Astronomical Computing

Conducted by Roger W. Sinnott

TRS-80 Versus a Giant Brain of Yesteryear

As much for curiosity as for any other reason, I wanted to benchmark my Radio Shack TRS-80 Model I as a scientific computer. The impact of microcomputers in word processing, video games, and so many other areas of modern life is undeniable, of course. But I wanted to try my machine on a scientifically significant problem.

Such a problem had long been familiar to me from celestial mechanics. Soon after World War II, when the first electronic computers were being designed with vacuum tubes, the astronomers W. J. Eckert, Dirk Brouwer, and G. M. Clemence saw the chance to compute new ephemerides for the outer planets Jupiter through Pluto with unprecedented accuracy. Their now-classic calculation, giving heliocentric coordinates to nine decimal places of an astronomical unit at 40-day intervals, spanned the years 1653 to 2060 and was published in 1951, in the 327-page Vol. XII of the U. S. Naval Observatory's *Astronomical Papers.*

These coordinates were used in the *American Ephemeris* from 1960 through 1983, and also in many studies in celestial mechanics. Eckert's team performed the calculation by numerically integrating the equations of motion — a progressive process of great power and simplicity, but with the disadvantage that a single machine error at any stage would render everything thereafter meaningless.

Eckert and his colleagues estimated that a human being working 40 hours a week with a mechanical desk calculator would have required 80 years to do the job, with little chance of avoiding some error along the way. They chose for the project the only machine then capable of it, the IBM Selective Sequence Electronic Calculator (SSEC), pictured below, which became operational in 1948. It had the necessary accuracy (14 significant digits) and also was able to do the work in duplicate as it went along. When the parallel results at any stage differed, the machine would automatically try again; if there was still a discrepancy, it would stop for servicing (such as replacement of a vacuum tube).

How would my TRS-80 fare on such a huge calculation? Fortunately, the introduction to Vol. XII of the *Astronomical Papers* gave complete details as to how the calculation was performed, and the book's printed results offered a check against programming or memory errors in my machine. I had previously written a double-precision floating-point package (working to about 16 significant digits) in Z-80 assembly language, which improved the TRS-80's speed of calculation. When done, I was quite pleased to find that my coordinates at the extreme dates differed from the SSEC's by no more than four units in the ninth decimal place (mostly less) — rather close to the expected accumulation of random rounding errors in a calculation of this kind.

The total problem occupied just under 10 hours and 25 minutes on the TRS-80, or about 10 seconds per 40-day step. The SSEC, for its part, had required almost two minutes per 40-day step, or perhaps 120 hours in all to calculate and to punch cards, not counting interruptions. Of course, my machine saved time by not being saddled with the duplicate calculation, and also by printing out the results only every 2,000 days to save paper.

Notwithstanding these advantages, the TRS-80, had it existed in 1950, would have nudged out the SSEC as the most powerful electronic computer in the world. It probably would have held the lead for a few more years. Even by today's dizzying standards, the humble personal computer appears quite capable of serious scientific work.

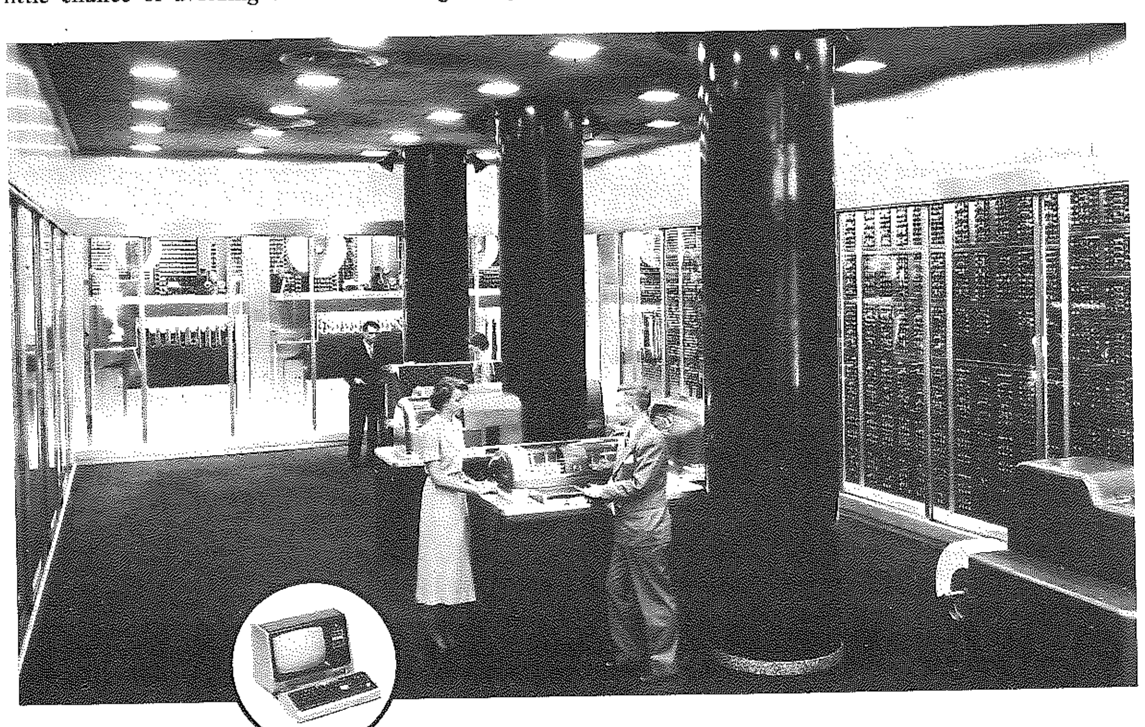
CHRISTOPHER A. FEUCHTER
1124 Bernalillo Place SE.
Albuquerque, N. M. 87123

EDITOR'S NOTE: By coincidence, last summer I too repeated the Eckert-Brouwer-Clemence calculation for a long-term project, using Radio Shack's newer and faster TRS-80 Model 4. My program was written entirely in the Basic language supplied with the machine. As an interpreted language Basic is necessarily slow, but even so the calculation took only three times as long to finish as did Chris Feuchter's, and still beat out the SSEC.

To put things in contemporary perspective, let's see what might be accomplished with one of today's supercomputers, such as the mighty Cray 1, or Control Data's Cyber 205. Their rated speeds are about 100 *megaflops,* where a megaflop is defined as a million double-precision floating-point operations (add, subtract, multiply, or divide) per second. The SSEC, which had no square-root function, executed some 12,000,000 arithmetic operations in the course of the 407-year integration. A supercomputer of today could probably do the job in two seconds or less!

To write the program for such a machine takes vastly longer, of course, and the same usually holds true of microcomputers as well. How often are we really "compute-bound," spending less time writing or typing in the program for our small boxes than waiting for them to issue their results? In that sense we can hardly complain, and the road to scientific progress and discovery seems more open to the home computerist than ever before.

R. W. S.



This 1948 photograph shows IBM's room-size Selective Sequence Electronic Calculator (the SSEC). Vacuum tubes along the right wall are part of the memory, arithmetic, and control units. In the middle of the room are a card reader and two card punches. Wallace J. Eckert (1902-71), who supervised the programming of the outer-planet calculation, had served as director of the U. S. Naval Observatory's Nautical Almanac Office in 1940-45 before becoming head of IBM's pure science department. *Inset:* Radio Shack's TRS-80 Model I, sold from 1978 to 1981.

ABOUT THIS DEPARTMENT

Readers are encouraged to submit *short* programs, in Basic, for our consideration. The few guidelines are almost self-evident. Programs that use special PEEK's and POKE's, fancy graphics commands, disk-file access, user-defined functions, and formatted printer output should be avoided. Excellent as these techniques are, they defeat the transportable quality that makes primitive Basic such a valuable language for exchanging programs.

Each program listed in this department will be reproduced by camera from a computer printout, after we get it to work on one of our office machines, thus avoiding typographical errors. For these first few months, we'll present some of the tested routines we've used at Sky & Telescope.

Electronic access. This is the department's most innovative feature. Here's how it works. First, you need to join the CompuServe Information Service by obtaining a special sign-up kit, available for about $20 from Radio Shack stores and elsewhere in the United States and Canada. Store personnel can explain further how to connect your own computer or terminal to a telephone modem and log onto the system. This gives you access to standard CompuServe features, including wire-service news and computer users' groups. In most parts of the country, you make a *local* phone call to a CompuServe node, and the evening connect charges are only $6.00 per hour. There is no monthly minimum fee.

To access Sky & Telescope's Astronomical Computing department, simply select Personal Computing and then the Programmer's Area when you log on, and at the OK prompt enter R ACCESS. At the Public Access prompt enter TYPE NEWS.DAT[70275,125]. This will immediately give you our latest bulletins on any comet or nova discoveries, for example. It will also catalogue the other items, particularly program listings, that are available on-line.

PROGRAMMING QUICKIES

The two programs offered this month barely scratch the surface of what is possible with a microcomputer. The first, in fact, could be adapted to run on most programmable calculators. But the second, in its full generality, could not.

Summing stellar magnitudes. Our present scale of star brightnesses is about 2,000 years old, but its modern mathematical formulation is due to Norman Pogson in 1856. He defined a five-magnitude range as a factor of 100 in brightness. Each step thus corresponds to the fifth root of 100, or 2.511886 . . . .

Frequently we wish to know the combined magnitude of two stars whose magnitudes are individually known. Since the magnitude scale is logarithmic, ordinary addition won't do. Program Listing 1 solves the problem. To test the program on your own computer, try verifying that the double star Albireo, with components of visual magnitude 3.08 and 5.11, appears to the naked eye at magnitude 2.92.

The program includes a loop, so that additional stars can be combined. This allows, for example, an accurate calculation of the overall brightness of a star cluster when the magnitudes of its components are known. Try this out on the Pleiades, for which about 50 stars are listed in *Sky Catalogue 2000.0,* Vol. 1. Your result should confirm the fact that the total light of this cluster is magnitude +1.3.

Which of the 88 constellations has the most luster? Many similar questions come to mind. To terminate the program at any point, type X.

Interpolation. Every astronomer's repertoire should include a program for interpolating values in an ephemeris or other table. The method given here (Listing 2) was developed by the French mathematician J. L. Lagrange (1736-1813). It is a curious instance in which the computer program looks simpler than the algebraic equations, which may be found in books on numerical techniques.

The program first asks how many known values you are going to enter from a table and allows you to input these one at a time. Then it asks you repeatedly for intermediate values of interest, returning the interpolated value for each. As before, type X to stop the program.

Let's suppose we want to know Mars' angular diameter in arc seconds on various dates in April. We look up and enter the values for four well-spaced dates in the *Astronomical Almanac.* The last value we'll enter is actually for May 4th, but we must pretend this is April 34th to obtain correct answers from the program. Finally, we ask for Mars' diameter at a few intermediate dates. The complete dialogue runs as follows:

```
>RUN
HOW MANY POINTS? 4

X,F? 2,12.87
X,F? 14,14.51
X,F? 22,15.58
X,F? 34,16.91

DESIRED X? 25.5
F:  16.01

DESIRED X? 26.0
F:  16.07

DESIRED X? X
READY
```

Agreement with the *Astronomical Almanac* is excellent, the correct value on April 26.0 being 16″.08, in fact.

A remarkable feature of Lagrange interpolation is that the values entered initially do not have to be in order, or evenly spaced. Accuracy is usually better with uniform spacing, however. Unlike some other interpolation techniques, this method gives no immediate clue to the accuracy of the result. To see how many digits of the result are meaningful in a particular problem, it's advisable to repeat the calculation with more initial values, or at least a different selection of them.

MINI BITS

The Houston Museum of Natural Science in Texas has set up a 24-hour computer bulletin board devoted to astronomy. For access to it with a terminal and 300-baud modem, call 713-526-5671. The service offers current sky happenings and a "Lost in the Universe" game. You may also submit questions to the museum's astronomy staff.

Microcomputers in Astronomy II is the title of the fifth annual Fairborn-IAPPP symposium, to be held July 12-14, 1984. The meeting will be devoted to the use of microcomputers in controlling telescopes and logging data. Amateur and professional astronomers are invited. For information, write to Fairborn Observatory, 1247 Folk Rd., Fairborn, Ohio 45324.

Listing 1

```
10 REM  ADDING MAGNITUDES
15 REM
20 B=100^.2: N=2: C=LOG(10)
25 INPUT "1ST STAR MAG";M1
30 INPUT "2ND STAR MAG";M2
35 M = B^-M1 + B^-M2
40 M = -2.5*LOG(M)/C
45 PRINT "TOTAL MAG:   ";M
50 M1=M: N=N+1: PRINT
55 INPUT "ANOTHER STAR";M$
60 IF M$="X" THEN 75
65 IF M$>"9" THEN 55
70 M2=VAL(M$): GOTO 35
75 PRINT "STAR COUNT:   ";N-1
80 END
```

Listing 2

```
10 REM  LAGRANGE INTERPOLATION
12 REM
14 INPUT "HOW MANY POINTS";N
16 PRINT
18 DIM X(N),F(N),L(N)
20 FOR I=1 TO N
22 INPUT "X,F";X(I),F(I)
24 NEXT I
26 FOR I=1 TO N: L(I)=1
28 FOR J=1 TO N
30 IF J=I THEN 34
32 L(I)=L(I)*(X(I)-X(J))
34 NEXT J
36 L(I)=F(I)/L(I)
38 NEXT I
40 PRINT
42 INPUT "DESIRED X";X$
44 IF X$="X" THEN 78
46 IF X$>"9" THEN 42
48 X=VAL(X$): F1=0
50 FOR I=1 TO N
52 IF X<>X(I) THEN 56
54 F=F(I): F1=1
56 NEXT I
58 IF F1=1 THEN 74
60 T=1: F=0
62 FOR I=1 TO N
64 T=T*(X-X(I))
66 NEXT I
68 FOR I=1 TO N
70 F=F+L(I)*T/(X-X(I))
72 NEXT I
74 PRINT "F: ";F
76 PRINT: GOTO 42
78 END
```