Early data reduction on the IBM Card Programmed Calculator

Between World War I and the end of World War II, Wright Field in Dayton, Ohio, was the center of US military aviation development. However, by the late 1940s, the growing Ohio population was crowding Wright Field. As a result, the engineering flight test of new aircraft coming into Air Force inventory eventually moved to Edwards Air Force Base (EAFB), California, an isolated desert area approximately 60 miles north of Los Angeles.

In 1951, as I was about to graduate from engineering school, the Air Force activated my ROTC Commission, and the college mailed me my diploma. After a trip around the US at the government’s expense, I ended up working under Paul F. Bikle (a particularly analytical engineer) at Edwards and set up the first on-base computer, an IBM Card Programmed Calculator.

Testing by the numbers

As airplanes got more complicated, engineering got more scientific. Seat-of-the-pants feel was gradually supplemented by measurement and, with numerical measurement, came calculation. Bikle deserves much credit for leading the transition to measured flight observations that could be reduced to standard conditions for comparison purposes. He devised a system of engineering spreadsheets to perform the necessary calculations. After the raw observed data were entered on the sheets, a series of stepwise calculations were performed on each data value. The plotted results could be compared from aircraft to aircraft and from flights completed on different days under different (but similar) flying conditions.

The day before I arrived, the base had no Second Lieutenants assigned; two days later, it had 21 on base. Most of the newbies went into support positions such as gofers, but I got to hang around airplanes undergoing flight tests. There were three types of tests performed: a (nonmathematical) evaluation of flying qualities by some of the finest pilots in the world, a performance evaluation that quantified whether the plane met the specification, and a stability evaluation that determined whether the plane had any woeful flight characteristics that made it dangerous. The Air Force arsenal was just turning to jets, and these were interesting times.

When an aircraft arrived at Edwards for evaluation, it spent a few weeks or months in a hanger getting special test instrumentation. This consisted of installing a special instrumentation panel (see Figure 1); bringing a full set of flight instruments to that panel; and mounting additional instruments to show the position of all control surfaces, total amount of fuel on board, and so forth. The panel was surrounded by a light box, and a movie camera recorded all the readings whenever the pilot or flight engineer found something noteworthy.

Individual test points were planned on the ground, and the pilot had a list of tests to fly, each with precise conditions specified. (An example would be a heavy weight takeoff and a maximum performance climb to 40,000 feet.) If there was an extra seat, a flight engineer accompanied the test pilot to make notes and ran the camera on the photopanel. If they had a lot of room, a flunky was even invited for a ride. My first ride, when I was just 22 years old, was thrilling.

When the test was over and the film developed, all the data from each film frame were read and tabulated. At EAFB, there were five or six women (wives of enlisted
men who were mathematically meticulous) who manually read the film images on a machine called a Recordek and hand-tabulated the raw data. The data reduction was then accomplished by engineers with 18-inch slide rules. After I had been around a few months, these women were assigned to the Data Reduction Branch, and I was put in charge.

The senior project test engineers devised spreadsheets to reduce the data from each flight. The data for one test point (speed, altitude, empty weight, fuel aboard, configuration, and so forth) were entered on the blank spreadsheet. Then project engineering personnel, armed with slide rules, were pressed into service to do data reduction. The National Advisory Committee for Aeronautics (NACA) had defined standard atmospheric conditions for each altitude. Naturally, we never had these conditions when we flew, so most of the performance calculations consisted of applying corrections to the observed data to correct observed conditions to the NACA standard day for that altitude. Once that was done, the corrected data point was comparable to other data points flown on different days by different airplanes or by airplanes built by competitive contractors vying for the same production contract.

Each data point resulted in one dot on a performance graph. These points were plotted with a straight pin on high-resolution graph paper so no errors were introduced in the presentation. Much of the work was classified, so the fewer personnel involved, the better. Time was of the essence—not only was there a huge investment tied up in the test airplane and crew, but sometimes production of a new model awaited the completion of the test program.

Enter the CPC

In late 1952, IBM delivered the first on-base computer, an IBM Card Programmed Calculator (see Figure 2). The CPC was a large machine consisting of five heavy boxes: a tabulator about the size of an upright piano (only heavier) that read cards, printed alphanumeric characters, and contained some system control circuits; a five-foot tall electronic calculator containing approximately 1,400 tubes; a summary punch for perforating IBM cards; and two three-foot cubes for auxiliary storage. (For more on the CPC, see the “Brief History of IBM’s Card Programmed Calculator” sidebar.) Thick cables connected all the components. The CPC was controlled by

---

**Brief History of IBM’s Card Programmed Calculator**

In 1949, some engineers from the Northrop Corporation cable-connected an IBM tabulator to an IBM electronic calculating punch. The result was the Card Programmed Calculator. A CPC had three blank plugboards from the factory. Early users of this setup prepared special-purpose plugboards and ran all their data through one plugboard setup before changing the boards for the next step in the calculation sequence. Evidently, this worked satisfactorily for payroll and some engineering calculations involving repetitive calculations on masses of data.

In September 1952, IBM announced a second version of the CPC with more features and faster electronic calculation. About this time, another creative person was faced with lengthy calculations on a small set of data. He (or perhaps she as the name is lost somewhere in history) devised a set of general-purpose plugboards that let calculation specifics be expressed in a deck of punched cards while the plugboards remained installed and intact for days at a time.1

The result was an externally programmed, general-purpose computer. The tabulator portion could read cards or print at 100 cards or lines per minute. During these card cycles, commands and data were shipped to the electronic calculating engine that provided all the basic arithmetic continued on p. 80
three removable plugboards. A set of boards was required for each setup, and sometimes in normal use, multiple sets of plugboards were required to support a lengthy calculation.

A set of plugboards for a CPC contained a large maze of wires, but one clever plugboard setup provided a user-friendly machine to program. When the machine’s arrival was imminent, Vern Kamm (another Second Lieutenant) and I were sent to a Saturday class on the University of California, Los Angeles campus to duplicate the set of plugboards they had. We thought this was good duty because our desert air force base lacked both water and girls. (To our pleasure, the government ordered us to Los Angeles once a week.) The Institute for Numerical Analysis, a part of the National Bureau of Standards, had an outpost at UCLA. The leader of this team was Everett C. Yowell, assisted by Pat Bremer and Fred Hollander. They instructed us in wiring the set of boards and supervised us in their testing.

After the boards were wired, the programming seemed simple. Each program card could contain an operation code and the numeric addresses of three fields. The “A” field told where to fetch the first operand, the “B” field told where to fetch the second operand, and the “C” field told where to store the result. In addition to the operation code for the function to be performed, there were a couple of optional control fields. One of these fields told the printer when to print if selective printing had been chosen by a manual switch setting.

When we returned to Edwards from our last UCLA session, we had a set of plugboards that had been tested, a set of test decks that checked all the machine functions exploited by those plugboards, and a set of listings displaying all the test calculations step by step as they progressed using those plugboards while running those tests—that is, a complete package.

Meanwhile at Edwards, an old World War II barracks had been cleaned out and slightly refurbished for the machine’s arrival. The equipment arrived in five big wooden crates on a low-bed trailer, which we had to quickly unload because they needed the trailer elsewhere. The floors of the barracks were so weak that we set the two heavy machines on bearing plates so the casters would not punch through. After about a week of working with the IBM customer engineer, we successfully ran the UCLA package through our machine and were in business. I immediately started converting spreadsheet calculating procedures into punched card calc decks and learning to operate the system in production. After about a month, we had an installation, trained operators, and a few calc decks that would give the same results as the manual spreadsheets when presented with the same input.

Production operations
The card decks were custom built for each procedure. After a set of input data (flight test point) was punched on a series of cards, these cards were inserted at specific points in a calc computation. For lengthy calculations requiring multiple calculation decks, the output from one calc deck could be punched into summary cards for input to the next pass.

IBM built approximately 700 CPCs. Most were used in accounting shops. However, some were used for scientific and engineering work until the technology moved to full electronic machines. The externally programmed electromechanical CPC was replaced by the internally programmed full electronic IBM 701/702 in the mid-1950s.

References and notes
deck. In addition, if special tables of standard data (NACA standard atmosphere in punched card form) were required, these were also inserted into the calc deck. All the cards were color coded by type, and each card had a type code and a sequential number punched into it. Identifying information from each card was printed (by a separate IBM machine called an Interpreter) on top of each card to further ease manual handling. Each card deck had a backup because card decks were sensitive to changes in humidity and mishandling.

When the machine was available, a calc deck was loaded into the hopper and the process started. While one deck was run, the data from the previous run was extracted from its twin and new data cards were inserted into the twin for the next run. Thus, every several minutes a new calc deck was loaded with the next data point. If a flight had as many as 50 data points, three hours were required to reduce all that data. If ever the results of a calculation looked strange, a hardware malfunction was immediately suspected, and a deck of test cards was run to verify operation.

The nearest hamburger place was down the street a few blocks. The standard card hopper would not hold enough cards to get something to eat before the machine stopped, so we rigged a hopper extension so we could load 2,000 cards. If we hurried, the machine was still running when we got back.

In the 1950s, electronic technology consisted of wire relays, vacuum tubes, and plug wires—none a model of reliability. For example, in 1953, when an earthquake shook our barracks, the machine jumped off the bearing plates, and one caster punched through the floor. It took IBM maintenance a week to find all the bad circuits (mainly cold solder joints). When I was discharged, this simple installation could reduce a day’s worth of test data overnight (if there weren’t too many test points and if the CPC held together), with only one operator (instead of several test engineers with slide rules) and fewer cleared personnel (important, if the test results were classified).

Moving on

I didn’t know it at the time, but IBM was in transition. They had been successful building punched card equipment and were somewhat late offering electronic systems. The CPC was the root of the electronic family tree that went CPC, 650, 700 series, 7000 series, and S360/370.

When I became a civilian again, I went to work at Convair-Fort Worth. Shortly thereafter, they received IBM 701 Serial No. Seven. Although most of the Convair programming crew struggled with binary, octal, assembly language, and early subroutines, I used a system of interpretative software provided by IBM called SpeedCode. SpeedCode was the brainchild of John Backus (who later provided us with Fortran) and John Sheldon. SpeedCode was an interpretative decimal three-address programming and operating system. It was so similar to the CPC plugboards I had been using that I was quickly productive. The transition was like moving between two versions of the Mac operating system.

Robert L. Patrick
clpatrick@mac.com

On the word Software

Some time during 1958 (probably spring or summer) while running an operations research group at Shell Development Company in Houston, I went to an ACM conference. There I encountered Nicholas Metropolis, director of the Institute for Computer Research at the University of Chicago. Years earlier, I had taken my first physics class from him. The result of our short conversation was that I received an offer for an appointment from the University of Chicago to the lofty position of assistant professor of applied mathematics in the Institute for Computer Research, commencing in fall 1958.

This institute was a newly formed addition to the University of Chicago Research Institutes, which included the Enrico Fermi Institute for Nuclear Studies and the Institute for the Study of Metals. We all shared the same building on Ellis Avenue on the southside of Chicago. People at the Institute for Computer Research were building a computer under a contract from the Atomic Energy Commission; it was probably the first transistorized computer to be built under AEC auspices.

When I arrived, I found an old friend, David Jacobsohn, among the crew that was building the computer. He and others referred to the physical implementation as hardware, a term that bemused me because I regarded hardware as something to be bought in a hardware store. Because I was the sole person in charge of providing the initial programming, I put a sign on my office door saying “Software Department” and thought that I was being incredibly clever. I don’t think enough people saw my sign for me to claim that the use of the term radiated from my office door. Perhaps, as an independent inventor, I can claim priority.

Herbert Kanner
kanner@acm.org