we desired all the responses since the menu. Since we had selected Option 1, the monthly payment of $116.98 is calculated and printed.
14. Amortize? <y> , and the message "Annual Debt Payment= 1403.76" is printed. That is the monthly payment multiplied by 12, which has very little to do with our nine month problem, but I don't know how to suppress it.
15. Subtotals? <n> , where the lower example at the right shows a printout for a response of <y>.
16. Cumulative Totals? <n> .
17. Enter First Payment #: <1> .
18. Enter Last Payment #: <9> , and the schedule is printed as shown at the right.

```
MONEY EVALUATOR

Nominal Interest
Compounding Prds/Yr= 12
# Pmts/Yr= 12

0=Menu 1=Payment 2=# Payments
3=Interest     4=Present Value
5=Future Value 6=Amortize


# Payments= 9
%Interest= 12.5
Pres Val= 1000
Future Val= 0

End of Period Payments

Discount Backward

Payment=       116.98
Annual Debt Payment=  1403.76

     **** Payment 1 ****
Principal Payment=      106.56
Interest Payment=        10.42
Balance=                893.44

     **** Payment 2 ****
Principal Payment=      107.67
Interest Payment=         9.31
Balance=                785.77

     **** Payment 3 ****
Principal Payment=      108.79
Interest Payment=         8.19
Balance=                676.98

     **** Payment 4 ****
Principal Payment=      109.93
Interest Payment=         7.05
Balance=                567.05

     **** Payment 5 ****
Principal Payment=      111.07
Interest Payment=         5.91
Balance=                455.98

     **** Payment 6 ****
Principal Payment=      112.23
Interest Payment=         4.75
Balance=                343.75

     **** Payment 7 ****
Principal Payment=      113.40
Interest Payment=         3.58
Balance=                230.35

     **** Payment 8 ****
Principal Payment=      114.58
Interest Payment=         2.40
Balance=                115.77

     **** Payment 9 ****
Final Payment=          116.98
Principal Payment=      115.77
Interest Payment=         1.21
Balance=                   .00


# Payments= 9
%Interest= 12.5
Pres Val= 1000
Future Val= 0

End of Period Payments

Discount Backward

Payment=       116.98
Annual Debt Payment=  1403.76

     **** Payment 5 ****
Subtotal Principal=     544.02
Subtotal Interest=       40.88
Total Principal=        544.02
Total Interest=          40.88
Balance=                455.98

     **** Payment 9 ****
Final Payment=          116.98
Subtotal Principal=     455.98
Subtotal Interest=       11.94
Total Principal=       1000.00
Total Interest=          52.82
Balance=                   .00
```

Loan Schedule with the Finance Module of the CC-40 - (cont)

In step 2 it was necessary to respond with <10.a=0> to obtain the 36 character per line printer option and get the nicely formatted printout shown on page 6. If the simpler response <10> is used the default condition of this program sets the 18 character per line printer option. The printout at the right illustrates the effect of the 18 character option on the printout. Clearly, the programmer of the Money Evaluator assumed the 36 character per line option, but failed to provide for it as a default option.

A different condition applies with the Prime Factors program (RUN "PRI") in the Mathematics module as discussed in V8N4P12. There the 36 character printer option is selected by the program, a the response to the prompt "Enter Device Name:" of <10.a=1> does not change the printer to the 36 character per line mode.

Similarly, the UP subprogram in the various modules will set the printer to the 36 character per line option whether or not you add the a=0 part to the device name.

```
End of Period Paym
ents

Discount Backward

Payment=       118
.98
Annual Debt Paymen
t=  1403.76

     **** Payment
1 ****
Principal Payment=
      108.56
Interest Payment=
       10.42
Balance=
      893.44
```

THE USE OF DMS TO TRUNCATE THE DISPLAY REGISTER TO THE DISPLAY - P. Hanson.

Step 4 of the instructions for the loan schedule program on V11N2P14 noted that fractional years could be entered. An example was given using 0.75 years. It is not surprising that when the program multiplies that value by 12 to yield the number of months that the answer is acceptable. But what about other fractions. If you wanted seven months you might enter 7 divided by 12. When the program multiplies that result by 12 the display is "7.", but the display register contains 7 - 1e-12. The use of an integer function as at step 336 on V11N2P15 will yield 6 instead of the desired 7. An EE-INV-EE sequence before the integer function would place the display value in the display register as noted on page C-1 of Personal Programming, but examination of the program doesn't reveal such a sequence. The equivalent result is obtained with the DMS at step 323, where page V-30 of Personal Programming notes that the DMS conversion acts only on the displayed value.

We know that the DMS function is mechanized as shown at the right. The curious feature of the mechanization is that the truncation to the display feature of the DMS function is cannot be determined from the listing. There is no EE-INV-EE sequence to be found. If you place 7 - 1e-12 (7 / 12 x 12 =) in the display register and run the routine at the right from user memory the displayed answer will be 7.677777778 which is derived as follows:

6.	degrees converts to	6.
99	minutes converts to	1.65
99	seconds converts to	.0275
.99999999	seconds converts to	.000277777775
	which sums to	7.677777777775
	for a display of	7.777777778

000	76	LBL	020	00	0
001	11	A	021	54	)
002	53	(	022	82	HIR
003	53	(	023	08	08
004	53	(	024	59	INT
005	82	HIR	025	54	)
006	08	08	026	65	×
007	59	INT	027	93	.
008	65	×	028	06	6
009	06	6	029	85	+
010	00	0	030	82	HIR
011	85	+	031	18	18
012	53	(	032	22	INV
013	82	HIR	033	59	INT
014	18	18	034	54	)
015	22	INV	035	55	÷
016	59	INT	036	03	3
017	65	×	037	06	6
018	01	1	038	54	)
019	00	0	039	92	RTN

MATS IS DIFFERENT IN THE MATH MODULES OF THE CC-40 AND THE TI-74

V11N2P19 reported that the program package of the Mathematica module for TI-74 was very similar to that of the Mathematica module for the CC-40. Two discrepancies noted with the CC-40 Mathematica module had been corrected. I had determined that the manuals were very similar, and that the calls to the subprograms were the same.

V8N5P14-17 presented a detailed look at the MATS routines in the CC-40 Mathematica module. A demonstration least squares polynomial curve fit was included. When I tried to convert programs which used the MATS subprogram from the CC-40 to the TI-74 I encountered difficulty. Investigations revealed that although the calls are the same, and the numerical results are the same, there are important differences in the way the subprograms are mechanized. There is even an error in the definition of the test output which tells if the matrix is singular. The differences I have identified so far are:

1. The solution for linear equations appeared in the first column of array1 in the CC-40. It appears in the first row of array1 in the TI-74. Therefore, the conversion of the linear equations program on V8N5P14 for the TI-74 requires that line 420 be changed from

```
PRINT X$;A(I,1):PAUSE    to  PRINT X$;A(1,I):PAUSE
```

Similarly, the subscripts for A must be reversed in lines 320 and 380 of the curve fit program on V8N5P17.

2. The last argument of the MATS call is the "test" value. The definition of "test" in the manuals for the CC-40 and TI-74 is that it passes in a dummy variable, and returns a zero if the matrix is singular. In the CC-40 the MATS program would leave the value of "test" as it was if the matrix was not singular, and would change the value of "test" to zero if the matrix was singular. Thus, to differentiate between the two it was necessary to set the value of test to a non-zero value prior to calling MATS. That is the reason for the R = PI at line 110 of the linear equations program on V8N5P14, and for the R = 1 in line 150 of the polynomial curve fit program on V8N5P17, where R is the "test" variable name used in those programs. Any old non-zero value will do.

In the TI-74 mechanization of MATS the value of "test" is set to -1 if the matrix is singular, and is set to zero if the matrix is not singular. There is no need to pre-load a value for "test" prior to the call of MATS. The definition of "test" on page A-11 of the manual is incorrect.

3. The CC-40 mechanization of the matrix inversion routine (option 4) returned the inverse of an input matrix to array1 but with some columns interchanged, and the inverse of the input matrix with the columns in proper order to array2. In the TI-74 the inverse of the input matrix is returned to array1 with the columns in proper order. The contents of array2 are not disturbed.

4. V8N5P14 noted that for some reason the dimension statement must provide array1 with one additional column if a "Bad Subscript" error is to be avoided when running the linear equations solution on the CC-40. This idiosyncrasy does not appear in the TI-74.

LEAST SQUARES REGRESSION WITH USER DEFINED FUNCTIONS - P. Hanson

Discussions of least squares regression, at least in TI PC Notes, have typically been limited to the use of polynomials. Examples include Thomas Wysmuller's article on polynomial regression in the November/December issue of PPX Exchange, Gene Friel's polynomial regression with variance (V9N2P20), and the least squares polynomial curve fit for the CC-40 from V8N5P17. There are many applications where the appropriate functions are not polynomials. One example is the solution for sin and (1- cos) functions for inertial navigation error analysis which appeared in V10N4P7. In the past I have also found it necessary to regress using functions such as $\sqrt{x}$, $|x|$, and the like, and have typpically written special programs for each application. To minimize such effort in the future I decided to structure a program such that a subroutine can be changed to define the regression functions, but the remainder of the program is independent of the regression functions. The resulting TI-74 program which follows requires that the Mathematics software module be installed. The program provides the necessary prompts so that separate user instructions are not required.

Line 110 is the printer useage subprogram. If the user responds "N" to the prompt "Use Printer?" then variable PN is set to zero. If the user responds with "Y" then PN is set to one, and an additional prompt "Enter File Name:" appears. For the TI-74 the response is 12. The text inside the quotation marks in the call is printed and displayed.

Line 120 displays a message for two seconds to remind the user that he must set the regression functions properly.

Line 130 asks for the number of data pairs, a value which is required by the AU subprogram which follows.

Line 140 uses the AU subprogram to enter and edit the two one dimensional arrays for the independent (x) and dependent (y) variables. The printer option set by line 110 provides annotation on one line followed by the input value on the next line and spaces between the pairs of input values (see the sample printout on page 15).

Line 150 asks for the order of the least squares solution.

Lines 160-180 clear the matrix and vector which are used in the least squares solution. These values will have been set to zero at program entry with a RUN command. The values must be reset to zero to permit multiple passes through the program without reentry with a RUN command, say with edited data or a different order (See lines 700-740 below).

```
100 DIM A(8,8),B(8),C(8,8),F(8),X(50),Y(50)
110 CALL UP("Least Squares Fit",PN)
120 PRINT "Are the functions correct?":PAUSE 2
130 INPUT "Number of Data Pairs? ";K
140 CALL AU("X","Y",X(),Y(),1,K,PN)
150 INPUT "Order of the solution? ";N
160 FOR I=1 TO N:FOR J=1 TO N
170 A(I,J)=0:NEXT J
180 B(I)=0:F(I)=0:NEXT I
200 FOR L=1 TO K
210 GOSUB 800
300 FOR I=1 TO N:FOR J=1 TO N
310 A(I,J)=A(I,J)+F(I)*F(J):NEXT J
320 B(I)=B(I)+F(I)*Y(L):NEXT I
330 NEXT L
340 CALL MATS(A(,),C(,),B(),1,1,5,1,N,1,R)
350 IF R=-1 THEN PRINT "Matrix is Singular":PAUSE
360 PRINT #PN
400 FOR I=1 TO N
410 X$="A"&STR$(I)&" = "
420 PRINT #PN,X$;A(1,I)
430 IF PN=0 THEN PAUSE
440 NEXT I
450 PRINT #PN
500 INPUT "Display Residuals (Y/N)? ";A$
510 S1=0:S2=0
```

Least Squares Regression - (cont)

Lines 200-330 calculate the matrix and vector values needed for the least squares solution. The GOSUB 800 at line 210 calls the subroutine which provides user defined functions.

Line 340 calls the MATS subprogram to solve the linear equations generated in lines 200-330.

Line 350 provides an indication to the operator if the matrix is singular based on the "test" output from the MATS subprogram.

Lines 400-450 provide printout or display of the solution with annotation by recalling values from the first row of the A matrix, array1 in the MATS subprogram call.

Lines 500-600 permit printout or display of the residual errors with annotation. Lines 570-590 provide the annotation and the associated residual on the same line, a technique that we would have preferred with the AU call in line 140.

Line 610 accumulates the sums which are needed for calculation of the mean error and the standard error.

Lines 620-680 calculate the mean error and the standard error, and display or print the results with annotation.

Lines 700-740 provide options for editing of bad input data as determined from the residuals, or for selection of a different order of solution.

Lines 800 through 890 are the subroutine which provides the user defined functions. The function shown is for a polynomial curve fit.

```
520 FOR L=1 TO K
530 GOSUB 800
540 YF=0:FOR J=1 TO N
550 YF=YF+A(1,J)*F(J):NE
XT J
560 D=Y(L)-YF
570 IF A$="N"OR A$="n"TH
EN 610
580 P$="d"&STR$(L)&" = "
590 PRINT #PN,P$;D
600 IF PN=0 THEN PAUSE
610 S1=S1+D:S2=S2+D*D:NE
XT L
620 PRINT #PN
630 PRINT #PN,"Mean Erro
r = ";S1/L
640 IF PN=0 THEN PAUSE
650 PRINT #PN
660 PRINT #PN,"Standard
Error = ";SQR(S2/(K-N))
670 IF PN=0 THEN PAUSE
680 PRINT #PN
700 INPUT "Edit Input Da
ta (Y/N)? ";A$
710 IF A$="N"OR A$="n"TH
EN 730
720 CALL AU("X","Y",X(),
Y(),1,K,PN)
730 INPUT "Different ord
er (Y/N)? ";A$
740 IF A$="Y"OR A$="y"TH
EN 150
799 STOP
800 REM USER DEFINED FUN
CTIONS
810 F(1)=1
820 FOR W=2 TO N
830 F(W)=F(W-1)*X(L)
840 NEXT W
890 RETURN
```

The polynomial generating function in steps 800 through 890 of the baseline program can be replaced by other user-defined functions. The center column on page 14 illustrates the use of sin and (1 - cos) functions to permit solution of a problem such as that presented on V10N4P7. Line 810 scales the equal time increment x values (every six minutes for the problem shown) to angles by dividing by the Schuler frequency (84 minutes) and multiplying by 360 degrees. This problem is not the same as fitting a function A + Bcos(w) +Csin(w); the lack of an independent constant term leads to the non-zero mean for the residuals.

Now consider the problem of finding Fourier series coefficients of a periodic function. An example problem is provided with the MU-19 Discrete Fourier Series program in the manual for the Math/Utilities module for the TI-59. The twelve function values corresponding to twelve angles equally spaced over the range from 30 to 360 degrees are: 2.0, 3.2, 3.7, 2.5, 1.2, 1.5, 2.7, 2.4, 0, -3.3, -3.3, and 0.

Least Squares Regression - (cont)

The regression functions defined in the right hand column below permit the user to use this program to find the coefficients through the third harmonic. The multiplication of the X(L) value by 30 in line 810 allows the user to enter the sequence numbers (1 through 12) associated with each function value instead of entering the angles. Once the data entry is completed the solution appears after about one minute. The answers are in agreement with those on page 73 of the Math/Utilities manual if the user remembers that the A0 coefficient from the MU-19 program is twice the A1 coefficient from this program. Of course, the real power of using a least squares program is that there is no requirement for equally spaced data over the fundamental frequency range as with a discrete Fourier analysis.

```
Least Squares Fit

X(1)=
-7
Y(1)=
-81

X(2)=
-4
Y(2)=
 27

X(3)=
 0
Y(3)=
 3

X(4)=
 2
Y(4)=
 27

X(5)=
 5
Y(5)=
 243


A1 =  18.61841343
A2 =  26.3477128
A3 =  2.460625362

d1 = -35.75506659
d2 =  74.40243196
d3 = -15.61841343
d4 = -54.15634047
d5 =  31.12738854

Mean Error =
 3.166667E-12

Standard Error =
 74.02670139
```

```
800 REM USER DEFINED FUN
CTIONS
810 W=360*X(L)/84
820 F(1)=SIN(W)
830 F(2)=1-COS(W)
890 RETURN
```

```
Least Squares Fit

X(1)=
 6
Y(1)=
 1

X(2)=
 12
Y(2)=
 2

X(3)=
 18
Y(3)=
 3

X(4)=
 24
Y(4)=
 4

X(5)=
 30
Y(5)=
 5

X(6)=
 36
Y(6)=
 6


A1 =  .8934012668
A2 =  2.786304468

Mean Error =
 .0525609272

Standard Error =
 .3178704425
```

```
800 REM USER DEFINED FUN
CTIONS
810 W=30*X(L):F(1)=1
820 F(2)=COS(W)
830 F(3)=SIN(W)
840 F(4)=COS(2*W)
850 F(5)=SIN(2*W)
860 F(6)=COS(3*W)
870 F(7)=SIN(3*W)
890 RETURN
```

```
Least Squares Fit

X(1)=
 1
Y(1)=
 2

X(2)=
 2
Y(2)=
 3.2

X(3)=
 3
Y(3)=
 3.7

X(
 1)=
-3.3

X(12)=
 12
Y(12)=
 0


A1 =  1.05
A2 = -1.417222017
A3 =  1.885961277
A4 = -.55
A5 =  1.905255888
A6 =  .5833333333
A7 =  .0166666667
```

Least Squares Regression - (cont)

The user defined functions must be designated by F(1) up to F(8). Each pair of X,Y values are available in turn at the entry to the subroutine. To obtain a constant in the solution define one of the functions as one, as in line 810 of the program on page 13.

To modify the program on pages 12 and 13 for use with the Mathematica module installed in the CC-40:

* In line 100 change A(8,8) to A(8,9) if you wish to solve eighth order systems.
* Add line 250: 250 R = 1
* In line 350 change IF R=-1 to IF R=0
* In line 420 change A(1,I) to A(I,1)
* In line 550 change A(1,J) to A(J,1)

MORE ON NUMERIC REPRESENTATION - In his column "The Art of Programming" in the March 1987 issue of 80 Micro Bruce Tonkin discusses the effects of digital representation in a computer and proposes a sample problem to illustrate the effects. His sample program is similar to the one at the right. A user who repetitively enters the value 1.11 will find the following display of T for various machines:

```
10 Input X
20 T = T + X
30 PRINT T
40 GOTO 10
```

Commodore 64	TI-74/CC-40
1.11	1.11
2.22	2.22
3.33	3.33
4.44	4.44
5.55	5.55
6.66000001	6.66
7.77000001	7.77
8.88000001	8.88
9.99000001	9.99
11.1	11.1
12.21	12.21

The Radio Shack Color Computer yields the same output as the Commodore 64. The Radio Shack Model 100 yields the same output as the TI-74. Where does the "garbage" come from with the CC-64 and Color Computer? Tonkin explains that the effect is due to the digital representation in those machines, and goes on to explain that the effect would be eliminated if numbers were stored in BCD (base 10). That is exactly what the Model 100 does. The TI-74 and CC-40 actually use base 100. How can a user be sure which base is used another machine? The best way is to use the "Itsy Bit of Paranoia" from the February 1985 issue of BYTE. See V10N2P16 for a description of the test.

IT'S CALL SQ(R,I) NOT CALL SQR(R,I) FOR THE CC-40 & TI-74 MATH MODULES

Page 103 of the manual for Mathematics Cartridge for the CC-40 shows that CALL SQR(REAL,IMAGINARY) allows the user to pass in the real and imaginary parts of a complex number and obtain the real and imaginary parts of the square of the input. Page A-19 of the manual for the Mathematics cartridge for the TI-74 says the same. However, if you write a short demonstration program:

```
10 INPUT A,B        20 CALL SQR(A,B)        30 PRINT A,B:PAUSE
```

you will find that you get an error message "Program not found" on the CC-40 and "E13 at 20 not found" on the TI-74. The solution is to change the call to CALL SQ(A,B), that is to delete the R. Then, an input of 3,4 returns the expected -7,24.

MORE TI-74 AVAILABILITY - Page 56 of issue 32 of the Educalc Catalog offers TI-74 hardware at the following prices:

TI-74	BASICALC	$109.95
PC-324	Portable Printer	89.95
AC-9201	Adapter	8.95
TP-324	Printer Paper	4.95
CI-7	Cassette Interface	26.95
74-695	8K RAM Cartridge	39.95
74-696	Learn PASCAL Cartridge	36.95
74-697	Statistics Cartridge	36.95
74-698	Mathematics Cartridge	36.95

where the AC-9201 has not been listed in other TI-74 literature. The AC-9201 was used with the CC-40 which had an AC adapter socket. My TI-74 does not have an adapter socket. Perhaps the AC-9201 is used with the printer or the cassette interface. I have not yet been able to obtain those units. You can order with Mastercard or Visa from Educalc by calling 714-582-2637 or toll-free 1-800-633-2252, extension 342. There is a one dollar shipping/handling charge per order.

ERRATA:

Matrix Inversion Benchmark - During a review of the matrix inversion routine in the TI-74 I found that I had truncated the CC-40 results for the "Mathematics Written in Sand" benchmark problem in V10N1P5. The correct result, whether read from array2 in the CC-40 or array1 of the TI-74 is:

-4.99999 99999 61	-5.99999 99999 56	22.99999 99998 19	8.99999 99999 30
-10.99999 99999 17	-12.99999 99999 08	49.99999 99996 20	19.99999 99998 54
-6.99999 99999 47	-7.99999 99999 42	30.99999 99997 59	11.99999 99999 07
-0.99999 99999 925	-0.99999 99999 917	4.99999 99999 66	1.99999 99999 87

The changes reflect the availability of a fourteenth digit for those values which have two digits to the left of the decimal point.
